Paul Johnston

CS 478

Decision Tree

1. This Decision Tree learner uses the ID3 decision tree algorithm and is able to handle unknown data. It uses standard information gain as it's basic attribute evaluation metric. It uses a node class with the capabilities of making a tree structure. Each node has children, data (features and labels), connecting attribute (meaning the attribute that connects it to its parent), connecting value, (instance of the connecting attribute possible values),remaining attributes, and depth. When training, this learner first computes the base info possible from just dividing the data into the possible output classes of the given node. It then calculates what the info gain would be if it split on each of the remaining attributes. It then creates children nodes based on the attribute that provides the best info gain (base info – attribute info). It then recursively splits on all the children nodes until there are either no more attributes to split on or the child nodes are pure, meaning all their data is of the same output class. In addition to this, the learner uses reduced error pruning to avoid overfitting to the training data. On the lenses data, it is able to predict with an average of 75 percent accuracy

2.

Cross validation results

	Cars dataset		Voting dataset	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
Fold 0	1.0	0.9651162790697675	1.0	0.9767441860465116
Fold 1	1.0	0.9248554913294798	1.0	0.9545454545454546
Fold 2	1.0	0.9364161849710982	1.0	0.8604651162790697
Fold 3	1.0	0.930635838150289	1.0	0.9545454545454546
Fold 4	1.0	0.9017341040462428	1.0	0.9302325581395349
Fold 5	1.0	0.8953488372093024	1.0	0.9318181818181818
Fold 6	1.0	0.9364161849710982	1.0	0.9069767441860465
Fold 7	1.0	0.9190751445086706	1.0	0.9545454545454546
Fold 8	1.0	0.9421965317919075	1.0	0.9302325581395349
Fold 9	1.0	0.9248554913294798	1.0	0.9318181818181818
Average		0.9276650087377336		0.9331923890063425

The table above shows this decision tree's accuracies for 10-fold cross validation. The learner got 100% accuracy for each dataset when training, this is because it would basically memorize the training data, splitting the tree until the nodes are pure. This means that even when the data is shuffled, the tree will still send the same instance down the same path of the tree, resulting in the same output class. I observed from the table above that the fold accuracies were consistent except for fold 5 for the cars dataset. It seemed like a fluke, as it was very close to 90 percent. The other outlier was fold 2 for the voting dataset which I think happened because when it trained on this fold, it happened to grab a set of training data that was more clumped together from the testing set. It trained a lot on the clumped training set and then when it saw novel data that didn't follow the same clumping as the training, it just had to take the best output possibility from the parent node (because there were no children nodes with the attribute value as the novel instance).

3. What the model learned

For the cars dataset, the decision tree learned that the best attribute to split on at the beginning was attribute 5 (zero indexed) which was safety. It then learned that for cars with high and medium safety, the best attribute to split on was persons. For cars with low safety, it didn't have to learn anymore because all cars with low safety in the training set were

unacc. The decision tree continued recursing like this on the nodes, for high safety cars which fit 2 persons, it didn't have to learn anymore because all cars within these parameters were unacc as well. One of the more shallow, uninformative features was the lug boot attribute.

For the vote dataset, the decision tree learned that the best attribute to split on at the beginning was the "physician fee freeze" question. For people who said no to this question, the next most informative question was "adoption of the budget". For people who said yes to the "physician fee freeze" question, the next most informative question was "synfuels corporation cutback". For people who didn't respond the "physician fee freeze" question, the next most informative question was "mx missile". One of the shallowest, least informative question was the education spending question and the water project cost sharing question.

4. Unknown values

This decision tree learner handles unknown attributes by creating a new category of attribute values. This allows the tree to, instead of just ignoring these values, infer knowledge from the missing value. For example, in the voting set, not voting for a certain issue may mean that the person who chose not to vote was leaning one way politically. I chose this approach because it allowed the learner to retain information from the unknown values and the numpy unique function allowed for this implementation very nicely.

5. Results with reduced error pruning

Cars dataset						
	Before pruning			After pruning		
	Nodes	Depth	Accuracy	Nodes	Depth	Accuracy
Test 0	277	6		111	5	0.8703703703703703
Test 1	308	6		121	5	0.8912037037037037
Test 2	269	6		129	5	0.8819444444444444
Test 3	295	6		117	5	0.8634259259259259
Test 4	297	6		120	5	0.8796296296296297
Test 5	291	6		123	5	0.8958333333333334
Test 6	260	6		122	5	0.8888888888888888
Test 7	308	6		139	5	0.9074074074074074
Test 8	284	6		114	5	0.8657407407407407
Test 9	272	6		101	5	0.8634259259259259
Average	286.1	6	0.9276650087377336	119.7	5	0.880787037037

Vote dataset						
	Before pruning			After pruning		
	Nodes	Depth	Accuracy	Nodes	Depth	Accuracy
Test 0	39	7		10	3	0.944954128440367
Test 1	35	6		6	1	0.944954128440367
Test 2	36	7		13	4	0.944954128440367
Test 3	35	6		17	6	0.9357798165137615
Test 4	39	6		4	1	0.963302752293578
Test 5	41	6		9	1	0.9541284403669725
Test 6	39	6		15	4	0.9541284403669725
Test 7	31	7		16	4	0.926605504587156
Test 8	34	6		10	3	0.8990825688073395
Test 9	36	5		9	3	0.944954128440367
Average	36.5	6.2	0.9331923890063425	10.9	3	0.9412844036697

It is interesting to note that when using early stop pruning with the cars dataset, the generalization accuracy decreased by four percent. I think this is because it took out a hundred nodes or more making the model essentially learn less. The effect of pruning could also be explained by the fact that there were nearly 2,000 instances of data making the combinations of attributes more varied. Also, the depth never changed by more than one level, which I'm guessing is because if we were to take one more level off of a model trained on such an expansive dataset, the info learned would decrease too much for our validation testing to allow.

The vote dataset is interesting because the depth changed by up to 5 levels sometimes and the number of nodes decreasing by nine tenths, but with the accuracy improving significantly. I think that with this smaller dataset, when not pruning, the model was able to memorize the combinations of votes more easily than the cars dataset, and was therefore overfitting more than the model trained on the cars dataset. With reduced error pruning, the model trained on the vote set improved by one percent overall.

The validation set was twenty percent of the training set in both these instances. It was then not used for training, and after training the model on the training data, the validation set was used to check if the MSE on the validation set was less when taking off leaf nodes. If there was a decrease of MSE on the validation set when pruning child nodes, those nodes stayed pruned off, if there was an increase of the MSE on the validation set after pruning child nodes, those nodes were kept on the tree.

6. Creative Experiment

For my creative experiment, I chose at random which attribute to split on and then applied reduced error pruning to the model. Because finding the shallowest tree takes exponential time, initially a greedy approach was taken to determine which attribute to split on. Instead I used a random approach with reduced error pruning to see what changes this would make to the accuracy, node count, and tree depth. Below are the results from ten trials on the cars dataset.

Cars dataset, random attribute splitting			
	Nodes	Depth	Accuracy
Test 0	191	5	0.7314814814814815
Test 1	127	5	0.7037037037037037
Test 2	144	5	0.7106481481481481
Test 3	161	5	0.8024691358024691
Test 4	149	5	0.7708333333333334
Test 5	161	5	0.7083333333333334
Test 6	161	5	0.7361111111111112
Test 7	120	5	0.7824074074074074
Test 8	138	5	0.6944444444444444
Test 9	170	5	0.7800925925925926
Average	152.2	5	0.7420524691358

What's interesting to see is that with random attribute splitting, the accuracy decreased by fourteen percent. A decrease was expected, but I was expecting more along the lines of twenty five percent, because there are four output classes for the cars dataset and one over four is twenty five percent. It is interesting to note that even when blindly choosing attributes, the decision tree model is able to correctly identify data with close to seventy five percent accuracy. The node count and tree depth stayed about the same, the node count had a slight increase. To compensate for the lack of judgement, the pruning had to keep more nodes than when using info gain as the metric for measuring attribute strength.