Lab 5: The Sound of Gunfire, Off in the Distance

Ben Thomas

10/16/2019

Data

Today, I'm going to do some classification. I'll start by downloading our dataset, which contains data on a set of 161 countries from 1960 to 1995, in 5 year chunks. This dataset includes information about whether they're in a civil war in addition to many economic and social characteristics.

```
war <- read.csv("http://www.stat.cmu.edu/~cshalizi/uADA/15/hw/06/ch.csv", row.names = 1)
```

Logistic model

Let's create a simple logistic model to predict whether or not there is currently a civil year for a given country and year. Note that I've created a new dataset, which includes a squared term for the "exports" variable.

Part 1: Estimation

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------------
## v ggplot2 3.2.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------------------- ti
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
war_new <- war %>%
  mutate(sqr.exports = exports^2)
m1 <- glm(start ~
            exports +
            sqr.exports +
            schooling +
            growth +
            peace +
            concentration +
            lnpop +
            fractionalization +
            dominance,
          data = war_new,
          family = binomial)

summary(m1)$coef
```

```
##                        Estimate   Std. Error   z value     Pr(>|z|)
## (Intercept)       -1.307308e+01 2.795232e+00 -4.676920 2.912151e-06
## exports            1.893704e+01 5.865136e+00  3.228747 1.243336e-03
```

```
## sqr.exports       -2.944321e+01 1.178128e+01 -2.499153 1.244907e-02
## schooling         -3.155633e-02 9.784271e-03 -3.225210 1.258804e-03
## growth            -1.152294e-01 4.307150e-02 -2.675305 7.466130e-03
## peace             -3.713408e-03 1.093156e-03 -3.396962 6.813847e-04
## concentration     -2.486984e+00 1.005201e+00 -2.474115 1.335665e-02
## lnpop              7.677375e-01 1.657549e-01  4.631763 3.625655e-06
## fractionalization -2.134524e-04 9.101928e-05 -2.345134 1.902023e-02
## dominance          6.703907e-01 3.535247e-01  1.896305 5.791973e-02
```

As noted in the table above, all of the variables in this model are significant at the 5% level aside from the "dominance" variable.

Part 2: Interpretation

While it's great that the majority of the variables in this logistic model are significant, it's not immediately intuitive as to how we should interpret them. To illustrate this, I'll use the results of this model to see the predicted probability that India in 1975 had started a civil war.

From the "war_new" dataset:

- "exports" = 0.026
- "schooling" = 36
- "growth" = 0.322
- "peace" = 112
- "concentration" = 0.537
- "lnpop" = 20.23462
- "fractionalization" = 2937
- "dominance" = 0
- "sqr.exports" = 0.000676

We can calculate the probability (in log odds) that India had started a civil war by adding together these values, each multiplied by their respective coefficients from the model, in addition to the coefficient. This equals:
$-1.307308*10^{1}+1.893704*10^{1}*0.026-2.944321*10^{1}*0.000676-3.155633*10^{-2}*36-1.152294*10^{-1}*0.322-3.713408*10^{-3}*112-2.486984*10^{0}*0.537+7.677375*10^{-1}*20.23462-2.134524*10^{-4}*2937+6.703907*10^{-1}*0$

After solving out the algebra, the log odds that India had started a civil war in 1975 is -0.6171975. Since we want to understand the probability, we can convert this using R. I've done so below.

```
log.odds = (m1$coefficients[[1]] +
              m1$coefficients[[2]] * war_new$exports[500] +
              m1$coefficients[[3]] * war_new$sqr.exports[500] +
              m1$coefficients[[4]] * war_new$schooling[500] +
              m1$coefficients[[5]] * war_new$growth[500] +
              m1$coefficients[[6]] * war_new$peace[500] +
              m1$coefficients[[7]] * war_new$concentration[500] +
              m1$coefficients[[8]] * war_new$lnpop[500] +
              m1$coefficients[[9]] * war_new$fractionalization[500] +
              m1$coefficients[[10]] * war_new$dominance[500])
odds = exp(log.odds)
prob = odds/(1+odds)
prob
```

```
## [1] 0.3504199
```

From this, we see that per our logit model, the probability that India had started a civil war is 0.3504199, or around 35%. Note that this prediction (no) would be correct, as India had not begun a civil war in this period.

We can see how this probability changes as the variables change. For example, if we take a country identical to India in 1975, except with its male secondary school enrollment rate is 30 pts higher, then the probability that it had started a civil war is now 0.17309, as seen below.

```
log.odds = (m1$coefficients[[1]] +
              m1$coefficients[[2]] * war_new$exports[500] +
              m1$coefficients[[3]] * war_new$sqr.exports[500] +
              m1$coefficients[[4]] * (war_new$schooling[500] + 30) +
              m1$coefficients[[5]] * war_new$growth[500] +
              m1$coefficients[[6]] * war_new$peace[500] +
              m1$coefficients[[7]] * war_new$concentration[500] +
              m1$coefficients[[8]] * war_new$lnpop[500] +
              m1$coefficients[[9]] * war_new$fractionalization[500] +
              m1$coefficients[[10]] * war_new$dominance[500])
odds = exp(log.odds)
prob = odds/(1+odds)
prob
```

```
## [1] 0.17309
```

If we instead increased the ratio of commodity exports to GDP by 0.1, the new probability would be 0.6961378.

```
log.odds = (m1$coefficients[[1]] +
              m1$coefficients[[2]] * (war_new$exports[500]+.1) +
              m1$coefficients[[3]] * (war_new$exports[500]+.1)^2 +
              m1$coefficients[[4]] * war_new$schooling[500] +
              m1$coefficients[[5]] * war_new$growth[500] +
              m1$coefficients[[6]] * war_new$peace[500] +
              m1$coefficients[[7]] * war_new$concentration[500] +
              m1$coefficients[[8]] * war_new$lnpop[500] +
              m1$coefficients[[9]] * war_new$fractionalization[500] +
              m1$coefficients[[10]] * war_new$dominance[500])
odds = exp(log.odds)
prob = odds/(1+odds)
prob
```

```
## [1] 0.6961378
```

Let's look at a couple more cases. The predicted probability that Nigeria had begun a civil war in the 1965 period would then be 0.1709917.

```
log.odds = (m1$coefficients[[1]] +
              m1$coefficients[[2]] * war_new$exports[802] +
              m1$coefficients[[3]] * war_new$sqr.exports[802] +
              m1$coefficients[[4]] * war_new$schooling[802] +
              m1$coefficients[[5]] * war_new$growth[802] +
              m1$coefficients[[6]] * war_new$peace[802] +
              m1$coefficients[[7]] * war_new$concentration[802] +
              m1$coefficients[[8]] * war_new$lnpop[802] +
              m1$coefficients[[9]] * war_new$fractionalization[802] +
              m1$coefficients[[10]] * war_new$dominance[802])
odds = exp(log.odds)
prob = odds/(1+odds)
prob
```

```
## [1] 0.1709917
```

If we look at an identical country, with a higher secondary school enrollment, the probability changes to

0.07410315.

```
log.odds = (m1$coefficients[[1]] +
              m1$coefficients[[2]] * war_new$exports[802] +
              m1$coefficients[[3]] * war_new$sqr.exports[802] +
              m1$coefficients[[4]] * (war_new$schooling[802] + 30) +
              m1$coefficients[[5]] * war_new$growth[802] +
              m1$coefficients[[6]] * war_new$peace[802] +
              m1$coefficients[[7]] * war_new$concentration[802] +
              m1$coefficients[[8]] * war_new$lnpop[802] +
              m1$coefficients[[9]] * war_new$fractionalization[802] +
              m1$coefficients[[10]] * war_new$dominance[802])
odds = exp(log.odds)
prob = odds/(1+odds)
prob
```

```
## [1] 0.07410315
```

If we instead changed the exports variable, the new probability would be 0.3310044.

```
log.odds = (m1$coefficients[[1]] +
              m1$coefficients[[2]] * (war_new$exports[802] + 0.1) +
              m1$coefficients[[3]] * (war_new$exports[802] + 0.1)^2 +
              m1$coefficients[[4]] * war_new$schooling[802] +
              m1$coefficients[[5]] * war_new$growth[802] +
              m1$coefficients[[6]] * war_new$peace[802] +
              m1$coefficients[[7]] * war_new$concentration[802] +
              m1$coefficients[[8]] * war_new$lnpop[802] +
              m1$coefficients[[9]] * war_new$fractionalization[802] +
              m1$coefficients[[10]] * war_new$dominance[802])
odds = exp(log.odds)
prob = odds/(1+odds)
prob
```

```
## [1] 0.3310044
```

In the above examples, the changes in probability are not equal, even though we changed the inputs by the same amount. This is because what has changed equally in each case was the log odds, and a 1 unit change in log odds at any specific value will not result in the same change in probability as a 1 unit change in log odds at a different starting log odds value. This is because there are a couple transformations between the two.

Part 3: Confusion

Let's now try to assess how good my model is at predicting whether or not there is war in a specific country in a specific year. One way to do this is to build a confusion matrix, which displays the number of cases predicted as no or yes, alongside their actual values. To do this, we first need to select the portion of the dataset which has values for all variables used as predictors (otherwise we won't be able to make a prediction).

```
war.trimmed <- na.omit(war_new)
m2 <- glm(start ~
            exports +
            sqr.exports +
            schooling +
            growth +
            peace +
            concentration +
            lnpop +
            fractionalization +
            dominance,
          data = war.trimmed,
          family = binomial)
log.war.pred <- ifelse(m2$fit < 0.5, "No", "Yes")
actual.war.trim <- ifelse(war.trimmed$start < 0.1, "No", "Yes")

conf_log <- table(log.war.pred, actual.war.trim)
conf_log
```

```
##             actual.war.trim
## log.war.pred  No Yes
##          No  637  43
##          Yes   5   3
```

We see that generally, our model tended to predict that there was no war. We can develop a more sophisticated assessment of how good our model is by calculating the miscalculation rate (i.e. the fraction of cases that the model predicted incorrectly). This miscalculation rate is calculated below, and is 0.06976.

```
log.miss.rate <- (1/nrow(war.trimmed))*(conf_log[2, 1] + conf_log[1, 2])
log.miss.rate
```

```
## [1] 0.06976744
```

We can compare the miscalculation rate of our model to that of a foolish pundit, who always predicts no war. We'll first call it on the whole dataset, and then call it on only the section of the dataset used by the logistic model as well. Over the whole dataset, the pundit's miscalculation rate is 0.06055. Over just the section of the data also used by the logistic model, the pundit's miscalculation rate is 0.06686. Both of these rates are actually better than the calculation rate of our model! This is a little concerning.

```
# Make the confusion matrix for the pundit using the whole dataset
full.pundit.pred <- ifelse(war$start < 2, "No", "Yes")
actual.war <- ifelse(war$start <.5, "No", "Yes")
full.conf.pundit <- table(full.pundit.pred, actual.war)

# Find the pundit's miscalculation rate using the whole dataset
full.pundit.miss.rate <- (1/nrow(war))*(full.conf.pundit[1, 2])
full.pundit.miss.rate
```

```
## [1] 0.06055901
```

```r
# Make the confusion matrix for the pundit using the trimmed dataset
trim.pundit.pred <- ifelse(war.trimmed$start < 2, "No", "Yes")
trim.conf.pundit <- table(trim.pundit.pred, actual.war.trim)

# Find the pundit's miscalculation rate using the trimmed dataset
trim.pundit.miss.rate <- (1/nrow(war.trimmed))*(trim.conf.pundit[1, 2])
trim.pundit.miss.rate
```

```
## [1] 0.06686047
```

Part 4: Comparison

We've created a logistic model to predict whether or not a given country will be at war in a given year, and found that this model is worse than a model which always predicts no war. Now, let's try some different modeling approaches to see if we're able to create a model that beats the foolish pundit (or just our earlier model).

Let's start with an LDA (Linear Discriminant Analysis) model.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda.fit <- lda(start ~
                 exports +
                 sqr.exports +
                 schooling +
                 growth +
                 peace +
                 concentration +
                 lnpop +
                 fractionalization +
                 dominance,
               data = war.trimmed)
lda.fit
```

```
## Call:
## lda(start ~ exports + sqr.exports + schooling + growth + peace +
##     concentration + lnpop + fractionalization + dominance, data = war.trimmed)
##
## Prior probabilities of groups:
##          0          1
## 0.93313953 0.06686047
##
## Group means:
##     exports sqr.exports schooling     growth    peace concentration
## 0 0.1574330  0.04505594  45.64548 1.73095794 357.7850     0.6038349
## 1 0.1668478  0.04127454  28.34783 0.04384783 204.2826     0.5762391
##      lnpop fractionalization dominance
## 0 15.68224          1764.882 0.4376947
## 1 16.58465          2146.696 0.4565217
##
## Coefficients of linear discriminants:
##                              LD1
## exports             7.5279499420
## sqr.exports        -9.3781631631
## schooling          -0.0063973381
## growth             -0.1242737735
## peace              -0.0041224852
## concentration      -1.1570459065
## lnpop               0.3813814561
## fractionalization -0.0001052021
```

```
## dominance          0.3644566472
```

This is a lot of output, and hard to interpret. Let's try to use this LDA model to predict whether or not there will be war in our dataset and see how well this model does at prediction.

```
lda.pred <- predict(lda.fit, war.trimmed)
lda.class <- lda.pred$class

conf_lda <- table(lda.class, actual.war.trim)
conf_lda
```

```
##          actual.war.trim
## lda.class  No Yes
##         0 636  40
##         1   6   6
```

Let's calculate the miscalculation rate, to be able to compare the predictive power of the LDA model with that of the logistic model.

```
lda.miss.rate <- (1/nrow(war.trimmed))*(conf_lda[2, 1] + conf_lda[1, 2])
lda.miss.rate
```

```
## [1] 0.06686047
```

As we see here, the miscalculation rate of the LDA model is 0.06686047, slightly lower than that of the logistic model (0.06976744). Note that this is still ever so slightly above the miscalculation rate of the foolish pundit (using the trimmed dataset).

Let's now turn to quadratic discriminant analysis (QDA) to see how that compares.

```
# Let's fit the model
qda.fit <- qda(start ~
                 exports +
                 sqr.exports +
                 schooling +
                 growth +
                 peace +
                 concentration +
                 lnpop +
                 fractionalization +
                 dominance,
               data = war.trimmed)

# Make a confusion matrix
qda.class <- predict(qda.fit, war.trimmed)$class
conf_qda <- table(qda.class, actual.war.trim)
conf_qda
```

```
##          actual.war.trim
## qda.class  No Yes
##         0 618  26
##         1  24  20
```

```
# Find the miscalculation rate
qda.miss.rate <- (1/nrow(war.trimmed))*(conf_qda[2, 1] + conf_qda[1, 2])
qda.miss.rate
```

```
## [1] 0.07267442
```

The QDA model, as we see above, actually has the highest miscalculation rate of any model we've tried, and

none of our models have had a miscalculation rate lower than that of the foolish pundit. Very concerning! Of our 3 models:

- LDA had the lowest miscalculation rate, which might make sense given that its parameter estimates are more stable than that of the logistic model
- the logistic model had the second-lowest miscalculation rate
- the QDA model had the highest miscalculation rate

Chapter 4 exercises

Problem 4

(a) If X is uniformly distributed on [0,1], and we use observations that are within 10% of the range of X closest to the test observation, then on average we will use 10% of the available observations to make our prediction.

(b) Now, given two X values, each ranging from [0,1], when we use only observations within 10% of the range of each X closest to the test observation, then on average we will use 1% of the available observations to make a prediction.

(c) In this case, with 100 predictors, all evenly distributed, the fraction of observations we would use to make a prediction (given that we use only observations within the 10% of the range of each feature closest to the test observation) would be $\frac{1}{10^{100}}$, a tiny number.

(d) As we saw in parts a through c, as the number of predictors increases, the amount of observations "close" to the test observation decreases rapidly. When we had 100 predictors, the fraction of observations we used to make a prediction was miniscule. Therefore, when we have many predictors, it will be difficult to have a large sample to predict any particular value using KNN.

(e) We want to find the range of X (for each predictor, given differing numbers of predictors) that will generate 10% of the available observations for use in prediction. We can generalize this by saying that the fraction of available observations is equivalent to the range we use of each predictor, to the power of the number of predictors used. In equation terms:

$$\frac{used\ \ observations}{total\ \ observations} = Range^p$$

When p = 1, and the fraction of observations used = 0.1, then the range of each predictor used is also 0.1.
When p = 2, and the fraction of observations used = 0.1, then the range of each predictor used is 0.316.
When p = 100, and the fraction of observations used = 0.1, then the range of each predictor used is 0.977.

Problem 6

(a) We know that the coefficients from the logit model here represent the log of the odds ratio. To convert these to probabilities, we find the odds and take the ratio, as follows:

$$probability = \frac{e^{-6+(0.05)(40)+(1)(3.5)}}{1+e^{-6+(0.05)(40)+(1)(3.5)}} = 0.377541$$

The probability that this particular student gets an A in the class is therefore around 38%.

(b) Let's do the same here to find how many hours this student would need to have a 50% chance of getting an A in the class.

$$\frac{e^{-6+(0.05)(x)+(1)(3.5)}}{1+e^{-6+(0.05)(x)+(1)(3.5)}} = 0.5$$

This solves to:

$$x = 50$$

The student would therefore need to study for 50 hours to have a 50% chance of getting an A in the class.

Problem 7

We're given:

- the mean of X (last year's percent profit) for companies who gave out dividends $\overline{X} = 10$
- the mean of X for companies who didn't: $\overline{X} = 0$
- the variance of X for both sets: $\hat{\sigma}^2 = 36$
- the percent of companies who gave out dividends (80%)

Assuming that X is normal, we want to predict the probability that a company will give out dividends this year, given that its percentage profit last year (X) was 4.

To find this, we can reduce Equation 4.12 from the book to the following form. (Thanks to Josh Dey for computational ability.) Note that 1 denotes that the company pays out a dividend, and 0 denotes that they do not.

$$\frac{\pi_1 \times e^{\frac{-(x-\mu_1)^2}{2\sigma^2}}}{\pi_1 \times e^{\frac{-(x-\mu_1)^2}{2\sigma^2}} + \pi_0 \times e^{\frac{-(x-\mu_0)^2}{2\sigma^2}}}.$$

We can then use the given values and this form of Equation 4.12 to find the the probability that the company pays out a dividend. I've used R to do so below.

```
pi_d <- .8
pi_n <- .2
mu_d <- 10
mu_n <- 0
variance <- 36
x <- 4
f <- (pi_d * exp((-(x-mu_d)^2)/(2*variance)))/
  (pi_d*exp((-(x-mu_d)^2)/(2*variance))+pi_n*exp((-(x-mu_n)^2)/(2*variance)))
f
```

```
## [1] 0.7518525
```

As we see here, there is a 75.2% chance that the company in question pays out a dividend this year.