Assignment #4: Decision Trees
CSCI 374 Fall 2019 Oberlin College
Due: Thursday November 7 at 3:00 PM

Background

Our fourth assignment this semester has three main goals:

1. Implement the ID3 algorithm for learning decision trees (with human-interpretable learning),
2. Extend ID3 with the ability to consider numeric attributes (using the approach from C4.5 and CART)
3. Practice inspecting the information learned by a machine learning algorithm.

Gitting Started

To begin this assignment, please follow this link:

https://classroom.github.com/a/OCAwCrNU

Data Sets

For this assignment, we will learn from four pre-defined data sets:

1. **monks1.csv**: A data set describing two classes of robots using all nominal attributes and a binary label. This data set has a simple rule set for determining the label: `if head_shape = body_shape OR jacket_color = red, then yes, else no`. Each of the attributes in the monks1 data set are nominal. Monks1 was one of the first machine learning challenge problems (http://www.mli.gmu.edu/papers/91-95/91-28.pdf). This data set comes from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/MONK%27s+Problems
2. **iris.csv**: A data set describing observed measurements of different flowers belonging to three species of *Iris*. The four attributes are each continuous measurements, and the label is the species of flower. The Iris data set has a long history in machine learning research, dating back to the statistical (and biological) research of Ronald Fisher from the 1930s (for more info, see https://en.wikipedia.org/wiki/Iris_flower_data_set). This data set comes from Weka 3.8: http://www.cs.waikato.ac.nz/ml/weka/ and is also on the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/iris
3. **occupancy.csv**: A data set of measurements describing a room in a building for a Smart Home application. The task in this data set is to predict whether or not the room is occupied by people. Each of the five attributes are continuous measurements. The label is 0 if the room is unoccupied, and a 1 if it is occupied by a person. This data set comes the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+

4. **opticalDigit.csv**: A data set of optical character recognition of numeric digits from processed pixel data. Each instance represents a different 32x32 pixel image of a handwritten numeric digit (from 0 through 9). Unlike MNIST from Homework 1, each image was preprocessed into a smaller number of attributes. Each image was partitioned into 64 4x4 pixel segments and the number of pixels with non-background color were counted in each segment. These 64 counts (ranging from 0-16) are the 64 attributes in the data set, and the label is the number from 0-9 that is represented by the image. This data set is more complex than the Monks1 data set, but still contains only nominal attributes and a nominal label. This data set comes from the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

As in Homework 1, the file format for each of these data sets is as follows:

- The first row contains a comma-separated list of the names of the label and attributes
- Each successive row represents a single instance
- The first entry (before the first comma) of each instance is the label to be learned, and all other entries (following the commas) are attribute values.
- Some attributes are strings (representing nominal values), some are integers, and others are real numbers. Each label is a string.

Program

Your assignment is to write a program called **tree** that behaves as follows:

1) It should take as input four parameters:
    a. The path to a file containing a data set (e.g., opticalDigit.csv)
    b. The percentage of instances to use for a training set
    c. A random seed as an integer
    d. Either True or False indicating whether we should handle numeric attributes as numeric (if False, then we treat them as categorical values)

    For example, if I wrote my program in Python 3, I might run

    ```
    python3 tree opticalDigit.csv 0.75 12345 True
    ```

    which will learn a decision tree for opticalDigit.csv with a random seed of 12345, where we treat numeric attributes as numeric (rather than categorical) and 75% of the data will be used for training (the remaining 25% will be used for testing)
2) Next, the program should read in the data set as a set of instances, which should be split into training and test sets (using the random seed input to the program). Note: you do not need to use a validation set in this assignment, nor do you need to pre-process the data as we did in Homeworks 2 and 3
3) The training set should be fed into the ID3 learning algorithm to construct a decision tree fitting the training data. In addition to ID3, you should also implement the ability to

handle numeric attributes from C4.5 and CART (where a threshold is found so that we filter all instances whose value is less than or equal to the threshold to the left branch of the node and we filter all the other instances whose values are greater than the threshold to the right branch of the node). Whether you use that numeric handling is determined by the last parameter passed in by the user (if they pass in False, you should treat the attribute as categorical, instead).

4) The learned decision tree should be used to predict labels for all of the instances in the test set created in Step 2 in order to evaluate the learning performance.
5) A confusion matrix should be created based on the predictions made during Step 4, then the confusion matrix should be output as a file with its name following the pattern: results_tree_<DataSet>_<Numeric?>_<TrainingPercent>_<Seed>.csv (e.g., results_tree_monks1_True_0.75_12345.csv).

Please note that you **are** allowed to reuse your code from Homework 1 for generating random test/training sets, as well as for creating output files.

Program Output

The file format for your output file should be the same as in Homework 1. Please refer back to that assignment for more details.

Programming Languages

I would recommend using either the **Java** or **Python** programming languages to complete this assignment. If you have a different preferred language, please talk to me to make sure that I will be able to run your submission in that language.

As in Homeworks 2-3, **you are allowed to use external libraries** (e.g., Pandas and numpy in Python) for things like reading in the data (no preprocessing necessary), although there is less need for external libraries in this assignment. To receive ***full credit*** on the assignment, you should *not* use any pre-created implementations of decision trees (e.g., using scikit-learn in Python), but instead implement your own from scratch.

However, if you have difficulties getting the decision trees to work**,** you *can* use a pre-created implementation to answer the research questions. If you do so, please make sure to cite your source in the README file.

Questions

Please use your program to answer these questions and record your answers in a README file:

1) Pick a single random seed and a single training set percentage (document both in your README) and run your program on each of the four data sets. You should pass in True as the final parameter to your program to treat all numeric attributes as numeric.
    a. What is the accuracy you observed on each data set?
    b. Calculate a 95% confidence interval for the accuracy on each data set.
2) Create an image of the tree that your program learned in Question 1 for the monks1.csv data set (you can draw by hand and scan your image into a PDF, or you can use a drawing program to create an image file). Make sure to upload your image to GitHub.
    a. What are the rules learned by the algorithm?
    b. How do these rules compare to the true rules in the data set (described on page 1 of the assignment)?
3) Using the same seed and training set percentage from Q1, rerun your program on the opticalDigit.csv data set and pass in False for the final parameter so that your algorithm treats each attribute as categorical values (instead of numeric):
    a. What is the accuracy you observed?
    b. Calculate a 95% confidence interval around that accuracy
    c. Compare the confidence intervals from your answer to Q1b and Q3b. What do you observe? What does this imply?
4) Choose 9 new seeds (document in your README). Rerun your program on opticalDigit.csv using these 9 new seeds using both True and False as the final parameter to the program.
    a. Calculate the average accuracy across the 10 seeds when you treated the attributes as (1) numeric and (2) categorical
    b. Did you observe the same trends as in Q3c? That is, if one approach achieved a statistically significantly higher accuracy in Q3c, did the same approach achieve a higher accuracy when averaged over 10 seeds? If they were not statistically significantly different in Q3c, are the averages very close?
    c. Did these averages fall in your confidence intervals calculated in Q1b and Q3b?

README

Within a README file, you should include:

1) Your name(s),
2) Your answers to the questions above,
3) A short paragraph describing your experience during the assignment (what did you enjoy, what was difficult, etc.)
4) An estimation of how much time you spent on the assignment, and
5) An affirmation that you adhered to the honor code

Please remember to commit your solution code, results files, and README file to your repository on GitHub. You do not need to wait to commit your code until you are done with the assignment; it is good practice to do so not only after each coding session, but maybe after hitting important milestones or solving bugs during a coding session. ***Make sure to document your code***, explaining how you implemented the different components of the assignment.

Honor Code

Each student is allowed to work with a partner to complete this assignment. Groups are also allowed to collaborate with one another to discuss the abstract design and processes of their implementations. For example, please feel free to discuss the pseudocode for the learning algorithm. However, sharing code (either electronically or looking at each other's code) between groups is not permitted.

Grading Rubric

Your solution and README will be graded based on the following rubric:

Followed input and output directions: /5 points
Properly read in and split the data into training and test sets: /5 points
Correctly implemented the ID3 learning algorithm: /20 points
Correctly implemented the ability to handle numeric attributes as numeric: /15 points
Correctly implemented classification: /10 points
Correctly answered the research questions: /35 points
Provided requested README information: /5 points
Appropriate code documentation: /5 points