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

Paul Nguyen

10/10/2019

Lab 5

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

mexports <- glm(start ~ exports,
          data = warnona,
          family = binomial)
summary(mexports)$coef
```

```
##                Estimate Std. Error    z value     Pr(>|z|)
## (Intercept) -2.7093739  0.2299152 -11.784229 4.707063e-32
## exports      0.4531007  1.0347670   0.437877 6.614754e-01
```

```
mschooling <- glm(start ~ schooling,
          data = warnona,
          family = binomial)
summary(mschooling)$coef
```

```
##                 Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept) -1.81282639 0.243074344 -7.457909 8.790620e-14
## schooling   -0.02273463 0.006439439 -3.530529 4.147293e-04
```

```
if (summary(mschooling)$coef[2,4] < .05) {
  print(summary(mschooling)$coef[2,4])
}
```

```
## [1] 0.0004147293
```

```
mgrowth <- glm(start ~ growth,
          data = warnona,
          family = binomial)
summary(mgrowth)$coef
```

```
##               Estimate Std. Error    z value     Pr(>|z|)
## (Intercept) -2.519588 0.15413761 -16.346357 4.617768e-60
## growth      -0.125805 0.04020436  -3.129138 1.753199e-03
```

```
if (summary(mgrowth)$coef[2,4] < .05) {
  print(summary(mgrowth)$coef[2,4])
}
```

```
## [1] 0.001753199
```

```
mpeace <- glm(start ~ peace,
          data = warnona,
```

```r
          family = binomial)
summary(mpeace)$coef
```

```
##                   Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept) -0.967329338 0.274512569 -3.523807 4.253938e-04
## peace       -0.005950402 0.001027654 -5.790276 7.027101e-09
```

```r
if (summary(mpeace)$coef[2,4] < .05) {
  print(summary(mpeace)$coef[2,4])
}
```

```
## [1] 7.027101e-09
```

```r
mconcentration <- glm(start ~ concentration,
          data = warnona,
          family = binomial)
summary(mconcentration)$coef
```

```
##                 Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)   -2.2735740  0.4364301 -5.2094803 1.893704e-07
## concentration -0.6139364  0.7069974 -0.8683714 3.851910e-01
```

```r
if (summary(mconcentration)$coef[2,4] < .05) {
  print(summary(mconcentration)$coef[2,4])
}


mlnpop <- glm(start ~ lnpop,
          data = warnona,
          family = binomial)
summary(mlnpop)$coef
```

```
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -8.1296159 1.54829319 -5.250695 1.515261e-07
## lnpop        0.3407347 0.09339644  3.648262 2.640201e-04
```

```r
if (summary(mlnpop)$coef[2,4] < .05) {
  print(summary(mlnpop)$coef[2,4])
}
```

```
## [1] 0.0002640201
```

```r
mfractionalization <- glm(start ~ fractionalization,
          data = warnona,
          family = binomial)
summary(mfractionalization)$coef
```

```
##                          Estimate   Std. Error   z value     Pr(>|z|)
## (Intercept)       -2.811553e+00 2.158063e-01 -13.02813 8.465489e-39
## fractionalization  9.000894e-05 7.198069e-05   1.25046 2.111317e-01
```

```r
if (summary(mfractionalization)$coef[2,4] < .05) {
  print(summary(mfractionalization)$coef[2,4])
}

mdominance <- glm(start ~ warnona[,11],
          data = warnona,
          family = binomial)
```

```r
summary(mdominance)$coef
```

```
##                  Estimate Std. Error     z value     Pr(>|z|)
## (Intercept)   -2.6700021  0.2068084 -12.9105132 3.926753e-38
## warnona[, 11]  0.0761699  0.3065082   0.2485085 8.037410e-01
```

```r
if (summary(mdominance)$coef[2,4] < .05) {
  print(summary(mdominance)$coef[2,4])
}

#try to make in one chunk
#modelvar <- colnames(warnona)
#odelvar[3]


#for (i in 4:11) {
 # modelvar[i] <- glm(start ~ warnona[,i],
              #data = warnona,
               #family = binomial)

#}
#this did not work
```

Note: I have now read the instructions more clearly and am now re-doing the estimate part now.

```r
finalmod <- glm(start ~ growth +
                        exports +
                        fractionalization +
                        I(exports^2) +
                        lnpop +
                        peace +
                        schooling +
                        dominance +
                        concentration,
                data = war,
                family = binomial)
summary(finalmod)$coef
```

```
##                        Estimate   Std. Error   z value     Pr(>|z|)
## (Intercept)       -1.307308e+01 2.795232e+00 -4.676920 2.912151e-06
## growth            -1.152294e-01 4.307150e-02 -2.675305 7.466130e-03
## exports            1.893704e+01 5.865136e+00  3.228747 1.243336e-03
## fractionalization -2.134524e-04 9.101928e-05 -2.345134 1.902023e-02
## I(exports^2)      -2.944321e+01 1.178128e+01 -2.499153 1.244907e-02
## lnpop              7.677375e-01 1.657549e-01  4.631763 3.625655e-06
## peace             -3.713408e-03 1.093156e-03 -3.396962 6.813847e-04
## schooling         -3.155633e-02 9.784271e-03 -3.225210 1.258804e-03
## dominance          6.703907e-01 3.535247e-01  1.896305 5.791973e-02
## concentration     -2.486984e+00 1.005201e+00 -2.474115 1.335665e-02
```

Significant Coefficients(P<.05): growth, exports, fractionalization, exports^2, lnpop, peace, schooling, concentration

```r
india <- data.frame(war[500,])
predindia <- predict(finalmod, newdata = india, type = "response")
predindia
```

```
##       500
## 0.3504199
```

```
indiaschool <- data.frame(india) %>%
  mutate(schooling = 66)
predindiaschool <- predict(finalmod, newdata = indiaschool, type = "response")
predindiaschool
```

```
##       1
## 0.17309
```

```
indiaexportsgdp <- data.frame(india) %>%
  mutate(exports = india$exports + .1)
predindiaexports <- predict(finalmod, newdata = indiaexportsgdp, type = "response")
predindiaexports
```

```
##         1
## 0.6961378
```

model’s predicted probability for India in 1975 is .3504199. with schooling raised by 30 points, the new probability for India is .17309. with exports raised by .1, the new probability for India is .6961378.

```
nigeria <- data.frame(war[802,])
prednig <- predict(finalmod, newdata = nigeria, type = "response")
prednig
```

```
##       802
## 0.1709917
```

```
nigeriaschool <- data.frame(nigeria) %>%
  mutate(schooling = nigeria$schooling +30)
prednigschool <- predict(finalmod, newdata = nigeriaschool, type = "response")
prednigschool
```

```
##          1
## 0.07410315
```

```
nigeriaexports <- data.frame(nigeria) %>%
  mutate(exports = nigeria$exports +.1)
prednigexports <- predict(finalmod, newdata = nigeriaexports, type = "response")
prednigexports
```

```
##         1
## 0.3310044
```

```
(prednig - prednigschool) - (predindia - predindiaschool)
```

```
##         802
## -0.08044127
```

```
(prednig - prednigexports) - (predindia - predindiaexports)
```

```
##       802
## 0.1857053
```

Under our model, the predicted probability for Nigeria under going civil war is .1709917. with schooling raised 30 points, new probability is .07410315 with exports raised by .1, new probability is .3310044

I think that the reason why the differences in the change in probability are different for the two countries, even though I changed each variable by the same amount, is due to the form of the logistic function, which I based my model on. In the log model, the probability is determined by e raised to my beta values.

In this problem, India and Nigeria have different values for their respective beta terms. If I just add .1 or 30 to a specific beta, even though I add identical amounts, the difference is not the same.

```
my_log_pred <- ifelse(finalmod$fit < 0.5, "0", "1")
data.frame(log_pred = my_log_pred[0:5],
           true = war$start[0:5])
```

```
##    log_pred true
## 10        1    0
## 11        0    0
## 12        0    0
## 13        0    1
## 14        0   NA
```

```
conflog <- table(my_log_pred, warnona$start)
conflog
```

```
##
## my_log_pred   0   1
##           0 637  43
##           1   5   3
```

```
zz <- war %>%
  drop_na(start)
sum(zz$start)
```

```
## [1] 78
```

```
nrow(zz) - sum(zz$start)
```

```
## [1] 1089
```

```
nrow(warnona) - sum(warnona$start)
```

```
## [1] 642
```

missclassificationrate = 5 + 43 / 688, .06976744 if pundit always predicts no war, the pundit will be correct 1089 / 1167 times. On my dataset witho no nas, the pundit will be correct 643 / 688 times.

```
est <- warnona %>%
  group_by(start) %>%
  summarise(n = n(),
          prop = n/nrow(war),
          mugrowth = mean(growth),
          muexports = mean(exports),
          mufractionalization = mean(fractionalization),
          muexports2 = mean(exports^2),
          mulnpop = mean(lnpop),
          mupeace = mean(peace),
          muschooling = mean(schooling),
          mudominance = mean(dominance),
          muconcentration = mean(concentration),
          ssxgrowth = var(growth) * (n - 1),
          ssxexports = var(exports) * (n - 1),
          ssxfractionalization = var(fractionalization) * (n - 1),
          ssxexports2 = var(exports^2) * (n - 1),
          ssxlnpop = var(lnpop) * (n - 1),
          ssxpeace = var(peace) * (n - 1),
          ssxschooling = var(schooling) * (n - 1),
            ssxdominance = var(dominance) * (n - 1),
            ssxconcentration = var(concentration) * (n - 1)
            )

pi_n <- pull(est[1,3])
pi_y <- pull(est[2,3])
mu_n <- (est[1,4:12])
mu_y <- (est[2,4:12])
sigma_sq <- (1/(nrow(warnona) - 2)) * sum(est$ssx)
```

```
## Warning: Unknown or uninitialised column: 'ssx'.
```

```
#can i use lda, ans: yes
ldamodel <- lda(start ~ growth +
                        exports +
                        fractionalization +
                        I(exports^2) +
                        lnpop +
                        peace +
                        schooling +
                        dominance +
                        concentration,
                data = warnona)
ldamodel
```

```
## Call:
## lda(start ~ growth + exports + fractionalization + I(exports^2) +
##     lnpop + peace + schooling + dominance + concentration, data = warnona)
##
## Prior probabilities of groups:
##          0          1
## 0.93313953 0.06686047
##
## Group means:
##       growth   exports fractionalization I(exports^2)    lnpop    peace
## 0 1.73095794 0.1574330          1764.882   0.04505594 15.68224 357.7850
## 1 0.04384783 0.1668478          2146.696   0.04127454 16.58465 204.2826
##   schooling dominance concentration
## 0  45.64548 0.4376947     0.6038349
## 1  28.34783 0.4565217     0.5762391
##
## Coefficients of linear discriminants:
##                              LD1
## growth            -0.1242737735
## exports            7.5279499420
## fractionalization -0.0001052021
## I(exports^2)      -9.3781631631
## lnpop              0.3813814561
## peace             -0.0041224852
## schooling         -0.0063973381
## dominance          0.3644566472
## concentration     -1.1570459065
```

```
ldapred <- predict(ldamodel, data = warnona)
lda.class <- ldapred$class
```

```
table(lda.class, warnona$start)
```

```
##
## lda.class   0   1
##         0 636  40
##         1   6   6
```

```
(40+6)/(636 + 40 +6 +6)
```

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

missclassification rate: .06686047

QDA Model

```
qdamodel <- qda(start ~ growth +
                        exports +
                        fractionalization +
                        I(exports^2) +
                        lnpop +
                        peace +
                        schooling +
                        dominance +
                        concentration,
                data = warnona)
qdamodel
```

```
## Call:
## qda(start ~ growth + exports + fractionalization + I(exports^2) +
##     lnpop + peace + schooling + dominance + concentration, data = warnona)
##
## Prior probabilities of groups:
##          0          1
## 0.93313953 0.06686047
##
## Group means:
##       growth   exports fractionalization I(exports^2)    lnpop    peace
## 0 1.73095794 0.1574330          1764.882   0.04505594 15.68224 357.7850
## 1 0.04384783 0.1668478          2146.696   0.04127454 16.58465 204.2826
##   schooling dominance concentration
## 0  45.64548 0.4376947     0.6038349
## 1  28.34783 0.4565217     0.5762391
```

```
qdapred <- predict(ldamodel, data = warnona)
qda.class <- qdapred$class
table(qda.class, warnona$start)
```

```
##
## qda.class   0   1
##         0 636  40
##         1   6   6
```

missclassification rate: .06686047. I would say that the lda and qda missclassification rates are lower because they are more flexible models; they predict more parameters than the logistic regression. Since we are only working with training data and not test data, we do not have to worry about overfitting the model. Thus, the more flexible models will have the lower misclassification rate.

Problem Set

Exercise 4

a. With $p = 1$ and X uniformly distributed on [0,1], if we choose to predict a test response using observations that are within 10% the range of X closest to the test observation, the average fraction of the available obsercations we will use to make the prediction is 1/10.
b. With $p = 2$ features, X_1 and X_2, and with (X_1, X_2) uniformly distributed on $[0, 1] \times [0, 1]$. When making a prediction using only observations that are within 10% of range of X_1 and X_2, we will on average only have 1/100 of our available observations to make this prediction
c. Now with $p = 100$ features, and observations uniformly distributed on each feature. If we want to predict a test observation using 10% of each features range that is closest to that test observation, we will only be using 1/(10ˆ100) of our available observations.
d. The drawback of using KNN when p is very large is that there are very few training observations "near" a test observation. As we increase our p's we decrease the amount of our observations that fulfil each "p". So when our p's become very large, the amount of data that we can use for our prediction is very small, and then our model may become very biased with a small amounf of observations. e*.To create a p-dimensional hypercube centered around the test observation that contains on average 10% of the training observations, the length of each side of the hypercube is going to be .1, assuming that the X, the predictor is uniformly distributed on [0,1]

Exercise 6

Variables: X_1 = hours studied, X_2 = undergrad GPA, and Y = receive an A. After fitting logistic regression, $\hat{\beta}_0$ = -6, $\hat{\beta}_1$ = .05, $\hat{\beta}_2 = 1$. a. If a student studies for 40 hours and has a GPA of 3.5, P(getting an A)?

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_1}}{1 + e^{\beta_0+\beta_1 X+\beta_2 X_2}}$$

$$P(X) = \frac{e^{-6+(.05\times40)+(1\times3.5)}}{1 + e^{-6+(.05\times40)+(1*3.5)}}$$

```
e <- exp(1)
b0 <- -6
b1 <- .05
b2 <- 1

probtest <- e^(b0 +(b1*40) +(b2*3.5)) / (1 +e^(b0 +(b1*40) +(b2*3.5)))
probtest
```

```
## [1] 0.3775407
```

```
probtest50 <- (log(1) - b0 - (b2*3.5))/b1
probtest50
```

```
## [1] 50
```

```
e^(b0 +(b1*50) +(b2*3.5)) / (1 +e^(b0 +(b1*50) +(b2*3.5)))
```

```
## [1] 0.5
```

a. The probability of a student with an undergrad GPA of 3.5 and studying for 40 hours getting an A in the test is .3775.

b. This student should study for 50 hours in order to to have a 50% chance of getting an A in the test.

Exercise 7

We want to predict if a stock will issue a dividend this year (Yes or No) based on X, last year's percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend (Yes) was $\bar{X} = 10$, while the mean for those that didnt was $\bar{X} = 0$. The variance of X for these two sets of companies was $\hat{\sigma}^2 = 36$. 80% of companies issued dividends. Assuming X follows a normal distribution, predict probability that a company will issue a dividend this year given that its percentage profit was 4 last year. Recall that the density function for a normal random variable is $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-u)^2/2\sigma^2}$ Use bayes theorem

$$\bar{X}_Y = 10$$

$$\bar{X}_N = 0$$

$$\hat{\sigma}^2 = 36$$

$$\hat{\pi}_Y = .8$$

$$\hat{\pi}_N = .2$$

$$X = 4$$

$$f_Y(x) = \frac{1}{\sqrt{2 \times (.8 * 36)}} e^{-(4-10)^2/2(36)^2}$$

$$f_N(x) = \frac{1}{\sqrt{2 \times (.2 * 36)}} e^{-(4-0)^2/2(36)^2}$$

$$P(Y = "yes" | X = 4) = \frac{f_Y(x)\pi_Y}{f_Y(x)\pi_Y + f_N(x)\pi_N} \tag{1}$$

$$P(Y = "yes" | X = 4) = \frac{\frac{1}{\sqrt{2 \times (.8*36)}} e^{-(4-10)^2/2(36)^2} * .8}{(\frac{1}{\sqrt{2 \times (.8*36)}} e^{-(4-10)^2/2(36)^2} * .8) + (\frac{1}{\sqrt{2 \times (.2*36)}} e^{-(4-0)^2/2(36)^2} * .2)} \tag{2}$$

```r
uy <- 10
un <- 0
piy <- .8
pin <- .2
sigma2 <- 36
x <- 4

f_ywp <- (exp(-(x-uy)^2/(2*sigma2)))/ sqrt(2*piy*sigma2)
f_nwp <- (exp(-(x-un)^2/(2*sigma2)))/ sqrt(2*pin*sigma2)

probdividend <- f_ywp / (f_ywp + f_nwp)
probnodividend <- f_nwp / (f_ywp + f_nwp)
probdividend
```

```
## [1] 0.2746962
```

```
probnodividend
```

```
## [1] 0.7253038
```

Probability that this company will issue a dividend : .4