Final Project

Parallel R

Sunny Wang*

AY-2023/2024

Project instructions

- You should do the project in groups of 3.
- The goal of this project is for you to acquire the necessary skills to perform a reproducible simulation study by using parallel computing.
- As a group, you should choose a statistical / ML topic (unique topic per group) that you are interested in, which is fairly computationally intensive. If you are unsure, you should double-check with me before proceeding.
- Some topic suggestions:
    1. Non-parametric density estimation
    2. Non-parametric regression
    3. Non-parametric classification
    4. Non-parametric inference
    5. Clustering
- Within your topic, identify two or three estimators / algorithms to study. Your goal is to perform a small scale simulation study to compare these methods using some metric (e.g mean-squared error etc).
- You are allowed to use out-of-the-box implementations if available. If you use an existing implementation, you should have a good idea about what the function is really doing.
- Your simulation study should be performed using parallel computing in R. You should perform a reasonably large number of replications in your simulation study (e.g 500 replications) before making conclusive statements about things like estimator performance.
- You should care about reproducibility of your simulation (things like seed setting etc). You should try to ensure that when I run your code, I can reproduce the results that you expect me to see.
- You should describe your problem and the estimators you are using. For example, write down the formula or algorithm in the report and briefly describe it. You don't have to go into deep details, such as the theoretical properties of a particular algorithm.
- To illustrate your findings, you should include suitable visualisations (e.g boxplots etc) in your report.
- You should profile your code, identifying and reporting the bottlenecks. You don't necessarily have to fix all of them, since they might be out of your control (for example, it might be due to a bad implementation of the algorithm you are interested in).
- In addition to considering estimator / algoroithmic performance, you should also record and compare the computational times between the methods.

*sunny.wang@ensai.fr

- Think like a scientist. For example, how does the data distribution affect your results? How important are the tuning parameters? How important is the metric you are using for your comparison? (e.g does using L^2 vs L_∞ norm change your results?) How does the sample size affect the results? Is the slow computational time arising from one method because of the poor implementation, or is it intrinsic to the problem?
- Your code needs to run!!! If I can't run it, I can't grade it. I won't be able to debug your scripts for you.
- The final output should be a PDF file, with all the relevant scripts included. If you write your own functions, ensure that you document them sufficiently well.

Some useful references:

1. Introduction to statistical learning with R (James, Witten, Hastie, Tibshirani)
2. Elements of statistical learning (Hastie, Tibshirani, Friedman)
3. All of nonparametric statistics (Wasserman)