Stats 598z: Homework 5

Due before midnight Fri, Mar 30

Important:

R code, tables and figures should be part of a single .pdf or .html files from R Markdown and knitr. See the class reading lists for a short tutorial.
Include R commands for all output unless explicitly told not to.
If you collaborated with anyone else, mention their names and the nature of the collaboration

1 Problem 1: Ridge regression [100pts]

(a) Sample a random 3×4 matrix X, and a random 4×1 matrix y. Solve $w = (XX^\top)^{-1}(Xy)$. Do not invert any matrices, directly use `solve`. The elements of the matrices can be Gaussian distributed. [3]

(b) What happens when `X` and `Y` are 4×3 and 3×1 matrices? [2]

(c) What's the solution to both for the regularized problem $w = (XX^\top + \lambda I)^{-1}(Xy)$? Let $\lambda = 5$. [5]

(d) Write a function `train.ridge` that takes as input a two element list `ip_data` and a scalar `lambda`. Internally, call the first element of `ip_data` as X (a matrix) and the second as y. Return the ridge regression solution for these values of `X, Y` and `lambda` [5]

(e) Store the `X` and `y` from part (a) as two elements of a list. Call `train.ridge` with this as the first input, and `lambda = 5` as the second. You should get the same output as part (c). [5]

(f) Assign the previous list the class `"ridge"` (it is now an object of type `ridge`). Also define a generic function `train`. Now you should get the same output as the previous part by calling `train` instead of `train.ridge`. Show this. [5]

(g) Write a function `pred_err.ridge` that takes as input a weight `w` and an object of type `"ridge"`. It should return the prediction error between the actual `y` and the prediction from `X` and `w`. [10]

(h) Finally, write a function `crossval`. This takes 4 inputs, an object of class `"ridge"`, a vector of `lambda`'s, and an integer `k`. The function works as follows: first create k 'folds' of the input object, splitting it into training and test objects of the same class as the input. For each fold, call `train` and then `pred_err` for all values of `lambda`. Return the $k \times l$ matrix of prediction errors, where l is the length of the `lambda` vector. [30]

(i) Download the credit dataset from http://www-bcf.usc.edu/~gareth/ISL/data.html. Load using `read.table`. This has a number of columns: extract column (Balance) as `y`, and extract (Income, Limit, Ratings Age and Education) as `X`. Convert this into a `ridge` object called `my_credit`. [10]

(j) Carry out 5-fold cross-validation with `my_credit` as the data. Set `lambda` to `c(0, 0.1, 0.5, 1, 5, 10, 50, 100, 1000)`. Show the output. [10]

(k) Calculate the mean prediction error for each values of `lambda`, and plot it. [8]

(l) Choose the best `lambda`. Now, find the ridge-regression coefficient vector for this `lambda` using the *entire* data. [7]