Lab 8, Week 11

Sidi Wu

As in the lecture 9 exercises, you should be working from your `mars` project for this lab. You will need the `testfwd_stepwise.RData` file from the `tests/testthat/` folder of your `mars` project. In addition, copy the files `testbwd_stepwise.RData` and `testLOF.RData` from the `Exercises/ProjectTestfiles` folder of the class GitHub repository to your `tests/testthat` folder.

1. Load the data from `tests/testthat/testfwd_stepwise.RData`. The object `testfwd` is a list output by `fwd_stepwise()` with components `y`, `B` and `Bfuncs`. (For later, note that you also load the test `mars.control` object `testmc`.) Execute the following R commands to combine the response `y` and basis functions `B` into a data frame, and then fit a model using all terms in the data frame.

```
load(".../testfwd_stepwise.RData")
dat <- data.frame(y=testfwd$y,testfwd$B)
ff <- lm(y~.,dat)
```

2. Print the coefficients of the model with `coefficients(ff)`. The `NA`s mean that there are collinearities in the model; i.e., some of the terms in the model are linear combinations of the others.

3. One obvious collinearity is that `B0` is an intercept, and `lm()` also adds an intercept when passed a formula with `y~.`. We can stop R from adding an intercept with the formula `y~.-1`. Re-fit using `lm(y~.-1,dat)` and re-print the coefficients. You should still see evidence of collinearity.

4. Write a version of `LOF()` that returns the GCV criterion described in the `mars3.pdf` notes. In addition to the formula and data, `LOF()` should take the `mars.control` object as an argument.

    - Note: The GCV formula includes the number of **non-constant** basis functions, which it calls `M`. You can deduce this from the model fitted in `LOF`. We will **not** make any adjustment to `M` for possible collinearity. However, we will account for collinearity by taking $C(M)$ to be the sum of the hatvalues from the fitted model; if `ff` is the fitted model $C(M)$ equals `sum(hatvalues(ff))`.

5. Calculate the LOF for the full model in the data frame `dat` from question 1 and the `mars.control` object `testmc` you loaded in question 1. Load `tests/testthat/testLOF.RData` to load the object `testLOF` and use `all.equal()` to compare it to your answer.

```
load(".../testLOF.RData")
lof <- LOF(y~.-1,dat,testmc)
all.equal(lof,testLOF)
```

6. Using the hints in the `mars6` lecture, implement the backward selection algorithm of MARS in a function named `bwd_stepwise()` that takes the output of your `fwd_stepwise()` as input. When your `bwd_stepwise()` calls `LOF()`, use the formula `y~.-1` as discussed in question 2. Test your implementation as follows.

    a. Load `tests/testthat/testbwd_stepwise.RData`
    b. Run `bwd_stepwise()` with inputs `testfwd` and `testmc` (loaded in question 1) and save the output as `bwd`.

c. Use `all.equal()` to test that `bwd` is the same as `testbwd`.

```
load(".../testbwd_stepwise.RData")
bwd <- bwd_stepwise(testfwd,testmc)
all.equal(bwd,testbwd)
```