Homework 9

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

Due Thurs May 1, 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 Between, within, and total variation

Fix a clustering assignment $C(1), \dots C(n) \in \{1, \dots K\}$ of points $x_1, \dots x_n$. Recall the definitions of within-cluster variation:

$$W_K = \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|_2^2,$$

and between-cluster variation:

$$B_K = \sum_{k=1}^{K} n_k \|\bar{x}_k - \bar{x}\|_2^2,$$

where $\bar{x}_k$ is the average of points in group k, $\bar{x}_k = \frac{1}{n_k} \sum_{C(i)=k} x_i$, and $\bar{x}$ is the overall average, $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$. Prove that the sum of these two measures of variation is equal to the total variation in the data set,

$$T = \sum_{i=1}^{n} \|x_i - \bar{x}\|_2^2,$$

i.e., prove that

$$T = W_K + B_K.$$

(Hint: start with total variation, and decompose this one big sum into K sums over the K clusters. Then, within each of these K sums, add and subtract the cluster averages.)

2 Bias and variance of linear regression

Consider a linear model
$$y_i = x_i^T \beta^* + \epsilon_i, \quad i = 1, \dots n,$$
where $x_1, \dots x_n \in \mathbb{R}^p$ are fixed predictor measurements, $\beta^* \in \mathbb{R}^p$ is a fixed unknown coefficient vector, and $\epsilon_1, \dots \epsilon_n$ are i.i.d. $N(0, \sigma^2)$ errors. Let $x \in \mathbb{R}^{n \times p}$ denote the predictor matrix (with rows $x_1, \dots x_n$), and $y = (y_1, \dots y_n) \in \mathbb{R}^n$ the outcome vector. You may assume that the predictor variables (columns of x) are linearly independent. Consider the linear regression estimate

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

and define the linear regression fit $\hat{r}$ by $\hat{r}(x_0) = x_0^T \hat{\beta}$, at an arbitrary point x_0.

(a) Recall that we define the bias of $\hat{r}$, at an arbitrary point x_0, by

$$\mathrm{Bias}\big(\hat{r}(x_0)\big) = \mathbb{E}[\hat{r}(x_0)] - r(x_0),$$

where $r(x_0) = x_0^T \beta^*$ is the value of the true regression function at x_0. Prove that $\mathrm{Bias}(\hat{r}(x_0)) = 0$. Hence argue (trivially) that the average bias across all inputs is still zero,

$$\frac{1}{n} \sum_{i=1}^n \mathrm{Bias}\big(\hat{r}(x_i)\big) = 0.$$

(b) Now we consider the variance of $\hat{r}$, averaged over all inputs. Prove that

$$\frac{1}{n} \sum_{i=1}^n \mathrm{Var}\big(\hat{r}(x_i)\big) = \frac{\sigma^2 p}{n}.$$

(Hint: write an equivalent expression for the left-hand side above in matrix notation, involving the trace of an $n \times n$ covariance matrix; it will make the calculation a lot easier.)

(c) From what you know about bias, variance, and their relationship to test error, what is the expected test error of the linear regression fit, averaged across all inputs,

$$\mathbb{E}\Big[\frac{1}{n} \sum_{i=1}^n \big(y_i' - \hat{r}(x_i)\big)^2\Big]?$$

Here $y_1,' \dots y_n'$ denotes an independent test set, i.e., a new independent copy of $y_1, \dots y_n$ (and $\hat{r}$ was only fit on the training set $y_1, \dots y_n$).

3 High-dimensional linear regression

Consider the usual linear regression setup, with outcome vector $y \in \mathbb{R}^n$ and predictor matrix $x \in \mathbb{R}^{n \times p}$. Let $x_1, \dots x_p$ be the columns of x (predictor variables). Let $\hat{\beta} \in \mathbb{R}^p$ be a minimzer of the least squares criterion
$$\|y - x\beta\|_2^2.$$

(a) Show that if $v \in \mathbb{R}^p$ is a vector such that $xv = 0$, then $\hat{\beta} + c \cdot v$ is also a minimizer of the least squares criterion, for any $c \in \mathbb{R}$.

(b) If $x_1, \dots x_p \in \mathbb{R}^n$ are linearly independent, then what vectors $v \in \mathbb{R}^p$ satisfy $xv = 0$?

(c) Suppose that $p > n$. Argue that there exists a vector $v \neq 0$ such that $xv = 0$. Then argue, based on part (a), that there are infinitely many linear regression estimates. Further show that there is a variable $i \in \{1, \dots p\}$ such that the regression coefficient of variable i can have different signs, depending on which estimate we choose. Comment on this.

4 Ridges, lassos, and baseball salaries

In this problem you'll use ridge regression the lasso to estimate the salary of various baseball players based on a bunch of predictor measurements. This data set is taken from the `ISLR` package, and R package that accompanies the Introduction to Statistical Learning textbook. Download the file "hitters.Rdata" from the course website and load it into your R session. You should now have the objects `x, y`, the former being a 263×20 matrix of predictor variables, and the latter a 263-dimensional vector of salaries. Type `colnames(x)` to see the list of predictor variable names. (For more information, download and install the `ISLR` package and type `?Hitters`.)

Download and install the `glmnet` package from the CRAN repository. We'll be using this package to perform ridge regression and the lasso. Finally, define

```
grid = 10^seq(10, -2, length=100)
```

This is a large grid of λ values, and we'll eventually instruct the `glmnet` function to compute the ridge and lasso estimates at each one of these values of λ.

(a) The `glmnet` function, by default, internally scales the predictor variables so that they will have standard deviation 1, before solving the ridge regression or lasso problems. This is a result of its default setting `standardize=TRUE`. Explain why such scaling is appropriate in our particular application.

(b) Run the commands

```
rid.mod = glmnet(x,y,lambda=grid,alpha=0)
las.mod = glmnet(x,y,lambda=grid,alpha=1)
```

This fits ridge regression and lasso estimates, over the whole sequence of λ values specified by `grid`. The flag `alpha=0` notifies `glmnet` to perform ridge regression, and `alpha=1` notifies it to perform lasso regression. Verify that, for each model, as λ decreases, the value of the penalty term only increases. That is, for the ridge regression model, the squared ℓ_2 norm of the coefficients only gets bigger as λ decreases. And for the lasso model, the ℓ_1 norm of the coefficients only gets bigger as λ decreases. You should do this by producing a plot of λ (on the x-axis) versus the penalty (on the y-axis) for each method. The plot should be on a log-log scale (i.e., with the argument `log="xy"` passed to the R `plot` command).

(c) Verify that, for a very small value of λ, both the ridge regression and lasso estimates are very close to the least squares estimate. Also verify that, for a very large value of λ, both the ridge regression and lasso estimates approach 0 in all components (except the intercept, which is not penalized by default).

(d) For each of the ridge regression and lasso models, perform 5-fold cross-validation to determine the best value of λ. Report the results from both the usual rule, and the one standard error rule for choosing λ. You can either perform this cross-validation procedure manually, or use the `cv.glmnet` function. Either way, produce a plot of the cross-validation error curve as a function of λ, with standard errors drawn, for both the ridge and lasso models.

(e) From the last part, you should have computed 4 values of the tuning parameter:

$$\lambda^{\text{ridge}}_{\text{min}},\ \lambda^{\text{ridge}}_{\text{1se}},\ \lambda^{\text{lasso}}_{\text{min}},\ \lambda^{\text{lasso}}_{\text{1se}}.$$

These are the results of running 5-fold cross-validation on each of the ridge and lasso models, and using the usual rule (min) or the one standard error rule (1se) to select λ. Now, using the predict function, with `type="coef"`, and the ridge and lasso models fit in part (b), report the coefficient estimates at the appropriate values of λ. That is, you will report two coefficient vectors coming from ridge regression with $\lambda = \lambda^{\text{ridge}}_{\text{min}}$ and $\lambda = \lambda^{\text{ridge}}_{\text{1se}}$, and likewise for the lasso. How do the coefficient estimates from the usual rule compare to those from the one standard error rule? How do the ridge estimates compare to those from the lasso?

(f) Suppose that you were coaching a young baseball player who wanted to strike it rich in the major leagues. What handful of attributes would you tell this player to focus on?