Problem Set 2

Due on Friday, October 14, 2016 at 11:55 pm

How to Submit

Create one archieve file (.tgz or .zip, but **not** .rar) of your code and submit it via `ilearn.ucr.edu`. Supply one function that runs all of question 1 (including loading the data and generating the plots). Name it `q1.m`. Supply another that does the same for all of question 2. Name it `q2.m`. You may also include other `.m` files containing functions.

Do not supply any directories in your zip file. Each file should (in comments) list

- Your name
- Your UCR student ID number
- The date
- The course (CS 229)
- The assignment number (PS 2)

Q1: Lasso Regression [5 points]
The supplied file `comm.txt` contains the data for the "Communities and Crime" data set from the UCI Machine Learning repository.[1] It can be loaded with the command `D = load('comm.txt','-ascii');`. This problem has 1994 examples, each with 100 attributes. This is small by modern machine learning standards, but will provide a simple example for this problem set that can be completed easily on a laptop. Each attribute represents some quantity about a US community in the 1990s. The last attribute represents the rate of violent crimes in the same community (per capita). All of these attributes have been "normalized," that is scaled to fit on the same range. In some data sets this is a good idea, in others it is not. It has already been performed here, so we do not have a choice.

The goal is to predict the violent crime rate from the other attributes. We would like to know how well our method would work on "future unseen" communities. However, we don't know what those future communities would be. So, instead we will reserve part of the data as *testing data*. The dataset has already been placed in random order. Use the first 1000 communities as the *training data*. Use the last 994 as the *testing data*.

part a. Run lasso regression on the training set for a variety of λ values from 10^{-6} to 10^{-1}. For each λ value, compute the average squared error on the training data *and* on the testing data. Plot this relationship (average error versus λ) for both training and testing data (that is, you should have two curves). Label the curves (using `legend`). A semi-log plot is most appropriate (see `semilogx` in Matlab). Optimizing the lasso criterion is difficult. Use the `lasso` command (from the Statistics Toolbox) to do this. It has many options. Use it *only* in the form `[w,stats] = lasso(X,Y,'Lambda',lambda)`. Note that the intercept is returned as stats.Intercept (and not in w)!

part b. In a separate figure, plot the relationship between λ and the weights. In particular, for each weight, plot it versus λ (again on a semi-log plot). In total, you should have 100 different lines.

part c. Of course, picking λ cannot be done by looking at the performance on the testing data. However, we can use *k-fold cross-validation*. In this method we split the training data into a number of equally sized parts (for the this assignment, we will pick $k = 10$ parts or folds). To judge the accuracy of using a particular value of λ, we train on all but one of these folds and testing the performance on the left-out fold. We repeat for all k folds that could be left-out and average the result. This average is our estimate of the true error.

Do this, and plot the 10-fold cross-validation average squared error on the same plot as that from part a.

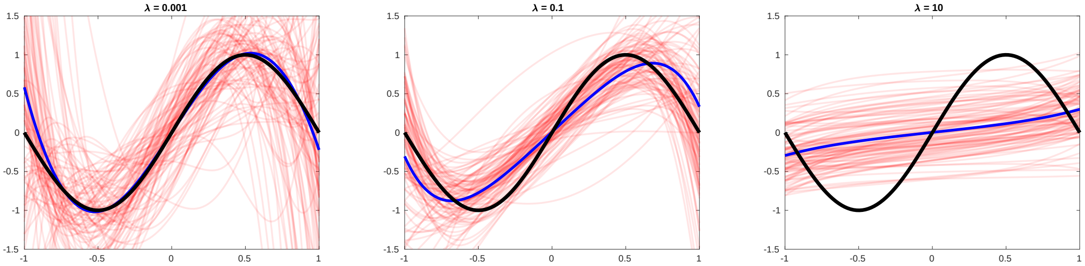
[1] The repository can be found at `http://www.ics.uci.edu/~mlearn/MLRepository.html`. The raw data can be found at `http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`

Q2: Ridge Regression Bias and Variance [5 points]

In this problem, you are to reproduce the bias-variance plots shown in class for ridge regression. In particular, you are to use the function $f(x) = \tan(\pi x/3) + (x - 0.5)^2$ and a dataset of 10 x-points, drawn uniformly at random from the interval $[-1, +1]$. The y-values should be $f(x)$ plus Gaussian noise with standard deviation of 0.5. You features should be all powers of x from 0 through 5.

You are to make three plots, one for each of three different values of lambda: $0.001, 0.1, 10$. For each plot, plot 100 samples of the resulting learned functions in red. Plot the average function across 1000 samples in blue. Plot the true function in black. Set the axes to be consistent across the plots: x ranging from -1 to +1, and y ranging from -0.5 to +4.5.

Below are the same plots, but for the function $f(x) = \sin(\pi x)$.


λ = 0.001
λ = 0.1
λ = 10
1.5
1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1