CSE 417A: Homework 5

Due: November 13, 2014

Notes:

- Please keep in mind the collaboration policy as specified in the course syllabus. If you discuss questions with others you **must** write their names on your submission, and if you use any outside resources you **must** reference them. **Do not look at each others' writeups, including code.**
- Instructions for how to get files from the SVN repository are available on the course website and on Piazza.
- Homework (in hardcopy) is due **at the beginning of lecture.** In addition, your code submissions must also be timestamped before lecture begins.
- Please comment your code properly.
- There are 4 problems on 2 pages in this homework.

Problems:

1. (50 points) The purpose of this problem is to write code for bagging decision trees and computing the out-of-bag error. You may use matlab's inbuilt `fitctree` function, which learns decision trees using the CART algorithm (read the documentation carefully), but do not use the inbuilt functions for producing bagged ensembles. In order to do this, you should complete the stub `BaggedTrees` function available in your SVN repository. Note that it only returns the out-of-bag error. You may want to use other functions that actually construct and maintain the ensemble. You may assume that all the $\mathbf{x}$ vectors in the input are vectors of real numbers, and there are no categorical variables/features. You will compare the performance of the bagging method with plain decision trees on the handwritten digit recognition problem from HW3 (the dataset is in `zip.train` and `zip.test` – the same files as in HW3, available from `http://amlbook.com/support.html`) . We will focus on two specific problems – distinguishing between the digit one and the digit three, and distinguishing between the digit three and the digit five. Here are the steps for this problem:

   (a) Complete the implementation of `BaggedTrees`. You may choose any reasonable representation that you wish; the two strict requirements are that you plot the out-of-bag error as a function of the number of bags from 1 to the number specified as input (`numBags`), and that you return the out-of-bag error for the whole ensemble of `numBags` trees. Include the plots (with clearly labeled axes) in your writeup, and, of course, commit your code.

(b) Run the provided `OneThreeFive` script, which creates training datasets based on the one-vs-three and three-vs-five cases we are interested in, and calls both the in-built decision tree routine and your bagging code, printing out the cross-validation error for decision trees and the OOB error for your bagging implementation. Report the results in your writeup.

(c) Now, learn a single decision tree model for each of the two specified problems (one-vs-three and three-vs-five) on the training data, and test their performance on `zip.test` – what is the test error? Similarly, learn a single ensemble of 200 trees on the training data for each of the two specified problems and test the performance of the ensembles on the test data. Report your results.

(d) Summarize and interpret your results in one or two concise paragraphs as part of your writeup.

2. (30 points) Download the "Adult" (also known as "Census") dataset from `https://archive.ics.uci.edu/ml/datasets/Adult` – read and understand the documentation. Using matlab's inbuilt `fitctree` and `TreeBagger` functions (and any extensions / options thereof) and **only** the `adult.data` dataset (which you may divide into training and validation sets as you please, or cross-validate, etc), decide on a model you will use (for example, decision trees with a certain level / kind of pruning, or a bagged ensemble of 150 trees). Explain your rationale for selecting this model and some measure of accuracy on the training dataset (e.g. cross-validation error, validation error, or OOB error), then learn a hypothesis on the full `adult.data` dataset and apply it to the `adult.test` dataset. Report the error of your hypothesis on the test data. Also, submit a matlab script file called `adultTest.m` that goes through the steps of learning the final hypothesis on the training dataset, applying it to the test dataset, and computing the classification error on the test dataset. **Note:** Be careful in applying matlab's inbuilt decision trees to this dataset – you need to tell `fitctree` which features are categorical and which are not. Also note that you have a lot of freedom in this problem, and should spend some time reading the matlab documentation and learning about the dataset – part of the point of this problem is to help you become self-sufficient in using available tools and applying them to real-world data, so we will provide limited guidance, especially for things that can be learned from looking up the documentation.

3. (15 points) AIMA Problem 18.22

4. (5 points) AIMA Problem 18.23 (you can assume squared error if that makes things easier to think about)