Assignment 4

Irina Gaynanova

2/19/2019

HW4: Coordinate-descent algorithm for LASSO

Introduction

In this HW, you will be asked to implement coordinate-descent algorithm for LASSO as described in class. Recall that LASSO is seeking the minimizer for the following problem

$$\hat{\beta} = \arg\min \left\{ (2n)^{-1} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}.$$

Here we assume that Y and X are already centered, and X is appropriately scaled (see class notes). You will be asked to implement the appropriate transformations in this assignment. We will use 5-fold cross-validation for tuning parameter selection.

Functions Instructions

The starter code for all functions with detailed description is provided in **LassoFunctions.R**. The described functions should work exactly as specified, but you are welcome to create any additional functions you need. You are not allowed to use any outside libraries for these functions. I encourage you to work gradually through the functions and perform frequent testing as many functions rely on the previous ones.

Things to keep in mind when implementing:

- I will test your functions speen on large p, small n dataset (below), so you want to use implementation that takes that into account
- You can and should indirectly check your code on simple examples before proceeding to the data (i.e. what happens when large lambda is supplied? What happens on toy example used in class?). I will use automatic tests to check that your code is correct on more than just the data example with different combinations of parameters
- You want to make sure that parameters supplied to one function are correctly used in subsequent functions (i.e. the convergence level ε)
- I expect it will take you some time to figure out how to split the data for cross-validation. Keep in mind that the split should be random, in roughly equal parts, and should work correctly with any sample size n and any integer number of folds K as long as $n \geq K$.

Data instructions

We will test your implementation using riboflavin data available from the R package `hdi`. The **RiboflavinDataAnalysis.R** gives starter code for loading the dataset and instructions. This is a high-dimensional dataset with the number of samples $n = 71$ much less than the number of predictors $p = 4088$. The goal is to predict the riboflavin production rate based on gene expression.

You will be asked to do the following:

- use **fitLASSO** function to see how the sparsity changes with λ value, and test the speed
- use **cvLASSO** function to select the tuning parameter, see how $CV(\lambda)$ changes with lambda

Grading criteria

Your assingment will be graded based on

- correctness (50%)

Take advantage of objective function values over iterations as a way to indirectly check the correctness of your function. Also recall that you know the right solution in special cases, so you can check your function in those cases (i.e. when λ is very large, or when $\lambda = 0$ and you have a nice problem)

- speed (30%)

You will get +5 points if your code is comparable to mine (fitLASSO with 30 tuning parameters on riboflavin data takes around 5.5 seconds on my laptop, you will get + 5 if your code is 5 - 7 seconds), and you will get +10 points if your code is faster than mine (< 5 seconds).

- code style/documentation (10%)

You need to comment different parts of the code so it's clear what they do, have good identation, readable code with names that make sense.

- version control/commit practices (10%)

I expect you to start early on this assignment, and work gradually. You want to commit often, have logically organized commits with short description that makes sense.