Homework 12 – Intro. to Computational Statistics

For all problems, please show all your work. As described in the Homework Guidelines, use RMarkdown to write up your work as a .Rmd file, "knit" the result to a PDF file, and submit only that PDF file. Be sure to use R code for all your calculations, and the latex equation format to write up any math. See the Homework Guidelines for more formatting details.

Using one or two high-dimensional datasets of your choice, estimate a shrinkage and an SVM model and test them out-of-sample.

A large number of high-dimensional datasets can be found here: http://archive.ics.uci.edu/ml/datasets.php . Be sure to choose those that make your life easier, rather than something that takes a lot of manipulation to get into shape. But feel free to use other data or a dataset you have already used, as long as they have at least 10 independent variables and a continuous dependent variable (for lasso/ridge) and/or a binary dependent variable (for SVM). You can also convert a continous dependent variable to a binary for the SVM stage, as we did in the lesson.

1. Use your dataset with a continuous dependent variable:

a. Divide your data into two equal-sized samples, the in-sample and the out-sample. Estimate the elastic net model using the in-sample data with at least three levels of alpha (ie, three positions in between full lasso and full ridge; eg, alpha = 0, 0.5, and 1), using cv.glmnet to find the best lambda level for each run. (Remember that glmnet prefers that data be in a numeric matrix format rather than a data frame.)

b. Choose the alpha (and corresponding lambda) with the best results (lowest error), and then test that model out-of-sample using the out-sample data. To find the lowest MSE associated the best lambda value from a cv.glmnet ouput, note that my_cv_output has inside it an object called $cvm, which is a vector of MSE values associated with each lambda level; min(my_cv_output$cvm) will give you the best MSE from that model.

c. Compare your out-of-sample results from b to regular multiple regression: fit a regression model using the in-sample data, predict the out-of-sample yhat using the out-of-sample X, and estimate the error between the predicted yhat and the true out-of-sample y. Which worked better, the model from b or the regression model?

d. Which coefficients are different between the multiple regression and the elastic net model? What, if anything, does this tell you substantively about the effects of your independent variables on your dependent variable?

2. Repeat the same process using your dataset with a binary dependent variable:

a. Divide your data into an in-sample and out-sample as before, and using the in-sample data, estimate an SVM using at least two different kernels and `tune` to find the best cost level for each. (Remember that SVM expects the dependent variable to be a factor, so if your dependent variable is numeric 0s and 1s, just us as.factor() to convert it first.)

b. Chose the kernel and cost with the best in-sample accuracy, and then test that model out-of-sample using the out-sample data. How does the out-of-sample accuracy compare to the in-sample accuracy?