61011 Lab - Week 5

David Wong and Christopher Yau

October 2020

1 Introduction

The targets for this week are listed below. You should aim, at least, to complete Level 1.

You should then focus on completing Level 2 and 3 tasks from previous weeks BEFORE attempting the Level 2 and 3 tasks for this week.

The Level 2 and 3 tasks for this week involve implementing some algorithms, rather than simply using the scikit pre-built methods. They are therefore somewhat more difficult than previous weeks' tasks.

The tasks this week correspond to the taught material on Ensemble Methods and Feature Selection.

During (or before) the live Wednesday lab sessions, you may request help by entering your name on the following form:
https://forms.gle/JFfp1TddLxRFNTYb8.

When you join the weekly blackboard collaborate session, a teaching assistant will move you to a break out room to discuss your issue.

Level 1

On a data set of your choice:

- Build a random forest classifier, and compare the performance (accuracy, sensitivity, specificity) to a single decision tree. Repeat this process on a second data set. Select the second data set to have a large range of features (> 10) and examples (> 1000). You may need to download data from the UCI repository at https://archive.ics.uci.edu/ml/index.php - note that there is a searchable interface here to find appropriate data sets. In the example notebook, we use a data set to predict obesity, but feel free to try another data set!
- Investigate the impact of changing the number of trees, and whether bootstrap sampling is used
- For the breast cancer dataset (included with sklearn) Build an adaboost classifier using sklearn. Plot the classification accuracy as you vary the number of estimators between 0 and 100. What do you notice?

Level 2

- Implement the bagging algorithm by writing the bootstrapping process from scratch. You may use scikit's inbuilt decision tree function. How does the accuracy of bagging compare to random forests?
- Using the pseudocode description in the course handbook, create a boosting algorithm, from scratch (without sklearn), that works on a data set with two features.
- for simple logistic regression, implement greedy forward feature selection. Compare the results to lasso regression by using sklearns inbuilt LogisticRegression function with the penalty parameter set to 'l1'. Note that lasso regression is another type of feature selection paradigm, sometimes called *integrated* feature selection. It is NOT part part of the testable syllabus.

Level 3

Add the bagging and bootstrap implementations from Level 2 to the test framework that you created last week. Extend the test framework by providing options to plot an ROC curve, and to provide p-values by permutation testing (see https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf for details