CIS 472/572, Winter 2018

Homework 1: Decision Trees

Supplementary Information

1 About the Chi-Squared Test

The basic idea is that you're testing to see if the distributions over y are different after you split on x_i. If they're pretty much the same, then your decision tree may just be fitting random noise.

For details of the Chi-squared test, see pages 93-94 of the classic ID3 paper: `http://dept.cs.williams.edu/~andrea/cs374/Articles/Quinlan.pdf`

This should tell you how to compute the Chi-squared statistic for a decision tree split. Note that, with binary data, you always have $v = 2$, which leads to a 1-dimensional Chi-squared test.

Wikipedia also has information on the Chi-squared test: `http://en.wikipedia.org/wiki/Pearson%27s_chi-squared_test`

After you compute the Chi-squared statistic, you need to compare it to a number in order to get a p-value. You can find critical values here: `http://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm`

For example, to get a p-value of 0.01, use a value of 6.635. If you get a number less than 6.635, then reject the split and stop recursing. This will stop splitting when there's more than a 1% chance (according to the assumptions of the Chi-squared test) that the observed leaf statistics would be generated from the same probability distribution.

2 Model Accuracies

Here are the results I get from my implementation, both with a Chi-squared test (using a critical value of 6.635) and without (continuing until information gain is zero). All accuracies are reported on the test data.

- Dataset 1: 76.85% with Chi-squared test / 75.50% without
- Dataset 2: 71.83% with Chi-squared test / 72.33% without

Model files with Chi-squared tests are `ds1-chi2.model` and `ds2-chi2.model`, and model files without are `ds1-full.model` and `ds2-full.model`. All models are available here: `https://www.cs.uoregon.edu/Classes/15W/cis472/example-models.zip`

3 Information Gain for Dataset 1

If you're not getting results similar to mine, you'll need to do some debugging. One possible error is that your information gain computation is wrong. To help you debug, here's the information gain for each attribute at the root node, on example dataset 1 (`training_set.csv`). If you get similar results, then your information gain computation is probably working correctly! (Note: You do not need to print out this info – this is just to help you debug.)

```
XB: 0.000983
XC: 0.004382
XD: 0.005153
XE: 0.002901
XF: 0.000072
XG: 0.001815
XH: 0.004647
XI: 0.015922
XJ: 0.001613
XK: 0.004724
XL: 0.001356
XM: 0.008296
XN: 0.003214
XO: 0.021075
XP: 0.002636
XQ: 0.003576
XR: 0.006988
XS: 0.002320
XT: 0.008286
XU: 0.006303
```

4 Mushrooms Dataset

Finally, here's a real dataset, just for fun! Mushrooms is a classic dataset where the goal is to predict if each mushroom is poisonous or not. Here's more information about the dataset: `http://archive.ics.uci.edu/ml/datasets/Mushroom`

I converted each multi-valued attribute into a set of binary-valued attributes so that it works with your decision tree learners. The class label is 1 if the mushroom is poisonous and 0 otherwise. I selected the first 6000 examples as the training set, and used the remaining 2125 examples as the test set. Training and test files are on the web site here: `https://www.cs.uoregon.edu/Classes/15W/cis472/mushrooms.zip`.

I get 97.93% accuracy on the test set (both with a Chi-squared threshold of 0 and 6.635). Depending on the exact train/test split, it's possible to get 100% accuracy on this dataset.

Take a look at the decision trees you learn and you'll have some idea about what differentiates poisonous mushrooms from edible ones.