Homework Assignment 2

Trevor Klar and Blaine Quackenbush

October 27, 2021

Linear regression (12 pts)

In this problem, we will make use of the *Auto* data set, which is part of the ISLR package and can be directly accessed by the name `Auto` once the `ISLR` package is loaded. The dataset contains 9 variables of 392 observations of automobiles. The qualitative variable **origin** takes three values: 1, 2, and 3, where 1 stands for American car, 2 stands for European car, and 3 stands for Japanese car.

```
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

Here we just remind ourselves how `origin` is coded:

```
# Origin
# 1 = American
# 2 = European
# 3 = Japanese
Auto$origin <- factor(Auto$origin,
                      levels = c(1,2,3),
                      labels = c("American", "European", "Japanese")
                      )
#Auto$origin <- as.factor(Auto$origin)
```

1. (2 pts) Fit a linear model to the data, in order to predict mpg using all of the other predictors except for name. Present the estimated coefficients. (2 pts) With a 0.01 threshold, comment on whether you can reject the null hypothesis that there is no linear association between mpg with any of the predictors.

Here we fit a linear model to the data, using all variables except `name` as predictors for `mpg`. We will also consider, with a 0.01 threshold, whether there is a statistically significant linear association between mpg and any of the predictors.

```r
auto.lmfit <- lm(mpg ~ cylinders + displacement + horsepower + weight
                 + acceleration + year + origin, Auto) # Fit a linear model.
summary(auto.lmfit)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = Auto)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.009 -2.078 -0.098  1.986 13.361
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.80e+01   4.68e+00   -3.84  0.00014 ***
## cylinders      -4.90e-01   3.21e-01   -1.52  0.12821
## displacement    2.40e-02   7.65e-03    3.13  0.00186 **
## horsepower     -1.82e-02   1.37e-02   -1.33  0.18549
## weight         -6.71e-03   6.55e-04  -10.24  < 2e-16 ***
## acceleration    7.91e-02   9.82e-02    0.81  0.42110
## year            7.77e-01   5.18e-02   15.01  < 2e-16 ***
## originEuropean  2.63e+00   5.66e-01    4.64  4.7e-06 ***
## originJapanese  2.85e+00   5.53e-01    5.16  3.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.31 on 383 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.821
## F-statistic:  224 on 8 and 383 DF,  p-value: <2e-16
```

Note that the F-statistic is quite large (224), and indeed the p-value associated with this F-statistic is is less than 2×10^{-16}. This is much smaller than 0.01, so we conclude (with 99% certainty) that there is a linear relationship between `mpg` and at least one of these variables.

2. (2 pts) Take the whole dataset as training set. What is the training mean squared error of this model?

```r
MSE <- function(model) {
  mean(residuals(model)^2)
}

MSE(auto.lmfit)
```

```
## [1] 10.68
```

3. (2 pts) What gas mileage do you predict for an European car with 4 cylinders, displacement 122, horsepower of 105, weight of 3100, acceleration of 32, built in the year 1991? (Be sure to check how year is coded in the dataset).

```r
# Origin
# 1 = American
# 2 = European
# 3 = Japanese
predict(auto.lmfit,
        data.frame(cylinders = 4, displacement = 122, horsepower = 105,
                   weight = 3100, acceleration = 32, year = 91, origin = "European"))
```

```
##     1
## 36.17
```

4. (1 pts) On average, holding all other covariates fixed, what is the difference between the mpg of a Japanese car and the mpg of an American car? (1 pts) What is the difference between the mpg of a European car and the mpg of an American car?

```r
# Origin
# 1 = American
# 2 = European
# 3 = Japanese

auto.lmfit
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = Auto)
##
## Coefficients:
##    (Intercept)       cylinders    displacement      horsepower          weight
##      -17.95460        -0.48971         0.02398        -0.01818        -0.00671
##   acceleration            year  originEuropean  originJapanese
##        0.07910         0.77703         2.63000         2.85323
```

As we can see, the coefficient of `originJapanese` is 2.85323, so a Japanese car will have 2.853 better MPG on average than an American car, and a European car will have 2.63 better MPG on average than an American car.

5. (2 pts) On average, holding all other predictor variables fixed, what is the change in mpg associated with a 10-unit increase in displacement?

0.2398 mpg.

Algae Classification using Logistic regression (15 pts)

Get the dataset `algaeBloom.txt` from the homework archive file, and read it with the following code:

```r
algae <- read_table("algaeBloom.txt",
                    col_names=c('season','size','speed','mxPH','mnO2','Cl',
                                'NO3','NH4','oPO4','PO4','Chla','a1','a2','a3',
                                'a4','a5','a6','a7'),
                    na="XXXXXXX")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mnO2 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double(),
##   a2 = col_double(),
##   a3 = col_double(),
##   a4 = col_double(),
##   a5 = col_double(),
##   a6 = col_double(),
##   a7 = col_double()
## )
```

```
head(algae)
```

```
## # A tibble: 6 x 18
##   season size  speed  mxPH  mnO2    Cl   NO3   NH4  oPO4   PO4  Chla    a1    a2
##   <chr>  <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 winter small medi~  8      9.8  60.8  6.24 578   105   170    50     0     0
## 2 spring small medi~  8.35   8    57.8  1.29 370   429.  559.    1.3   1.4   7.6
## 3 autumn small medi~  8.1   11.4  40.0  5.33 347.  126.  187.   15.6   3.3  53.6
## 4 spring small medi~  8.07   4.8  77.4  2.30  98.2  61.2 139.    1.4   3.1  41
## 5 autumn small medi~  8.06   9    55.4 10.4  234.   58.2  97.6  10.5   9.2   2.9
## 6 winter small high   8.25  13.1  65.8  9.25 430    18.2  56.7  28.4  15.1  14.6
## # ... with 5 more variables: a3 <dbl>, a4 <dbl>, a5 <dbl>, a6 <dbl>, a7 <dbl>
```

In homework 1, we investigated basic exploratory data analysis for the `algaeBloom` dataset. One of the explaining variables is `a1`, which is a numerical attribute. Here, after standardization, we will transform `a1` into a categorical variable with 2 levels: high and low, and conduct its classification using those 11 variables (i.e. do not include a2, a3, . . . , a7).

We first improve the normality of the numerical attributes by taking the log of all chemical variables. After log transformation, we **impute** missing values using the median method. Finally, we transform the variable `a1` into a categorical variable with two levels: high if a1 is greater than 5, and low if a1 is smaller than or equal to 5.

```
# Improve the normality of the numerical attributes by taking the log of all
#  chemical variables.
algae.transformed <- algae %>% mutate_at(vars(4:11), ~ log(.))
# Impute missing values using the median method.
algae.transformed <- algae.transformed %>% mutate_at(vars(4:11),
                                                     ~ ifelse(is.na(.),median(.,na.rm=TRUE),.))
# a1
```

```r
# 0 means "low"
# 1 means "high"
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))
```

Classification Task: We will build classification models to classify `a1` into `high` vs. `low` using the dataset `algae.transformed` as above, and evaluate its training error rates and test error rates. We define a new function, named `calc_error_rate()`, that will calculate misclassification error rate.

```r
calc_error_rate <- function(predicted.values, true.values){
  # Here predicted.values and true.values are lists of predictions, and
  #  true.values!=predicted.values is a list of 1s and 0s according to whether the
  #  values match.
  return(mean(true.values!=predicted.values))
}
```

Training/test sets*: Split randomly the data set in a train and a test set:

```r
# For reproducability
set.seed(1)
# Choose 50 random observations from from algae for training.
test.indices = sample(1:nrow(algae.transformed), 50)
# Split the data set into a training set and a test set
algae.train=algae.transformed[-test.indices,]
algae.test=algae.transformed[test.indices,]
```

In a binary classification problem, let p represent the probability of class label "1", which implies that $1 - p$ represents probability of class label "0". The *logistic function* (also called the "inverse logit") is the cumulative distribution function of logistic distribution, which maps a real number z to the open interval $(0, 1)$:

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

1. (2 pts) Show that indeed the inverse of a logistic function is the *logit* function:

$$z(p) = \ln\left(\frac{p}{1-p}\right)$$

Proof: Observe that the logit and logitic functions compose to form the identity:

$$\begin{aligned}
p \circ z(p) &= \exp\left(\ln\left(\frac{p}{1-p}\right)\right) \div \left(1 + \exp\left(\ln\left(\frac{p}{1-p}\right)\right)\right) \\
&= \left(\frac{p}{1-p}\right) \div \left(1 + \frac{p}{1-p}\right) \qquad (*) \\
&= \left(\frac{p}{1-p}\right) \div \left(\frac{1-p+p}{1-p}\right) \\
&= \left(\frac{p}{1-p}\right) \div \left(\frac{1}{1-p}\right) \\
&= \left(\frac{p}{1-p}\right) \cdot \left(\frac{1-p}{1}\right) \\
&= p
\end{aligned}$$

Similarly, composing in opposite order gives

$$z \circ p(z) = \ln\left[\left(\frac{e^z}{1-e^z}\right) \div \left(1 + \frac{e^z}{1-e^z}\right)\right]$$

and the argument of this expression is of the form $(*)$, yielding

$$= \ln[e^z]$$
$$= z$$

■

---

2. Assume that $z = \beta_0 + \beta_1 x_1$, and $p = \text{logistic}(z)$.

   - (2 pts) How does the odds of the outcome change if you increase x_1 by two?

   Given values for β_0, β_1, and x_1, then compute $p \circ z(x_1 + 2) - p \circ z(x_1)$. It's hairy.

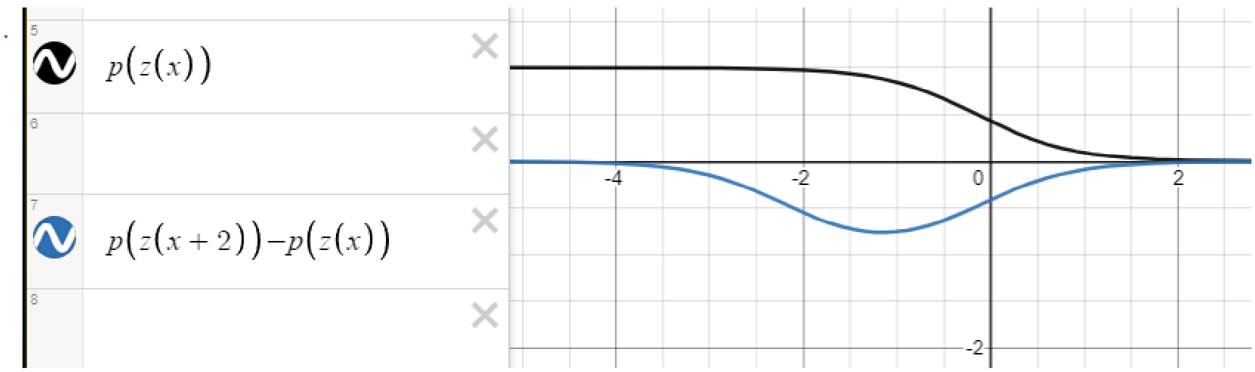

Figure 1: graph4

   - (1 pts) Assume β_1 is negative: what value does p approach as $x_1 \to \infty$?

$$\lim_{x_1 \to \infty} p \circ z(x_1) = 0$$

   - (1 pts) What value does p approach as $x_1 \to -\infty$?

$$\lim_{x_1 \to \infty} p \circ z(x_1) = 1$$

3. Use logistic regression to perform classification in the data application above. Logistic regression specifically estimates the probability that an observation has a particular class label. We can define a probability threshold for assigning class labels based on the probabilities returned by the `glm` fit.

In this problem, we will simply use the "majority rule". If the probability is larger than 50% class as label "1". + (2 pts) Fit a logistic regression to predict a1 given all other features in the dataset using the glm function.

```
algae.glm.fit <- glm(
  a1 ~ season + size+ speed  + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 + PO4 + Chla,
  data = algae.train,
  family = binomial
)
summary(algae.glm.fit)
```

```
##
## Call:
## glm(formula = a1 ~ season + size + speed + mxPH + mnO2 + Cl +
##     NO3 + NH4 + oPO4 + PO4 + Chla, family = binomial, data = algae.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.903  -0.638   0.122   0.577   1.981
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.3708    11.3012    0.21   0.8338
## seasonspring  -0.6582     0.7830   -0.84   0.4006
## seasonsummer   0.8865     0.8420    1.05   0.2924
## seasonwinter   0.6159     0.6773    0.91   0.3631
## sizemedium     0.6043     0.7567    0.80   0.4245
## sizesmall      1.9198     0.8672    2.21   0.0268 *
## speedlow       1.4404     0.8468    1.70   0.0889 .
## speedmedium    0.0774     0.6157    0.13   0.8999
## mxPH          -0.2468     5.4101   -0.05   0.9636
## mnO2           1.1671     0.9187    1.27   0.2039
## Cl            -0.3636     0.3765   -0.97   0.3342
## NO3           -0.1568     0.3718   -0.42   0.6732
## NH4            0.3828     0.2629    1.46   0.1453
## oPO4          -0.9784     0.4817   -2.03   0.0422 *
## PO4           -0.1558     0.5856   -0.27   0.7902
## Chla          -0.8376     0.2892   -2.90   0.0038 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 202.69  on 149  degrees of freedom
## Residual deviance: 113.73  on 134  degrees of freedom
## AIC: 145.7
##
## Number of Fisher Scoring iterations: 6
```

```
+ (2 pts) Estimate the class labels using the majority rule
```

```r
algae.train.predicted <- predict(algae.glm.fit, type = "response") %>% round
algae.test.predicted <- predict(algae.glm.fit, algae.test, type = "response") %>% round
```

```
+ (2 pts) calculate the training and test errors using the calc_error_rate defined earlier.
```

```r
calc_error_rate(algae.train.predicted, algae.train["a1"])
```

```
## [1] 0.2
```

```r
calc_error_rate(algae.test.predicted, algae.train["a1"])
```

```
## [1] 0.5267
```