# CS3600 P4 Writeup

Paras Jain

April 2016

# 1 Q6: Performance

## 1.1 Dummy Dataset 1

This dataset had 100% test performance with a relatively small tree at size 3 (compared to a training set of size 20).

It makes perfect sense that this dataset would produce 100% test performance on the dataset as looking through the dataset, it seems the target hypothesis is simply $\neg A_5$. As decision trees can learn arbitrary binary expressions, this simple expression is trivial to construct.

As decision trees have the inductive bias of finding shorter trees (as they select attributes with maximal entropy), this produces a short tree. Given how simple the target hypothesis is, this is why a short tree is generated.

## 1.2 Dummy Dataset 2

This dataset had 65% test performance with a moderately small tree at size 11 (compared to a training set of size 20).

The worse test set performance on this dataset makes sense as looking through the dataset, there is no obvious correlation like with Dummy Dataset 1. There are too few training examples to learn the more complex hypothesis which is why information gain struggles. With only 20 training examples, the decision tree cannot generalize well to the test set.

The slightly larger tree makes sense as there are only 20 training examples. Information gain minimizes the height of the tree so only 11 nodes are required to represent the 20 training examples.

## 1.3 Connect4

This dataset had 76% test performance[1] with quite a large tree at size 41521 (compared to 67557 training instances).

The performance makes sense as the large tree produced is likely overfitting the training dataset. There is also significantly more variance and it seems that no one attribute correlates with the class label.

The tree size makes sense as the tree is not being pruned. Given the large number of dimensions of the Connect4 instances, it makes sense that a large tree would be produced.

## 1.4 Cars

This dataset had 94% test performance with a moderate sized tree of 408 (compared to 1728 training instances).

The performance is not surprising as looking through the data, the attributes seem to correlate well with class label - see the safety attribute where low safety cars or 2 person cars are unacceptable.

The tree size makes sense as there are several attributes that correlate strongly with the class label. Information gain selects these attributes to be near the top of the tree. Therefore, the relatively small tree makes sense.

[1] 10 trial average performance

### 1.5 Extra credit: Mushroom edibility

This dataset had 100% test performance[2] with a small tree of size 44 (compared to 8124 training instances).

The performance makes sense when looking at the generated tree - it seems the tree is so small as odor is at the root. Foragers often will determine if a mushroom is poisonous by smelling it and therefore it makes sense that the decision tree achieves such a high performance by primarily using odor. Moreover, the dataset is very clean and doesn't appear to have too much variance or missing data. In fact, in the mushroom dataset format file, 4 binary rules are presented which encompass 100% of the variance in the dataset. As decision trees can learn arbitrary binary functions, the 100% performance is not at all surprising.

The small tree generated makes sense for the same reason that performance makes sense as odor was the primary distinguishing factor of poisonous mushrooms. This allowed the tree to be very flat. As 4 simple binary expressions can represent this dataset, it is not surprising how short the tree is.

## 2 Q7: Applications

### 2.1 Cars

This could be used to make a "Tinder for car search" where car buyers where they could say whether they like or dislike a car by swiping left or right. A decision tree could be learned on this training data and be used to present the user with more cars they would like. The attributes close to the root of the tree would represent the attributes that are most significant for the car buyer. Perhaps Tinder already uses decision trees or other machine learning algorithm to better match people.

### 2.2 Connect4

The decision tree could be used to create a powerful heuristic for a search algorithm which can help a bot evaluate how close a position is to a winning state. This could be used with A* to make a powerful bot that uses the decision tree to guide the bot to stronger positions. Pruning to reduce overfitting would make this heuristic even more accurate and powerful.

### 2.3 Extra credit: Mushrooms

This dataset could be used to make an app that tells foragers whether a mushroom they found is poisonous or not. If you are hiking and you find a mushroom which you wonder is edible or not, you can load up the app which would traverse the decision tree learned and ask you for each attribute to evaluate the mushroom you found. For example, if I found a mushroom that had an almond smell, then the mushroom should be safe to eat.

The structure of the dataset as-is would work for this app where 22 attributes and 1 class attribute provide ample training data. The decision tree does not seem to have overfit the training set so it may even generalize somewhat to mushrooms that haven't been seen before.

[2] 20 trial average