POLI SCI 490: Machine Learning & Text-as-Data

HW1

Sarah B. Bouchat

DUE 31 January 2018

1 Tidyverse!

```
library(dplyr)
library(ggplot2)
library(broom)
```

Packages in R return output in varying formats. Using the `mtcars` data, compare `coef()` for `glm` and `glmnet` objects, which return very different structures:

```
glm(mpg ~ ., data = mtcars) %>% coef()

glmnet::glmnet(x = as.matrix(mtcars[, -1]), y = mtcars$mpg,
               alpha = 1, lambda = 1) %>% coef()
```

Using the `tidy` function in the `broom` package, you can get your data back in a simple dataframe. Which columns are present depends on the model, but their names will be consistent: the name of the coefficient will always be `terms`, its estimate will always be `estimate`, etc. This makes writing code much easier.

```
glm(mpg ~ ., data = mtcars) %>% broom::tidy()

glmnet::glmnet(x = as.matrix(mtcars[, -1]), y = mtcars$mpg,
               alpha = 1, lambda = 1) %>% broom::tidy()
```

1.1 Tidytext vs. tm practice

2 Regression

The `sick_data.csv` file has data on the results of a medical test for a particular disease among 1,000 people. Their temperatures and blood pressures have also been recorded. Because the test is expensive, you would like to be able to predict whether or not people are sick only based on their temperature and blood pressure.

Do each of the following steps using OLS, logit, and ridge regression in turn. (*Hint*: Use $\lambda = 1$ for the ridge regression.)

a. Estimate a regression $y \sim x_1 + x_2 + \epsilon$ and display the results.
b. Calculate predicted values $\hat{y}$ for the observations in the data. Say $\hat{y} \geq 0.5$ is a prediction that someone will test positive for the disease, and $\hat{y} < 0.5$ is a prediction that they will test negative. How well does the regression predict the test results?
c. Using the regression results, calculate the equation of the line where $\hat{y} = 0.5$ as a function of blood pressure and temperature.
d. Display the blood pressure and temperature data on a single scatterplot, using either color or shape to distinguish between positive and negative results. Add the line you calculated from the previous step. (*Hint*: Use `ggplot2`!)

3 Regularization/Selection

You manage a widget factory and are trying to assess the performance of your widgets (`y`). Your engineering team has sent you a lot of data from the assembly line (`x1`, `x2`, etc.), but it's not clear how much of it is relevant for the outcome you want to measure. Because of the large number of variables, you also cannot be certain of the linearity of their relationship to widget performance. You decide to fit a ridge regression and a lasso to evaluate these data.

Complete the following tasks using `widget_data.csv`.

(*Hint*: Chapter 6 of ISL has useful example code for this part.)

a. Load the data, and plot the dependent variable `y`.
b. Use `glmnet()` from the `glmnet` package to estimate a ridge regression with a sequence of λ from $\frac{1}{100}$ to 100.
c. Use `tidy` from the `broom` package to extract the data from the regression into a useable format and use `ggplot2` to plot the coefficient estimates as λ changes.
d. Use cross validation with `cv.glmnet` to pick the value of λ that will minimize mean squared error, and give the coefficients you get when using that λ.
e. Repeat the above steps for lasso (with $\alpha = 1$) instead of ridge.
f. Discuss the differences—what do you find?

4 Classification

Complete the following using the `e1071` package as suggested in ISL.

You are building a model to assess underlying party affiliation of unregistered voters across neighborhoods in a city. Will these individuals lean toward the Socialcrats or Politicalists? You have the following data:

- Politicalist vote margin in the neighborhood in the previous election,
- percent of people in the neighborhood with college degrees, and,
- household income.

Complete the following first using Naive Bayes and then Support Vector classification:

a. Split the data into $\frac{2}{3}$ training and $\frac{1}{3}$ test data.
b. Estimate each model using the training data only.
c. Use the model to predict the outcome in the test data.
d. Create a table of the predicted classes against the real classes.