STATISTICS 452/652: Statistical Learning and Prediction

Lecture 4: Extensions of regression variables and feature engineering

(Reading: ISLR 3.3)

1 Goals of lecture

- Some measurements are not numerical
  - Often, measurements are *categorical*, meaning that the response is a word label rather than a number
    - Sex
    - Gender
    - Political party
    - SFU Major Program
  - May still be useful in prediction
  - Need to understand how to handle these in regression
- Before we try to use data to identify which variables are important, we should try to represent information in the variables in the best ways possible.
  - Sometimes (often?) variables influence the population structure $g(\mathbb{X})$ in more complicated ways than linear
  - Sometimes (often?) effects of a variable $g(\mathbb{X})$ are modified by other variables
  - Sometimes (often?) the variables can be combined to form better, more effective predictors
- This lecture reviews how to handle these issues in the context of linear regression
  - Ideas apply to other forms of regression, and to classification

2 Categorical variables

- Data that are measured as word labels rather than as numbers are called CATEGORICAL VARIABLES
- Measurement is the name of a category
  - Female/Male
  - STAT/ACMA/DATA/ECON/CMPT/etc.
- We now need to address how categorical explanatory variables are handled, both mathematically and in R
  - Can't fit a model treating levels as numerical

2.1 Binary indicator/dummy variables

For the moment, assume that there is only one variable, X

- Suppose X is categorical with Q levels
- All observations at the same level of X have the same mean Y
  - Observations at different levels *may* have different means
- We can label the levels from $1, \ldots, Q$, but these are just labels, can't treat them as numerical
  - Could be alphabetical, or by some other ordering, or random
  - But need *something* numerical that we can do regression on
- SOLUTION: Create Q binary INDICATOR (DUMMY) variables, $X_1, X_2, \ldots, X_Q$
  - Each $X_q,\ q = 1, \ldots, Q$ is 1 if the observed value of X is the qth level, and 0 if not
  - "Indicates" the qth level
  - Called "one-hot encoding" in computing science and machine learning
- We can try putting all Q indicators into a linear model
  - Instead of fitting nonsensical $f(X) = \beta_0 + \beta_1 X$, fit model

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_Q X_Q$$

  :

  - Notice that when you plug in the 0-1 values of each x_q into this model, something simple comes out ($Q = 4$ here)

	Indicator Variable				
X-level	X_1	X_2	X_3	X_4	$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$
1	1	0	0	0	$f(1) = \beta_0 + \beta_1$
2	0	1	0	0	$f(2) = \beta_0 + \beta_2$
3	0	0	1	0	$f(3) = \beta_0 + \beta_3$
4	0	0	0	1	$f(4) = \beta_0 + \beta_4$

  - Each level of X has potentially a different mean
    - Depends on true values of $\beta_1, \ldots, \beta_Q$

- This particular model has problems, however
  - With Q groups, there should be Q means, one for each level
  - This model has $Q + 1$ parameters—it is OVERSPECIFIED and there are infinitely many solutions
    - e.g., $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) = (0, 1, 2, 3, 4)$ yields exactly the same results as $(1, 0, 1, 2, 3)$ or $(4, -3, -2, -1, 0)$ or...
  - LS estimation may fail with an error, depending on software
    - `lm()` in R seems to still work when you feed it all Q indicators in a model, but it refuses to provide an estimate for the last indicator

- Fix the problem by *dropping one of the indicators*
  - Then have Q parameters to estimate Q means
    - Mathematically, there is one unique solution to the problem
  - The dropped level becomes the "baseline" level (see below)
  - *Equivalent to forcing the parameter to be 0 in the overspecified model*

- Which one to drop?
  - Mathematically, is doesn't matter; pick any and adjust interpretations accordingly
  - R stores categorical data as `factor`-class objects
    - It will automatically drop the first indicator, X_1, when you put a `factor`-class explanatory variable in a `formula`
  - Which one is X_1?
    - R stores factor levels in *alphanumeric order* according to the level labels.
    - To see the ordering of levels in any factor, use `levels(<factor name>)`
    - Various R commands can reorder the levels if you would rather use a different level as baseline

- Here is the regression that R fits with `lm(Y~X)` when $X$is a `factor`:

	Indicator Variable			
X-level	X_2	X_3	X_4	$f(X) = \beta_0 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$
1	0	0	0	$f(1) = \beta_0$
2	1	0	0	$f(2) = \beta_0 + \beta_2$
3	0	1	0	$f(3) = \beta_0 + \beta_3$
4	0	0	1	$f(4) = \beta_0 + \beta_4$

- From this, you can interpret the parameters:
  - β_0 is the mean Y *at the first level of* X
    - This is the "baseline" level against which others are compared.
  - For $q = 2, \ldots, Q$, $\beta_q = f(q) - f(1)$,
- For convenience of notation, we can sometimes pretend that we are using the original model including a $\beta_1 X_1$ term, and that $\beta_1 = 0$.

Example: Categorical variables in simulated data (L4 - Extensions of Variables.R) In this example, we create some data with properties we understand so that we can see how R reacts to the properties. Specifically

- `X1` is a categorical factor with 4 levels, `A`, `B`, `C`, `D`, with five observations at each level
- `X2` is numerical
- `y` has a mean that changes for different values of `X1`—means 3, 1, 5, and 1, respectively—but is not related to `X2`

We create the data and explore regressions in `lm()`. See the program for more details.

First, the data creation and a display of the data:

```
> # Creating some data, saving to data frame, "dat"
> set.seed(3894270)
> X1 = rep(x=c("A", "B", "C", "D"), each=5)
> X2 = round(rnorm(20),2)
> mean.y = rep(c(3,1,5,1), each=5)
> epsilon = rnorm(20)
> y = round(mean.y + epsilon, 2)
> dat = data.frame(y,X1,X2)
>
> cbind(dat[1:5,],dat[6:10,],dat[11:15,],dat[16:20,])
     y X1    X2    y X1    X2    y X1    X2    y X1    X2
1 3.47  A  0.83 0.20  B -2.15 5.64  C  0.03 3.45  D -0.38
2 3.64  A -0.77 1.43  B  0.91 6.38  C -0.86 1.22  D  0.43
3 3.50  A -1.41 1.26  B  0.86 5.81  C -0.58 1.69  D  1.14
4 2.25  A  0.33 1.13  B -0.65 2.62  C -0.90 1.17  D  0.17
```

```
5 4.39  A  0.18 0.63  B  0.51 4.84  C -2.80 0.71  D  1.12
>
> # Here is how to see the information in the variable X1
> class(dat$X1)
[1] "factor"
> levels(dat$X1)
[1] "A" "B" "C" "D"
```

Next, we show the regression model that `lm(y~X1)` fits. Since the four factors are stored as "`A`", "`B`", "`C`", "`D`", `lm()` creates indicators for each level, say x_A, x_B, x_C, x_D. It runs the regression model on all but the first, fitting $f(X) = \beta_0 + \beta_B x_B + \beta_C x_C + \beta_D x_D$, where I have just used factor level names instead of numbers to label the variables and coefficients. See below.

```
> mod.cat = lm(y ~ X1, data=dat)
> summary(mod.cat)

Call:
lm(formula = y ~ X1, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4380 -0.4405  0.0460  0.5205  1.8020

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.4500     0.4551   7.581 1.11e-06 ***
X1B          -2.5200     0.6436  -3.916  0.00123 **
X1C           1.6080     0.6436   2.498  0.02375 *
X1D          -1.8020     0.6436  -2.800  0.01284 *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.018 on 16 degrees of freedom
Multiple R-squared:  0.7573,    Adjusted R-squared:  0.7118
F-statistic: 16.65 on 3 and 16 DF,  p-value: 3.554e-05
```

Note that the four estimated group means are

	Indicator Variable			
X-level	x_B	x_C	x_D	$f(X) = \beta_0 + \beta_B x_B + \beta_C x_C + \beta_D x_D$
A	0	0	0	$\hat{f}(A) = \hat{\beta}_0 = 3.5$
B	1	0	0	$\hat{f}(B) = \hat{\beta}_0 + \hat{\beta}_B = 3.5 - 2.5 = 1.0$
C	0	1	0	$\hat{f}(C) = \hat{\beta}_0 + \hat{\beta}_C = 3.5 + 1.6 = 5.1$
D	0	0	1	$\hat{f}(D) = \hat{\beta}_0 + \hat{\beta}_D = 3.5 - 1.8 = 1.7$

These results are close to the true means, 3, 1, 5, and 1.

- Now suppose that there is an additional numeric variable, say Z, and you want to use both variables in a regression.
- Just add it to the model,

$$f(X, Z) = \beta_0 + \beta_2 X_2 + ... + \beta_Q X_Q + \beta_z Z$$

  - For level q of X, $f(q, Z) = (\beta_0 + \beta_q) + \beta_z Z$
  - This is just a linear regression with intercept $(\beta_0 + \beta_q)$ and slope β_z, where $\beta_q = 0$ for $q = 1$
  - In other words, the model fits Q linear regressions with Q intercepts but a common slope.
  - In R, nothing special is needed
    - `formula=Y~X+Z` for X a `factor` and Z any form of numeric

3 Feature Engineering

Now return to the case where X is numeric.

3.1 Transformations of variables

- Sometimes the model with linear explanatory variables does not fit well
  - We have seen examples where a polynomial model in X is better than a straight line
  - Polynomials are a special case of a general notion called transformations
- A TRANSFORMATION of a numerical variable X is just some function $h(X)$ that creates a non-linear form
  - Polynomials are special cases with $h(X) = X^d$ for some order d
  - Other examples include
    - square root, $h(X) = \sqrt{X}$
    - log, $h(X) = \log(X)$ (we always use base e natural logs here)

    - inverse, $h(X) = 1/X$
  - Basically any function that provides a one output for each input is fine
    - Although not often recommended, can include transforming numeric into categorical, such as
      - If $X \leq a$ then $h(X) =$ "A"
      - If $a < X \leq b$ then $h(X) =$ "B"
      - If $X > b$ then $h(X) =$ "C"

- Goal of a transformation is that Y should have a shape that is better explained by $h(X)$ than by X

- There are R functions to compute many transformations, like `log()`, `sqrt()`
  - Others can be computed within `formula` in `lm()`.
  - Or can create new variables and add them to data set
  - Some details and example are in the R program for this lecture.

- In SL, we often fit models that allow the shape of $f(X)$ to adapt to data
  - So don't often need to use transformations of X, but in linear regression they can be very helpful.

3.2 Interactions

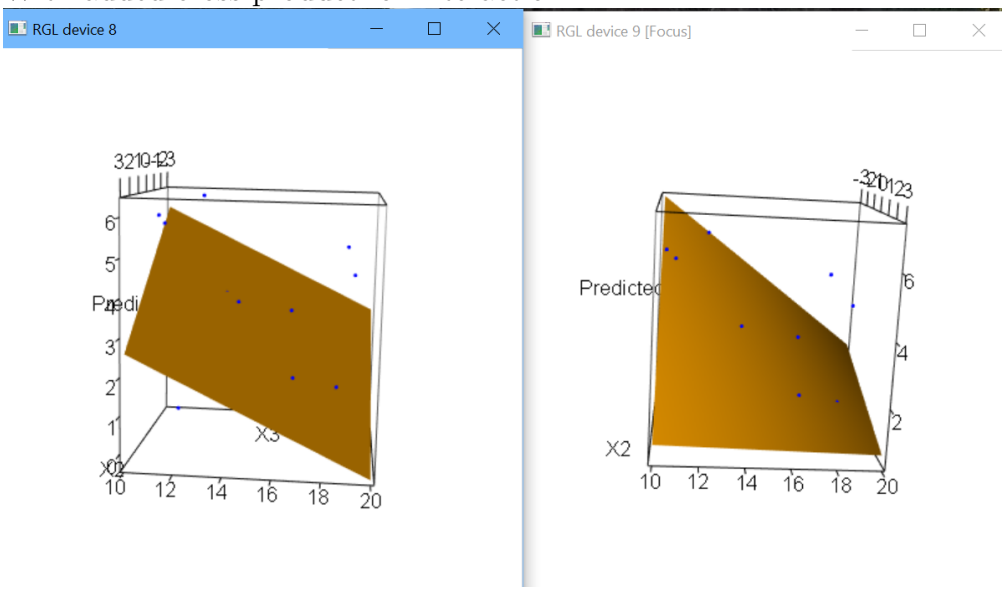
- Sometimes the effect of one explanatory variable on response Y changes depending on the value of another explanatory variable
  - Relationship between sodium intake and blood pressure may depend on weight
  - Relationship between temperature and ozone may depend on wind speed

- This is called an INTERACTION between the two explanatory variables.

- An interaction effect in a linear regression occurs when the slope of one variable changes according to the value of another variable
  - A very simple *model* for interaction is to assume that the slope changes linearly with the other variable
  - That is, say, SLOPE$(X_1) = \gamma_0 + \gamma_1 X_2$
  - This is achieved by adding a CROSSPRODUCT, X_1X_2, to the linear regression with X_1 and X_2
  - Then $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$
    - If we hold X_2 fixed at some value x_2 and increase X_1 by 1 unit, then $f(X)$ changes by

      $$[\beta_0 + \beta_1(x_1 + 1) + \beta_2 x_2 + \beta_3(x_1 + 1)x_2] - [\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2] = \beta_1 + \beta_3 x_2$$

Figure 1: Linear Regression model in two variables. Left: Only the two linear effects. Right: With added cross-product for interaction



* So the slope on X_1 changes linearly as X_2 changes.
* The same relationship holds the other way: the X_2 slope changes linearly as X_1 changes.

Example: Interaction in simulated data (L4 - Extensions of Variables.R) Continuing the previous example, we randomly generate some data for a third variable, `X3`. We then fit two models:

- $f(X) = \beta_0 + \beta_1 X_2 + \beta_2 X_3$ using `lm(y~X2 + X3 , data=dat)`
- $f(X) = \beta_0 + \beta_1 X_2 + \beta_2 X_3 + \beta_3 X_2 X_3$ using `lm(y~X2 + X3 + X2:X3, data=dat)`

See the program for details. The resulting surfaces are shown in Figure 1. The model on the left is a flat plane with only the linear effects of each variable. The slope from left to right (`y` vs. `X3`) is constant regardless of the level of the third variable (`X2`, representing the "depth" dimension in the plot). The model on the right includes the cross-product. The image is a "twisted plane": the relationship between `y` and `X3` is still linear as long as `X2` is held fixed, but the slope is different fro each value of `X2`.

3.3 Other functions of multiple variables

- Transformations and interactions are examples of FEATURE ENGINEERING, the process of creating new variables from existing variables

- LOTS of other ways to create new variables that may measure important concepts better than single variables
  - For example, instead of a crossproduct, we might use a ratio between two variables
    - Body Mass Index, BMI, a common surrogate for body fat, is calculated from Weight and Height as W/H^2
    - In a 10-minute exercise study, might measure heart rate every 10 seconds.
      - This creates 60 heart-rate measurements
      - Maybe most relevant function is the maximum, or the maximum over 5 consecutive measurements, or the average or...
- Feature engineering *can* be done by the data analyst,
  - Careful analysis and luck sometimes reveals new and useful relationships
- But usually it is better to involve the subject matter expert
  - They can determine what combinations of variables make sense

4 What to take away from this

- There are numerous options for extending the linear regression model to
  - adapt to categorical variables
  - create different shapes
  - allow variables to have interactive effects on the mean of Y
- Even with these, linear regression is still limited in its ability to adapt to different shapes.

5 Exercises

Concepts

1. Suppose you fit a model, $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 (X_1/X_2)$
   (a) For a fixed value of $X_2 = c$, what is the "slope" of X_1 (i.e., how much does $f(X)$ change for a 1-unit change in X_1)? **Write the answer in terms of parameters and c.**
   (b) For a fixed value of $X_1 = d$, what is the "slope" of X_2 (i.e., how much does $f(X)$ change for a 1-unit change in X_2)? **Write the answer in terms of parameters and d.**
2. Suppose that X is numerical and Z is categorical (factor) with two levels. Someone has shown you a model where they fit `lm(Y~X + Z + X:Z)`. **Write the model that R fits as a single regression model of the form $f(x) = ...$** Use variables $z_q, q = 1, \ldots, Q$, to represent indicators for level q of Z

Application

A. Feature Engineering

Refer to the Air Quality data described previously. We will add some new features to the data and repeat the CV model-selection process as in the previous assignment.

I theorize that there is an interaction relationship between `Temp` and `Wind` on `Ozone`. I am not sure how it works though:

- Maybe it works to model the `Temp` slope as changing linearly over levels of `Wind`, in which case adding a cross-product term to the model, `Temp*Wind`, will work.
- Maybe high temp and high wind have similar ozone as low temp and low wind, but when only one of these is high, the ozone can change quite a bit. This suggests adding a ratio `Temp/Wind` to the model.

Both of these calculations can be performed within the `lm()` function, but I want to examine some plots first. So we will create two new variables:

- `AQ$TWcp = AQ$Temp*AQ$Wind`
- `AQ$TWrat = AQ$Temp/AQ$Wind`

where `AQ` is what I have called the data frame that we created previously for this problem.

1. Compute a `summary` on `TWcp` and `TWrat`. **Report the minimum, maximum, and mean for each variable.**
2. Create two new models: `Temp + Wind + TWcp` and `Temp + Wind + TWrat`. Fit these two models in `lm()`.
   (a) **Report the t-test results for the two new variables**.
   (b) Based on the test results, which variable seems to be the most useful, or are neither particularly helpful? **(1 sentence)**
   (c) From the model with the cross-product term, compute and **report the slope of the `Temp` effect when `Wind` is at its minimum value. Repeat for the maximum value of `Wind`.** (You can do this by hand from the output if you want.)
3. Fit each model on the training data and **report the MSPEs from the validation data**.
   (a) **Which model wins this competition?**
4. Add these models the five you compared in the previous exercise, and rerun the CV 20 times.
   (a) **Make boxplots of the RMSPE, and narrow focus if necessary to see best models better.**
   (b) Are any of the new models competitive, or even best? **(1 sentence)**

B. Categorical Explanatories

Refer to the Insurance data in the Canvas module. The data consist of 2182 observations of insurance claim data from Sweden in the 1970s. I am certain that you are excited by thrill of studying 45-year-old insurance claims, but, hey, it's data.

The data are also available within R in the `faraway` package (https://rdrr.io/cran/faraway/man/motorins.h
A description of the data is here: http://www.statsci.org/data/general/motorins.html. I suggest you use my version of it, in case there are any differences in sets. The response variable is in the first column, `per`, representing the average payment per claim. The other variables are explanatory.

I suggest you read in the data and look at it. Make some plots, check correlations, compute summaries, etc. Note that, although all data are numerical in the file, *there are actually two categorical variables*, `zone` and `make`. These will need to be converted to factors in R and then handled according to how each different analysis method deals with categorical variables. Specifically, some can handle categorical factors directly, while others will require that they be converted to numerical indicators ("hot one encoding"). As a first step, we need to convert them to factors. This can be done with the code below:

```
ins = read.csv(..., header=TRUE)
ins$zone = as.factor(ins$zone)
ins$make = as.factor(ins$make)
```

Also note that there are some observations for which 0 claims were made, and so the response does not make sense. (An alternative use for these data is to do classification and predict whether any claims are made.) We need to delete these observations prior to doing regression modeling, leaving 1797 full observations for modeling. You can use `ins = ins[ins$claims>0,]` to do this.

Finally, I must admit that I have no idea what unit one "observation" was measured on. It seems like it must be an insurance agent or some other aggregate over many customers, because there are way too many claims than can be explained if each observation is one customer...unless Swedes in the 1970s were *horrifically poor drivers*!

1. Once `zone` and `make` have been converted to factors, run the linear regression with `per` as the response and the other six variables as explanatory.
   (a) Create a summary of the `lm` object.
      i. Although you fit a model with 6 variables, **how many parameters are estimated**?
      ii. **What is the intercept of the regression when `make` and `zone` are both at their first level, 1**?
      iii. **What is the intercept of the regression when `make` and `zone` are both at their last levels,** 9 and 7, respectively?