MTH6101 Introduction to Machine Learning

Laboratory week eleven

This lab has a two fold intention. The **first part** completes the analysis of the `diabetes` data set (Efron et al. 2004), using lasso. The data set had $n = 442$ diabetes patients measured on $p = 10$ baseline variables. A prediction model was desired for the response variable `y`, a measure of disease progression one year after baseline. In the **second part**, we will have a look at the automatic function from the `lars` library for lasso cross-validation using the same `diabetes` data.

Before you **start** your `RStudio` session, install and load the following libraries: `cvTools`, `lars`.

1. The initial part of the analysis will see you replicate loading the `diabetes` data, formatting it and creating the same 3 : 1 partition for analysis.

2. Using the training data and function `lars` from the library of the same name, fit a **lasso** analysis to these data and store it in variable `LS`.

3. Using the test partition of data, build lasso predictions and store them in a variable `PL`. Akin to what was done earlier, build variable `Yobs` with command

   ```
   matrix(nrow=nrow(DAT[Test,]),ncol=ncol(PL$fit),byrow=FALSE,DAT$y[Test])->Yobs
   ```

   This variable `Yobs` is then used with function `apply` to compute values of MSE at every breakpoint in the lasso path. Store these values in variable `MSEL`.

4. Plot MSE at breakpoints using the values of `MSEL` and identify the minimum MSE.

5. Give the coefficients of the path at that stage of minimal MSE and compute the shrinkage.

6. Plot the lasso path.

7. **(Extra)** The package `lars` includes K fold cross validation capabilities in the function `cv.lars` that produces automatically an error plot against fraction of L_1 norm (shrinkage). Explore this function with the method `lasso` and the `diabetes` data set with `K=10`, response `y=diabetes$y` and using options `x=diabetes$x` and `x=diabetes$x2`.

8. **(Extra)** Besides lasso, the function `lars` includes several methodologies for computing families of models. Using the same `diabetes` dataset and the syntax `lars` with `normalize=FALSE`, produce paths for different types of analyses: `"lasso", "lar", "forward.stagewise", "stepwise"`.