Homework 1

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

Due Thursday January 23, 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 Hello kernel regression

Recall that we learned kernel regression at the end of Lecture 1. In this problem you will consider the same data set from Lecture 1, but fit kernel regression, and consider its performance. It's going to be helpful to look over the code from this first lecture carefully, in "01-intro.R".

(a) Download the file "nonlin.Rdata" from the course website, and load it into your R session, with `load("nonlin.Rdata")`. You can type `ls()` to see the R objects that have been loaded into memory.

The matrices `xtrain` and `ytrain`, are each 100×50, containing 50 training data sets of x and y points along its columns. That is, the first column of `xtrain` and the first column of `ytrain` make up a training data set of 100 x-y pairs.

Hence, amassing the data sets together, there are 5000 x points and 5000 y points in total. Plot these 5000 x points versus these 5000 y points, on a single plot, to get an idea of the trend. (Hint: there is an easy way to do this with a single call to the `plot` function.)

(b) For the next bit, we'll restrict our attention to just the first training set, i.e., the first columns of `xtrain` and `ytrain`. Using the function `ksmooth`, fit a kernel regression on these training points, with 3 different values of the bandwidth parameter: 0.01, 0.25, and 1. You should be setting the option `kernel="normal"`. For each bandwidth value, plot the estimated regression function from kernel regression over top of the training points.

(c) By inspection, what happens to the kernel regression fit as we drive the bandwidth parameter down to 0? What procedure does this remind you of (that we've already seen)? What happens as we drive the bandwidth parameter up to 1? Again, what procedure does this remind you of?

(d) Sticking with the same training set, i.e., the first columns of `xtrain` and `ytrain`, we're going to investigate our predictive performance on the first test set, i.e., the first columns of `xtest` and `ytest`. For a set of 20 bandwidth values, equally spaced between 0.01 and 1, fit a kernel regression to the training points and predict the regression function at the test x points. Evaluate its test error, measured in terms of squared error loss to the test y points. Hence, you will have a curve of 20 test errors; plot this test error curve as a function of the underlying bandwidth values.

(e) According to this test error curve, what is the optimal bandwidth value? What is its associated test error? Plot the kernel regression fit, over top of the training points, at this optimal bandwidth value. Looking at the plot, does your eye agree that this is really the best bandwidth value? Why or why not?

(f) Now repeat part (d), but do the same for each of the 50 training and test data sets in turn, and report the *average* test error curve at each bandwidth value over the 50 sets. Again, plot the test error curve with respect to the bandwidth values. What do you see now? Has the optimal value of the bandwidth changed, and has its associated test error?

Bonus: There was a big difference between the test error curve from computed from a single data set, and the test error curve averaged over 50 data sets, when we looked at k-nearest-neighbor regression in lecture. There's not as big a difference here with kernel regression. Why is this?

2 Goodbye regression assumptions

Consider arbitrary random variables $X \in \mathbb{R}^p$, $Y \in \mathbb{R}$ with absolutely no assumptions relating the two, and consider regressing Y on X (in the population), with regression coefficients

$$\beta = \mathrm{Var}(X)^{-1}\mathrm{Cov}(X, Y), \quad \beta_0 = \mathbb{E}(Y) - \beta^T \mathbb{E}(X).$$

From X, our prediction for Y is hence $\beta_0 + \beta^T X$.

(a) Define the error term $E = Y - \beta_0 - \beta^T X$. Prove that E has mean zero, $\mathbb{E}(E) = 0$.

(b) Prove that E is uncorrelated with the predictor variables, $\mathrm{Cov}(E, X) = 0$.

(c) By construction, we have the relationship $Y = \beta_0 + \beta^T X + E$, i.e., we've written Y as a linear function of X plus an error term E. This error term has mean zero by part (a). Does part (b) imply that the error term is independent of X? What in particular does this mean about the conditional variance $\mathrm{Var}(E|X)$? Need this be constant with X?

(d) Consider i.i.d. samples (x_i, y_i), $i = 1, \ldots n$, with the same distribution as (X, Y). For simplicity you may assume from now on that $\mathbb{E}(X) = \mathbb{E}(Y) = 0$ (though this is not really necessary). Use the same sample notation from Lecture 2, i.e., $y = (y_1, \ldots y_n) \in \mathbb{R}^n$ for the vector of outcomes, and

$$x = \begin{bmatrix} x_1^T \\ x_2^T \\ \ldots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times p}$$

for the matrix of predictors. Consider the linear regression estimate

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

Compute $\mathbb{E}(\hat{\beta}|x)$. Is this necessarily equal to $\beta = \mathrm{Var}(X)^{-1}\mathrm{Cov}(X,Y)$? If not, under what assumptions will it be?

(e) Compute $\mathrm{Var}(\hat{\beta}|x)$. In your formula, you can denote the conditional variance of $e = y - x\beta$ on x by $\mathrm{Var}(e|x) = \Sigma(x)$. What does your formula reduce to in the case that $\Sigma(x) = \sigma^2 I$?