CARNEGIE MELLON UNIVERSITY

10601a: Homework #6 - “Decision Trees”

TA-in-charge:
Seth Flaxman (sflaxman@cs)

Assigned: 11 February 2014.
Due: 11:59pm on 17 February 2014.
Late Penalty: 25% per day.
This assignment is current as of 1:17pm on Tuesday 11th February, 2014.

Policy on Collaboration among Students

The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes are shared, or are taken at that time, and provided learning is facilitated, not circumvented. The actual solution must be done by each student alone, and the student should be ready to reproduce their solution upon request. The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved, on the first page of their assignment. Specifically, **each assignment must contain a file named collaboration.txt where you will answer the following questions**:

- Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered ‘yes’, give full details? (e.g.“Jane explained to me what is asked in Question 3.4”).
- Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered ‘yes’, give full details? (e.g. “I pointed Joe to section 2.3 to help him with Question 2”).

Collaboration without full disclosure will be handled severely, in compliance with CMU?s Policy on Cheating and Plagiarism. All violations (even first one) will carry severe penalties, up to failure in the course, and will in addition always be reported to the university authorities.
Some of the homework assignments used in this class may have been used in prior versions of this class, or in classes at other institutions. Avoiding the use of heavily tested assignments will detract from the main purpose of these assignments, which is to reinforce the material and stimulate thinking. Because some of these assignments may have been used before, solutions to them may be (or may have been) available online, or from other people. It is explicitly forbidden to use any such sources, or to consult people who have solved these problems before. You must solve the homework

assignments completely on your own. I will strictly enforce this policy, and if a violation is detected it will be dealt with harshly. Collaboration with other students who are currently taking the class is allowed, but only under the conditions stated above.

General Instructions

The goal of this assignment is for you to implement a decision tree learner, entirely from scratch. We're going to try decision trees in two domains. For simplicity, all variables are discretized into just two categories. The datasets for this assignment are available at https://autolab.cs.cmu.edu/10601a-s14/homework6/handout (alternatively, go to Homework 6 on autolab and click on "Options" -> "Download handout").

The first task is to predict whether a song was a "hit" meaning it made it onto the Billboard Top 50—each instance has a label **hit** equal to "yes" or "no". Attribute names are listed in bold; check the csv file to see their possible values: **year** of release, **solo** recording or band, **vocal** or instrumental, **length** of recording (< 3 minutes or > 3 minutes), **original** composition or a "cover", **tempo**, **folk** song, **classical** piece, **rhythm** and blues, **jazz**, **rock** and roll.

The second task is to predict the final **grade** (A, not A) for high school students. The attributes (covariates, predictors) are student grades on 5 multiple choice assignments **M1** through **M5**, 4 programming assignments **P1** through **P4**, and the final exam **F**. Again, check the csv files to see the attribute values.

Before you begin, think about whether decision trees are appropriate for these two tasks. Write down your thoughts in a text file `Q0.txt`: do you think a decision tree will work for the music dataset? What about for the education dataset?

Next, inspect the training data manually; look for any unusual findings and think about which variables seem useful. In `Q1_music.txt`, make your best guess about which variables are useful for the music task. Do the same in `Q1_education.txt`.

You may use Java or python to complete this assignment. (Throughout, examples are shown assuming you're using python, but Java is fine.)

We've provided you with attributes and labels split into training and testing data in files "music*.csv" and "education*.csv". Throughout, we show results for "example1.csv" and "example2.csv," a small, purely for demonstration version of the music dataset. The format is comma separated, one row per observation, one column per attribute. Your program will take a training and a testing dataset as input.

1 Warmup

Write a program `inspect.py` to calculate the label entropy at the root (i.e. the entropy of the labels before any splits) and the error rate (the percent of incorrectly classified instances) of classifying using a majority vote.

```
$ python inspect.py example1.csv
entropy: 0.981
error: 0.42
```

Test your program on both datasets—this error rate is a baseline over which we'd (ideally) like to improve.

2 Training the Tree

Implement a decision tree learner with the following guidelines. (As a reference, consult Mitchell, Chapter 3, page 56.)

- Use mutual information to determine which attribute to split on.
- Be sure you're correctly weighting your calculation of mutual information. For a split on attribute X, $I(Y;X) = H(Y) - H(Y|X) = H(Y) - P(X=0)H(Y|X=0) - P(X=1)H(Y|X=1)$. Equivalently, you can calculate $I(Y;X) = H(Y) + H(X) - H(Y,X)$.
- As a stopping rule, only split on an attribute if the mutual information is $\geq .1$.
- Do not grow the tree beyond depth 2. Namely, split a node only if the mutual information is $\geq .1$ and the node is the root or a direct child of the root.
- Use a majority vote of the labels at each leaf to make classification decisions.

Hints on getting started: write helper functions to calculate entropy and mutual information. Write a function to train a stump (tree with only one level). The correct tree and output format for the example data are shown below, where we are training on example1.csv and testing on example2.csv. For the music data, use "+" for **hit** = "yes" and "-" for **hit** = "no". With the education data, use "+" for final **grade** = "A" and "-" for final **grade** = "not A". Don't worry about the order in which you list the left and right children, the autograder will take care of it.

```
$ python decisionTree.py example1.csv example2.csv
[58+/42-]
love = yes: [46+/11-]
| debut = yes: [29+/0-]
| debut = no: [17+/11-]
love = no: [12+/31-]
| debut = yes: [12+/11-]
| debut = no: [0+/20-]
error (train): 0.22
error (test): 0.48
```

Remember, the tree might not be full. Here's what happens when we train on example2.csv and test on example1.csv:

```
$ python decisionTree.py example2.csv example1.csv
[28+/72-]
love = yes: [27+/25-]
| debut = yes: [26+/0-]
| debut = no: [1+/25-]
love = no: [1+/47-]
error (train): 0.02
error (test): 0.29
```

The numbers in brackets give the number of positive and negative labels in that part of the tree. The last two numbers are the error rate on the training data and the error rate on the testing data.

3 Evaluation

Train and test a decision tree for the music dataset and the education dataset. Which is more accurate on the training data? Which is more accurate on the testing data? Write down your observations in a text file Q3.txt and include it with your submission.

4 Autolab Submission

Submit a .tgz containing your source code, written assignments, and a file collaboration.txt). You can create that file by running " *tar -cvf hw6.tgz *.py *.txt*". **DO NOT** put the above files in a folder and then tar gzip the folder. You must submit this file to the "homework6" link on Autolab.