605.449 — Introduction to Machine Learning

Programming Project #4

Due: October 22, 2017

The purpose of this assignment is to give you a chance to get some hands-on experience learning decision trees for classification and, for extra credit, regression. This time around, we are not going to use anything from the module on rule induction; however, you might want to examine the rules learned for your trees to see if they "make sense." Specifically, you will be implementing a standard univariate (i.e., axis-parallel) decision tree and will compare the performance of the trees when grown to completion on trees that use either early stopping (for regression trees) or reduced error pruning (for classification trees).

For decision trees, it should not matter whether you have categorical or numeric attributes, but you need to remember to keep track of which is which. In addition, you need to implement that gain-ratio criterion for splitting in your classification trees. For the regression trees, all of the attributes will be numeric.

For this assignment, you will use three classification datasets (plus three regression data sets for the extra credit) that you will download from the UCI Machine Learning Repository, namely:

1. Abalone — `https://archive.ics.uci.edu/ml/datasets/Abalone`

   [Classification] Predicting the age of abalone from physical measurements.

2. Car Evaluation — `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`

   [Classification] The data is on evaluations of car acceptability based on price, comfort, and technical specifications.

3. Image Segmentation — `https://archive.ics.uci.edu/ml/datasets/Image+Segmentation`

   [Classification] The instances were drawn randomly from a database of 7 outdoor images. The images were hand segmented to create a classification for every pixel.

4. Computer Hardware — `https://archive.ics.uci.edu/ml/datasets/Computer+Hardware`

   [Regression] The estimated relative performance values were estimated by the authors using a linear regression method. The gives you a chance to see how well you can replicate the results with these two models.

5. Forest Fires — `https://archive.ics.uci.edu/ml/datasets/Forest+Fires`

   [Regression] This is a difficult regression task, where the aim is to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data .

6. Wine Quality — `https://archive.ics.uci.edu/ml/datasets/Wine+Quality`

   [Regression] This contains two data sets, one for red wine and one for white. Either combine the data sets into a single set for the regression task or build separate regression trees. This is your choice; however, we expect the separate trees to be better. The objective is to learn a model to assess the quality of wine.

For this project, the following steps are required:

- Download the six (6) data sets from the UCI Machine Learning repository. You can find this repository at `http://archive.ics.uci.edu/ml/`. All of the specific URLs are also provided above.
- Implement the ID3 algorithm for classification decision trees using gain-ratio as the splitting criterion.
- Implement reduced-error pruning to be used as an option with your implementation of ID3.
- Run your ID3 implementation on each of the three classification data sets, comparing both pruned and unpruned versions of the trees. These runs should be done with 5-fold cross-validation so you can compare your results statistically. You should pull out 10% of the data to be used as a validation set and then use the remaining 90% for cross validation. You should use classification error for your loss function.

- Extra credit (for up to 20 additional points):
  - Implement the CART algorithm for regression decision trees using mean squared error as the splitting criterion.
  - Incorporate a cut-off threshold for early stopping. If the threshold is set to zero, this should indicate no early stopping.
  - Run your CART implementation on each of the three regression data sets, comparing both full and stopped versions of the trees. You will need to tune the stopping threshold and should use the same procedure for extracting a validation set to serve as your tuning set. The runs should be done with 5-fold cross-validation so you can compare your results statistically. You should use mean squared error for your loss function.
- Write a very brief paper that incorporates the following elements, summarizing the results of your experiments.
  1. Title and author name
  2. A brief, one paragraph abstract summarizing the results of the experiments
  3. Problem statement, including hypothesis, projecting how you expect each algorithm to perform
  4. Brief description of algorithms implemented
  5. Brief description of your experimental approach
  6. Presentation of the results of your experiments
  7. A discussion of the behavior of your algorithms, combined with any conclusions you can draw
  8. Summary
  9. References (you should have at least one reference related to each of the algorithms implemented, a reference to the data sources, and any other references you consider to be relevant)
- Submit your fully documented code, the outputs from running your programs, and your paper. Your grade will be broken down as follows:
  - Code structure – 10%
  - Code documentation/commenting – 10%
  - Proper functioning of your code, as illustrated by the code outputs – 30%
  - Summary paper – 50%