EECS 440: Programming Assignment 1
Decision Tree Learning

Connor Wolfe, Jonathan Yau

September 2017

Introduction

This project describes an implementation of the ID3 algorithm for decision tree learning for the binary classification problem. The algorithm follows the following Pseudocode:

If all examples positive: return single node tree, label = +
If all examples negative: return single node tree, label = -
If attributes empty, return single node tree root, label = most common label
Otherwise begin:
A <- attribute from Attributes that best classifies examples (IG or IGR)
Decision Attribute for Root <- A
For each value vi of A
Add new branch below Root corresponding to test A = vi
Let examples_vi be subset of examples with val vi for A
If examples_vi empty
Below branch add leaf with label = most common val of target_attribute in examples
Else
Below branch add subtree ID3(examples_vi, target_attributes, Attributes - A)

Analysis

Note that all experiments are conducted using information gain ratio and not information gain in order to avoid overfitting.

What is the CV accuracy of the classifier on each dataset when the depth is set to 1?

When the maximum depth is set to 1, the root node can only perform one split, on the best attribute. Therefore the performance of the tree is entirely

dependant on how well this attribute splits the tree. It is important to note that this is the best attribute, so the difference in accuracy from depth 0 to depth 1 is greater than any subsequent depths. It is feasible that this split will capture the a large portion of the accuracy gains the tree would obtain if it were grown to full length, as further splits become increasingly insignificant (and overfitted potentially).

Voting

When voting is grown to depth 1, the accuracy averages to 98.40%, with a deviation of +/ 2.5%. The tree splits on feature 1 'Repealing-the-Job-Killing-Health-Care-Law-Act' first, which has 3 nominal values: '-', '0', '+'. Therefore the tree has a size of 4 which makes sense. As explained above, splitting on just this one attribute proved to be highly effective. Growing the full tree produces accuracy of 96.82% so we see that we eventually overfit and splitting on just the first attribute us preferable. Of course with a pruning algorithm we may find a more optimal depth in between 1 and the full tree. Compared with the majority class classifier, which yields accuracy of 55.40%, we see that splitting on this first attribute is highly effective

Volcanoes

When volcanoes is grown to depth 1, the accuracy averages to 73.32% with a deviation of +/- 2%. It selects the first feature, 'image_id' which has 56 values, one of them unrepresented, producing a tree of size 56. Though this data produced a less accurate tree than voting, it proves more accurate than growing a full tree which yields accuracy of 71% (overfitting). The majority class classifier, however, produces an accuracy of 55.17%, so we see that splitting on the first attribute is an immense improvement, and actually quite close its optimal depth.

Spam

When spam is grown to depth 1, the accuracy averages to 66.39% with deviation of +/- 1%. It selects the sixth feature, 'OS' which has 6 values producing a tree of size 7. We can again compare this to the accuracy of the full tree which is 80% and see that splitting on just the first feature does not provide a nearly optimal classification. The majority class classifier produces accuracy of 62.53% so it is not a large improvement to split by feature 'OS'.

For spam and voting, look at first test picked by your tree. Do you think this looks like a sensible test to perform for these problems? Explain.

Voting

Voting will split on feature 1, 'Repealing-the-Job-Killing-Health-Care-Law-Act', which has 3 nominal values: '-', '0', '+'. When we sort the data by feature 1, we notice that the $dataset_{x=-}$ is pure, with label = 1. The $dataset_{x=0}$ is quite mixed, but only represented 6 times (which decreases its effect when using gain ratio), and finally the $dataset_{x=+}$ is almost pure with 242 label = 0 and only 3 with label = 1. Other features, namely feature 3, does seem to split quite well as well, but from simply looking at the data, attribute 1 seems very sensible

Spam

Spam splits first on feature 6, 'OS', which has 6 values. This is the only nominal feature in the data set and it seems reasonable to first split on a nominal feature and then the continuous because we split the data into 6 groups rather than 2. Both volcanoes and spam split their only nominal feature first, which intuitively makes sense since it has the chance of being more effective by making more bins, depending on the data set of course. When we look at the dataset with respect to feature 6, we see it splits for values Windows and Linux quite well, with 70% purity, and these two values comprise 62% of the data. The rest of the values are more mixed. Given though that the tree maximally splits 80% this split does not make much progress in that direction.

For volcanoes and spam, plot the CV accuracy as the depth of the tree is increased. On the x-axis, choose depth values to test so there are at least five evenly spaced points. Does the accuracy improve smoothly as the depth of the tree increases? Can you explain the pattern of the graph?

Volcanoes

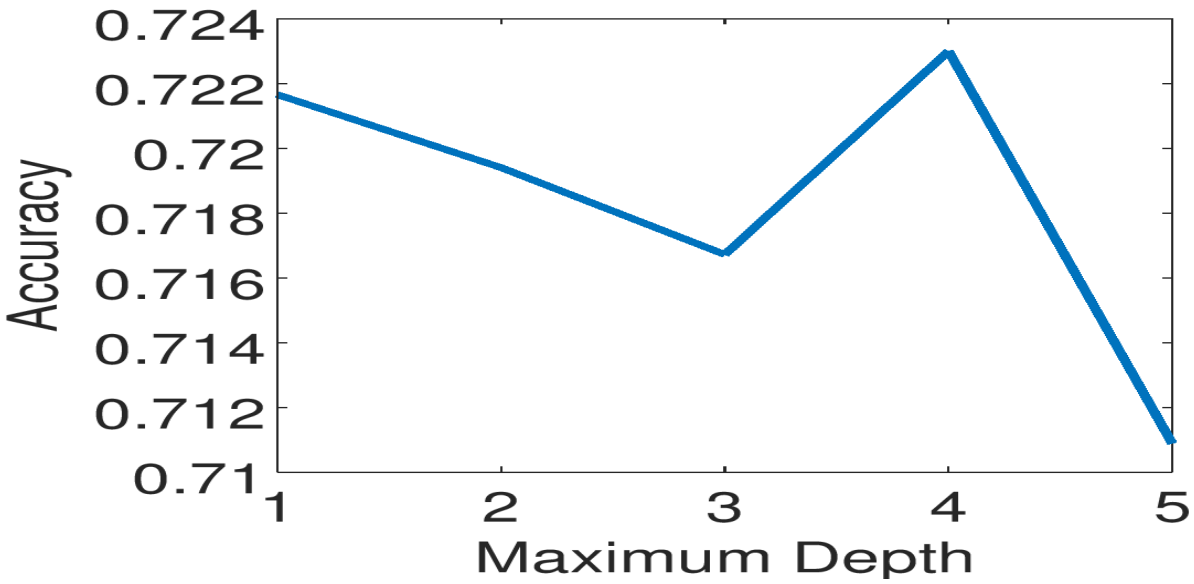

Figure 1: A plot of the how the CV accuracy varies with an increasing depth for the volcanoes set. Note this tree grows a maximum depth of 7 so we increment depth by 1

The accuracy for the tree at depth 1 is 72.2%. As we increase the depth, the accuracy decreases before jumping to it's maximum at maximum depth of 4. It then goes back down for 5. We believe that this shape is caused by our program underfitting the data between 1 and 3, fitting the data better at 4, and then overfitting at maximum depth of 5.

Spam

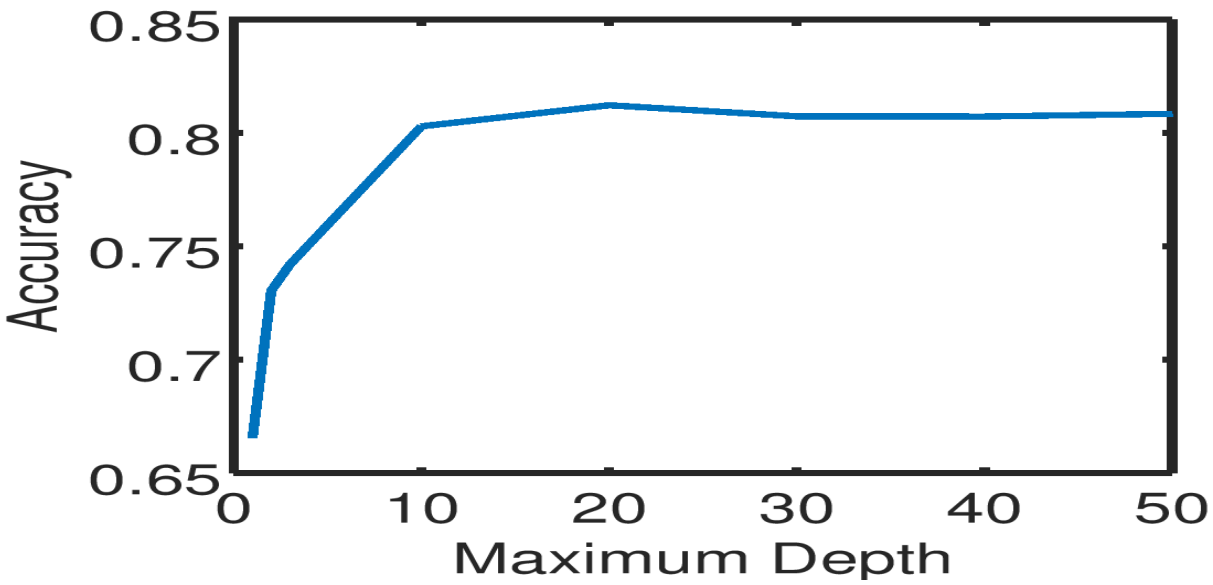

Figure 2: A plot of the how the CV accuracy varies with an increasing depth for the spam set. Note this tree grows a maximum depth of 55 so we increment by 10

For the spam data, the accuracy smoothly increases as we increase the depth, before leveling off to around 81% accurate at a depth of 15. This data set does not suffer from overfitting, though at a certain point, growing the tree becomes computationally expensive with little reward

Pick 3 different depth values. How do the CV accuracies change for gain and gain ratio for the different problems for these values?

Information gain is a means of measuring the expected reduction in entropy when a dataset is split about a feature. The information gain about feature X is computed as

$$IG(X) = H(Y) - \sum_{vinValues(A)} P(X = v)H(Y|X = v)$$

. Information gain can be a misleading metric though, because it favors attributes that have many values, though they possibly have no connection to the output label. This frequently causes overfitting which leads to the use of information gain ratio. This will penalize attributes depending on how broadly and

uniformly they split the data using a term called split information. Information gain ratio is computed as follows

$$IGR(X) = IG(X)/SplitInformation(X)$$

Voting

For the voting dataset, the CV accuracy values do not have a large variance as depth or IG vs IGR vary, and stay close to 97%. We chose to analyze the tree of depth 1, the tree of maximum difference in accuracy (depth = 4) and the full tree. Depth = 1:
IG = 98.42%
IGR = 98.86%
Depth = 4:
IG = 98.40%
IGR = 97.74%
Depth = 11 (full tree):
IG = 96.84%
IGR = 97.30%

We do not see such a large difference between the two values which makes sense since IGR only diverges from IG when attributes are splitting the data in many terms nonuniformly. We do not expect to see that in voting because all features have only three values, leading to more even distributions.

Volcanoes

For the volcanoes dataset, we chose to analyze the tree of depth 3, of depth 4 and the full tree (which had the maximal difference). Depth = 1:
IG = 72.48%
IGR = 72.57%
Depth = 4:
IG = 72.10%
IGR = 72.66%
Depth = 7 (full tree):
IG = 71.04%
IGR = 71.90%

We again do not see such a large difference between IG and IGR. This makes sense because most features are continuous for volcanoes, so these features partition the data into groups of 2 and IG and IGR will have similar values. The only feature which is a likely candidate for divergence between IG and IGR is feature 1 which is nominal containing 56 values. Here, it is possible some of these 56 new data sets are sparse and the SplitInformation term would penalize this. However, for both IG and the IGR, feature 1 yielded the maximum value and this feature is chosen as the first split every time, despite the probable difference between the two computations. We therefore see the two trees chose very similar attributes, despite having small differences in IG and IGR leading to trees with very similar accuracies.

Spam

For the spam dataset, we chose to analyze the tree of depth 3, of depth 4 and the full tree (which had the maximal difference). Depth = 1:

IG = 72.48%

IGR = 72.57%

Depth = 4:

IG = 72.10%

IGR = 72.66%

Depth = 7 (full tree):

IG = 71.04%

IGR = 71.90%

We again do not see such a large difference between IG and IGR. This makes sense because most features are continuous for volcanoes, so these features partition the data into groups of 2 and IG and IGR will have similar values. The only feature which is a likely candidate for divergence between IG and IGR is feature 1 which is nominal containing 56 values. Here, it is possible some of these 56 new data sets are sparse and the SplitInformation term would penalize this. However, for both IG and the IGR, feature 1 yielded the maximum value and this feature is chosen as the first split every time, despite the probable difference between the two computations. We therefore see the two trees chose very similar attributes, despite having small differences in IG and IGR leading to trees with very similar accuracies.

Compare the CV accuracies and the accuracy on the full sample for depths 1 and 2. Are they comparable?

	Voting	Volcanoes	Spam
CV Depth 1	0.984	0.7212	0.6639
CV Depth 2	0.9864	0.7042	0.7316
Whole Depth 1	0.90886	0.7297	0.6639
Whole Depth 2	0.9932	0.8669	0.7323

Figure 3: A table representing CV accuracies and full tree accuracies for depths 1 and 2 of all data sets

Each set of values are comparable with an exception in volcanoes. With maximum depth of 1, the accuracies for the cv vs whole set for each of the datasets were different by less than 0.01.

Voting

Voting needs very few nodes to reach high accuracy normally. Thus, it is not surprising that maximum depth of 1 or 2 regardless of using the whole or CV always had an accuracy greater than 0.98. The difference for the maximum depth of 2 were more accurate but the test run on the whole set had a higher accuracy increase by 0.03 than its CV counter part.

Volcanoes

For maximum depth of 1, the accuracies were essentially the same, the whole set test was on 0.08 more accurate. When running on the maximum depth of 2, volcanoes seemed to be underfitting the data (at least that's what question c also appears to imply) as its accuracy dropped by 0.02. On the whole set, the accuracy jumped to the accuracy of 0.86. This is the only time where the accuracies were not comparable.

Spam

Spam's maximum depth tests had the same accuracy. This is not surprising as spam usually has a maximum depth of 50 when run on its own. A maximum depth of 1 gives very little information about how to classify when dealing with such a large data set. With maximum depth of 2, the CV and the whole set continue to have very similar values (around 0.732). This shows that the larger the dataset, the more similar the accuracy would be in comparison to using a large portion of the data set.