Assignment 1 - Decision Tree Learning

Updated January 19, 2018

This assignment is to be completed in **teams of two**. Your partner on this assignment may **not** be your partner on any of the three dataset interview assignments.

You may use any programming language (Python works well. C is not recommended.)

IMPORTANT: You will be implementing your OWN software to run the ID3 algorithm.

- You should complete this assignment using ONLY the following resources:
  a. Course content posted to Blackboard (Udacity ML lecture videos and associated notes, Mitchell textbook chapter, etc.)
  b. Lecture notes from class
  c. Conversations with course instructor and/or course assistants
  d. Conversations with other students in the class. (This is allowed, but if you discuss the assignment with students who are not on your team, please include their names as a cited reference in your report.)
- **You may NOT get help from anyone outside the class (in person or online), or from any other resources (e.g. online forums, ML code packages, etc.)**
- You may use built-in data structures such as lists and trees, but NOT any kind of built-in ML functions (such as anything specific to decision trees, train-test splitting, etc.)

Part 1: Implement the basic ID3 algorithm

1. Implement the basic ID3 algorithm using information gain as a way to choose the "best" attribute at each iteration.
2. Test your code on the PlayTennis example (which is discussed at length in the Mitchell book and in the Udacity ML ID3 notes).

Part 2: Test your algorithm on a new dataset

1. Find any other dataset (or create your own) that has at least 10 features (not including the class label) and 100 instances. Each feature can be either discrete or continuous. Each feature can also be either relevant or not relevant (in your opinion) to the classification problem at hand.
2. Randomly select 20 of your examples to set aside as your **test set**.
3. Run your code against the training data from this new dataset. At each iteration, test on your test set. Generate a plot that shows both training error and testing error as a function of iteration. Explain what you see in your plot.

Part 3 –Prevent overfitting

1. Divide your training data into a **training** subset and a **validation** subset.
2. Implement reduced-error pruning as a way to prevent overfitting. (See p. 69 of Mitchell, Chapter 3.)
3. Run ID3 and your pruning code on the new training subset. Generate a plot that shows training error, validation error, and testing error as a function of the tree building and subsequent pruning process.
4. Repeat #3 using different selections of training and validation subsets. Do your results change? Why or why not? How do these results relate to the **test error** that you observe?

Deliverables

Only one assignment submission is needed per team. Please include each deliverable as a separate attachment in your submission.

1. ZIP file containing at least 80 instances that make up your training set.
2. ZIP file containing at least 20 instances that make up your test set.
3. ZIP file containing source code file(s), with sufficient documentation (e.g. comments, function descriptions, etc.) that someone could read through your code and easily understand how it works.
4. PDF of project report that includes the following elements:
    a. Part 1: High level description of your ID3 code and how it works.
    b. Part 1: Results of running your code on the PlayTennis example dataset.
    c. Part 2: Description of your dataset, where you found it, number of instances, what each feature represents, whether each feature is discrete or continuous, how you chose to discretize any continuous features, etc.
    d. Part 2: Results of running your code on this new dataset, including the error plot described above in Part 2.
    e. Part 3: High level description of your pruning code and how it works.
    f. Part 3: Results of running the pruning on your own dataset, including the error plots and responses to the questions described above in Part 3.