MATH 680 Homework 1 Fall 2015

September 10, 2015

This homework is due on Wednesday, September 16 at 11:59pm. Provide both pdf, R files (and LaTeXfile for exercise 3). The naming convention is yourlastname_hw1.pdf, yourlastname_hw1.R and yourlastname_hw1.lyx

[Friedman(2001)]

1. (20%) Walk through the following steps: Sign up on https://github.com/. Create a repository (repo). Add me as a collaborator (github ID: emeryyi). DownloadandinstallSourceTreehttps://www.sourcetreeapp.com/oralternativelyGitHub Desktop https://desktop.github.com/ as the graphical user interface (GUI) client. Clone the repository your created on github.com to your local folder using the GUI client. Add an R file in the local folder, then stage and commit changes. Push the changes to your origin repo on github.com Make a new branch and switch to the new branch using checkout. Make some changes in the new branch, then stage and commit. Switch to the master (old) branch using checkout. Make some changes in the old branch (but make sure they do not conflict with the changes in the new branch), then stage and commit. Merge the old branch with the new branch, and then commit. Push all the changes to the remote server. Provide the hyperlink and a screen shot of your github repo in a pdf file, which reflects all the commits, branching and merging you made in your repo. 2. (80%) In this exercise, we reproduce the random function generator (RFG) model by [1] (the paper is included, see page 19 section 6.1) in R. The RFG model generates very complicated data with non-linearity and higher-order interactions. The data generated by RFG can be used for 1 testing the performance of many fully non-parametric regression-based learning methods such as kernel support vector machine (KSVM), gradient boosting, random forests and others. The idea is to generate a data frame with N = 100 observations of simulation data yi,xiN1 according to yi =f(xi)+i, where is are independent generated from the normal distribution N(0, 1). In the data frame y takes up the first column, and the p = 10 dimensional vector x takes up the rest columns. Hence the data frame will be 100 11. Each row represents one observation. The f functions is randomly generated as a linear combination of functions g 20: 20 f(x) = algl(zl), (1) l=1 where coefficients a 20 are randomly generated from a uniform distribution a U[1, 1]. Each l1l gl(zl) is a function of a randomly selected pl-size subset (sub-vector) of the p-dimensional variable x, with p = 10, and the size of each subset pl is randomly chosen by pl = min(1.5 + r , p), and r is generated from an exponential distribution r

Exp(0.5) with mean 2. Each zl is a sub-vector of x defined as zl = xW (j)pl , (2) l j=1 where each Wl is an independent permutation of the integers 1, . . . , p. Each function gl(zl) is an pl-dimensional Gaussian function: 1T gl(zl) = exp 2(zl l) Vl(zl l) , (3) where each of the mean vectors 20 is randomly generated from the same distribution as that l1 of the input variables x. The pl pl covariance matrix Vl is also randomly generated by Vl = UlDlUTl , (4) where Ul is a pl pl uniformly distributed random orthonormal matrix and Dl = diagd1l...dpll. The variables djl are randomly generated from a uniform distribution djl U[0.1, 2.0]. We gen- erated x from joint normal distribution x N(0, Ip) with p = 10. Hints: In this exercise, you will probably use the following R functions: l1 rexp, runif, rnorm floor qr.Q, qr diag sample 2 To generate the uniformly distributed random orthonormal matrix Ul, please see this post http://math.stackexchange.com/questions/138512/sampling-q-uniformly-where-qtq-i. Specifically, you need to generate a Gaussian matrix G, and use QR decomposition to find the corresponding Q matrix, which will be a uniformly distributed random orthonormal matrix. Use R functions qr.Q and qr to achieve this. Or much easier, you can just use the genQ function in the package bootSVD https://cran.r-project.org/web/packages/bootSVD/ index.html to do the same job. If it is your first time using R, learn the above functions using their online manuals before starting the actual coding. 3. (Extra credit 40%) Walk through the following steps: Download and install LYX. To make LYX work correctly, Mac users may also need to install MacTEX https://tug.org/mactex/ while the Windows users may need to install MikTEX http://miktex.org/. Reproduce this document using LYX, make sure that all text contents, mathematical formulas, references and typesetting (bullet, bold text, url, title, etc.) are accurately reproduced. However, the layout of the document do not need to be exactly the same. (paper margin, line space, etc.).

References

[Friedman(2001)] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.