Homework 2

Advanced Methods for Data Analysis (36-402/36-608)

Due Tues February 4, 2014 at 11:59pm

Instructions: each homework will be submitted in two parts.

1. *Writeup.* An important part of the learning the data analysis trade is learning how to communicate. Prepare a writeup to the questions below and work hard to make your submission as readable as you possibly can—this means no raw R code as part of the writeup, unless you find it necessary for clarity; no raw R output with 12 significant digits, etc., unless again you deem it necessary; making sure figures are of proper (large) size with labeled, readable axes; and so forth. Your submitted writeup must be in PDF format. You can produce this PDF using LaTeX, Word, or whatever you want, but at the end of the day, it must be a PDF. Any other type of file will not be accepted.
2. *Code.* Also submit your R code for the homework as a plain text file. You should demarcate sections of the code that correspond to the different homework problems using clearly visible comments (e.g., as in `##### Problem 1 #####`).

1 Oh training error, so young and naive

In this problem you'll investigate the optimism of training error in linear regression. We are given training samples (x_i, y_i), $i = 1, \dots n$, and test samples (x'_i, y'_i), $i = 1, \dots m$ which we'll assume are all i.i.d., i.e., these points are all independent and come from the same distribution.

Assuming that the predictors are p-dimensional, let $\hat{\beta} \in \mathbb{R}^p$ denote estimated linear regression coefficients on the training data,

$$\hat{\beta} = (x^T x)^{-1} x^T y,$$

where x is the $n \times p$ matrix with ith row equal to x_i, and y is an n-dimensional vector with ith component y_i. We're going to prove that

$$\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^n (y_i - \hat{\beta}^T x_i)^2\Big] \leq \mathbb{E}\Big[\frac{1}{m}\sum_{i=1}^m (y'_i - \hat{\beta}^T x'_i)^2\Big],$$

where the expectations above are over all that is random, i.e., over the training set on the left hand side, and over both the training and testing sets on the right hand side. In words, we're proving that the expected training error is always less than or equal to the expected testing error (without many assumptions at all on the true model), meaning that the training error is naively optimistic.

(a) Argue that the expected test error is the same whether we have m test points or just 1 test point, i.e.,

$$\mathbb{E}\Big[\frac{1}{m}\sum_{i=1}^m (y'_i - \hat{\beta}^T x'_i)^2\Big] = \mathbb{E}\big[(y'_1 - \hat{\beta}^T x'_1)^2\big].$$

Hence argue that indeed it doesn't matter whether we have m test points or n test points,

$$\mathbb{E}\Big[\frac{1}{m}\sum_{i=1}^{m}(y_i' - \hat{\beta}^T x_i')^2\Big] = \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \hat{\beta}^T x_i')^2\Big],$$

and so we may assume without a loss of generality that $m = n$ (the testing and training sets have the same number of samples).

(b) Now it is our task to compare the sizes of $\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\beta}^T x_i)^2]$ and $\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \hat{\beta}^T x_i')^2]$. First consider the random variables

$$A = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\beta}^T x_i)^2 \quad \text{and} \quad B = \frac{1}{n}\sum_{i=1}^{n}(y_i' - \tilde{\beta}^T x_i')^2,$$

where $\tilde{\beta} \in \mathbb{R}^p$ denotes the estimated linear regression coefficients but fit on the *test set*. Argue that A and B have the same distribution, so $\mathbb{E}(A) = \mathbb{E}(B)$.

(c) Argue that the random variable B defined in part (b) is always less than or equal to the observed test error,

$$B = \frac{1}{n}\sum_{i=1}^{n}(y_i' - \tilde{\beta}^T x_i')^2 \leq \frac{1}{n}\sum_{i=1}^{n}(y_i' - \hat{\beta}^T x_i')^2.$$

(Hint: don't try to plug in a formula for $\tilde{\beta}$, just recall that it is characterized by least squares ...)

(d) Use the result of part (c) to conclude that

$$\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\beta}^T x_i)^2\Big] \leq \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}(y_i' - \hat{\beta}^T x_i')^2\Big],$$

as desired.

2 The advantages of backwardness

(credit to Cosma Shalizi)

Some theories of economic growth say that it's easier for poor countries to grow faster than rich countries—"catching up", or the "advantages of backwardness". One argument for this is that poor countries can grow by copying existing, successful technologies and ways of doing business from rich ones. But rich countries are already using those technologies, so they can only grow by finding new ones, and copying is faster than innovation. So, all else being equal, poor countries should grow faster than rich ones. A way to check this is to look at how growth rates are related to other economic variables.

We will examine data from the "Penn World Table"[1], for selected countries and years. See `penn-table.csv` on the course website. Each row of this table gives, for a given country and a five-year period, the starting year, the initial population of the country, the initial gross domestic product (GDP) per capita (adjusted for inflation and local purchasing power), the average annual growth rate of GDP over that period, the average population growth rate, the average percentage of GDP devoted to investment, and the average percentage ratio of trade (imports plus exports) to GDP.

[1] See `http://pwt.econ.upenn.edu/php_site/pwt_index.php`.

We will use the `np` package on CRAN to do kernel regression.[2] Install it, and load the data file `penn-table.csv` (linked from the course website).

(a) Fit a linear model of `gdp.growth` on `log(gdp)`. What is the coefficient? What does it suggest about catching up?

(b) Fit a linear model of `gdp.growth` on `log(gdp)`, `pop.growth`, `invest` and `trade`. What is the coefficient on `log(gdp)`? What does it suggest about catching up?

(c) It is sometimes suggested that the catching up effect only works for countries which are open to trade with, and learning from, more-developed economies. Add an interaction between `log(gdp)` and `trade` to the model from part (b). What are the relevant coefficients? What do they suggest about catching up?

(d) Use 5-fold cross-validation to decide which of these three linear models predicts best.

(Hint: look at the code in `cv_bws_npreg.R`, linked from the course website, which performs cross-validation to choose the best bandwidth value in kernel regression. Adapt it to meet your purpose here.)

(e) The `npreg` function in the `np` package does kernel regression. By default, it uses a combination of cross-validation and sophisticated but very slow optimization to pick the best bandwidth. Here, we will force it to use fixed bandwidths, and do the cross-validation ourselves. The command

```
kernfit = npreg(gdp.growth~log(gdp),data=penn,bws=0.1)
```

performs a kernel regression of `growth` on `log(gdp)`, using the default kernel (which is Gaussian) and a bandwidth of 0.1. (In the above, `penn` represents the name of the data frame saved from reading in `penn-table.csv`; you may have named this data frame something else.) You can run `fitted`, `predict`, etc., on the output of `npreg` just as you can on the output of `lm`.

The code in `cv_bws_npreg.R`, linked online, computes cross-validation error for kernel regression over a specified set of bandwidth values. Use it to compute the 5-fold cross-validation error as a function of the 20 bandwith values `bws = seq(0.1,1,length=20)`. Plot the cross-validation error curve (as a function of the bandwidth). Which bandwidth value predicts best?

(f) By calling

```
kernfit.auto = npreg(gdp.growth~log(gdp),data=penn)
```

we instruct the `npreg` function to select the bandwidth value via its own internal cross-validation mechanism. What value of the bandwidth did it choose? (You can see the selected bandwidth by, e.g., typing `summary(kernfit.auto)`.) Is it similar to the one you selected in part (e)?

(g) To choose between the best linear model, as picked by you in part (d), and the kernel regression from problem (f), use cross-validation again. Modify the cross-validation code you've used so far to get a cross-validation error for the kernel estimator (with automatic bandwidth selection). Recall that you already have a cross-validation error for the best linear model from part (d).

(h) Based on your analysis, does the data support the idea of catching up, undermine it, support its happening under certain conditions, or provide no evidence either way?

[2] Chapter 4 in the Shalizi text contains helpful examples, and also, a good tutorial for this package is available at `http://www.jstatsoft.org/v27/i05`.