Homework 6: Outlier-Robust Linear Regression

36-350, Fall 2011

Due at 11:59 pm on Tuesday, 11 October 2011

INSTRUCTIONS: Submit a single plain text file, whose name clearly includes both your Andrew ID and the assignment number. Inside the file, clearly indicate which parts of your responses go with which problems. Raw R output is not acceptable, and will be marked down accordingly; you are communicating with a human being, and need to write in a human language. Files in other formats (Word, PDF, etc.), homework not submitted through Blackboard, and late homework will receive a grade of zero.

Direct objective: Using functions as arguments and as return values. *Indirect objectives:* Alternatives to least squares regression.

In introductory statistics classes, you learned to do linear regression by minimizing the mean squared error. That is, for the model where the response Y is linearly related to the predictor X,

$$Y = X\beta + \text{noise}$$

you learned to estimate β through "ordinary least squares",

$$\hat{\beta}_{OLS} = \underset{\beta}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i\beta)^2$$

This is optimal (in some senses) when the noise has a Gaussian distribution with constant variance, but it is vulnerable to outliers. One way to make linear regression more robust to outliers is to change how the coefficients β are estimated. There are several ways to do this; one is to not minimize the mean squared error, but instead the median absolute error,

$$\hat{\beta}_{MAE} = \underset{\beta}{\text{argmin}}\, \text{median}(|y_i - x_i\beta|)$$

This assignment writes code to compute $\hat{\beta}_{MAE}$.

Some functions and data for this assignment are in the file `http://www.stat.cmu.edu/~cshalizi/statcomp/hw/06/hw-06.R`.

1. Write a function to calculate the median of the absolute values of a vector of numbers. Write test cases to check that it works properly for inputs containing a mix of positive and negative numbers. (10)

2. Write a function to calculate the predicted response $x\beta$ in a linear model with a vector of coefficients β and a vector of independent variables x. Do not use a loop. Check that it works correctly for at least two values of β and x. *Hint*: `%*%`. (15)

3. Write a function which takes a matrix `x` of predictor vectors, one for each row, and a vector `beta` of coefficients, and returns the vector of predicted responses. Do not use a loop. Check that it works correctly when the matrix has only one row. Check that it works correctly, when `x` has multiple rows and columns, for at least two combinations of predictor and coefficients. (10)

4. Write a function which takes a matrix of `x` of predictor vectors, one for each row, a vector `beta` of coefficients, and a vector `y` of response values, and returns the median absolute error of the corresponding linear model. Check that it works properly for a case where `x` has multiple rows and columns. (15)

5. Plot the median absolute error as a function of β for the case where the predictors are given by `mystery.x.4` and the responses by `mystery.y.4`. Adjust the plotting range until you visually find a minimum. *Hint*: Write a new function, and look at Lecture 11. (10)

6. Write a function, `linear.mae`, which takes a matrix of predictors `x`, a vector of responses `y`, and returns the β estimated by minimizing the median absolute error. (You may provide this function with other arguments if you find that helpful.) *Hints*: Try the version of `gradient.descent` in the accompanying R. (15)

7. Write a test of `linear.mae` where `x` has multiple rows and columns, and the linear relationship between `y` and `x` is exact (i.e., no noise). Check that your `linear.mae` returns the correct coefficients, at least to a reasonable precision. (15)

8. Run `linear.mae` with `x=mystery.x.4` and `y=mystery.y.4`. What are the estimated coefficients? Are they close to the minimum point of the error from question 5? Should they be? (10)