Giuliano Cancilla

Intro to Artificial Intelligence

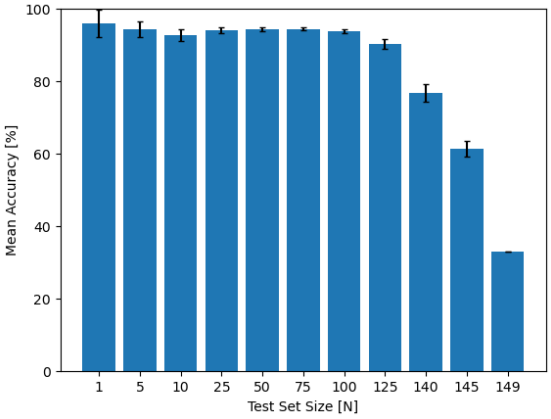
OLA3-ID3 Algorithm Analysis

ID3 Supervised Learning Algorithm

The ID3 algorithm is a supervised learning algorithm that is commonly used in classification problems. What ID3 does is allocate a tree called a decision tree. Essentially it is a tree that is trained on pieces of data as well as their outcomes. These pieces of data are known as attributes, and they define what a class label is. In this experiment, we work with two different datasets. One is the famous iris data set, that contains 4 attributes regarding length of pedals in a species of iris, as well as a classification of what species. The second is the Wisconsin Breast Cancer dataset, also known as the WDBC dataset, which includes 9 different measurements of a cell nucleus image of breast tissue and classifications of the tissue. So, the iris dataset has 4 attributes and a class label, and the WDBC dataset has 9 attributes with a class label. The goal of the ID3 Algorithm is to create a tree that finds an attribute with the maximum information gain (the attribute with the least amount of uncertainty) and make a classification on validation data that it hasn't seen before.

The algorithm uses a recursive function that recursively splits the dataset based off of the attribute with the highest information gain. Recursion is the idea of calling the function that you have defined within itself and shows a way to propagate nodes. Each node contained in the decision tree can be seen as a decision that the algorithm could make. Terminal nodes are final decision nodes, that propagate the final classification answer made by the tree for a specific subset of data and are determined when the base case of the recursive function is met. The base case fires if the probability of one of the class labels is 1 or if there are no more potential split points. By calling this function multiple times and returning a Node at the end we can hit that base case and propagate all the way back up to the original calling function, whilst returning nodes. Almost like leaving a trail of breadcrumbs back to the lowest depth of the tree, except the breadcrumbs are nodes that contain information. We first go to the farthest depth we can, and then propagate back up. We do this for each attribute.

Iris Data Set

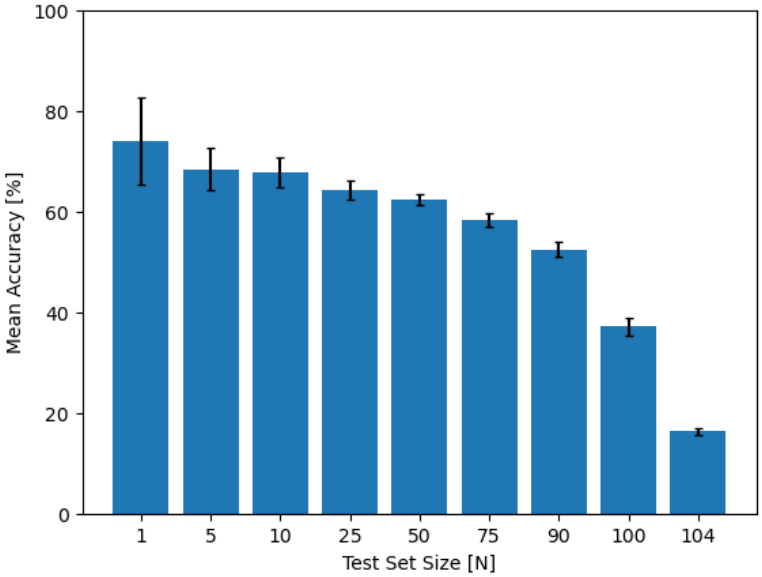


Iris Data Set ID3 predictions

The graph above shows the average accuracy of the classifications for the iris dataset using my implementation of the ID3 algorithm along the Y axis, and the amount of validation samples used. It is important to note that there are 150 examples in the iris data set, so if we look at the x value of 50, we know that we have a validation subset of 50 and a testing subset of 100. The algorithm ran well in general staying above 90% accuracy for most of the runs. As the number of validation examples (a.k.a test questions) increased, the data shows a hard dip as we approach from 140 validation examples to 149 validation examples. Since the algorithm is trying to find the best attribute with the highest information gain to make its guess as to what the classification is, it needs to be able to see more than just a few pieces of information before it can make its guess. The more validation examples you have, the less training the tree has and the harder a decision is to make. Although, we are right around 33% when the validation examples are 149 and the training data is just one example which is actually pretty good. The reason for this may be that it only knows one species of iris flower, so it always guesses that. Since there are 3 classes, that makes sense that it would be 33% since 33% of the data is one type of flower.

From looking at the graph we can see the outliers that measure how far the error is from the mean. As we look at the values where validation data is 1,5,10,140,145 we start to see more uncertainty in the answers that the tree is producing. With more training data it still performs better, however since there is so much to consider it may go through a process called overfitting. Overfitting is when there is either too much evidence or not enough evidence, which can lead to sporadic or uncertain behavior from the algorithm. When the validation data is 1 we can see that sometimes the algorithm works to classify all examples perfectly, but other times may be much worse than its closest relatives. Validation size 5 and 10 are getting better but still not quite too consistent. Then for validation 50,75, and 100 we hit the sweet spot, where ID3 is very certain of its choices. Again, we go down to validation 125,140,145 and we start to see diminishing return. Then we get to the maximum that the validation set can be at 149, and something interesting is happening here. The standard deviation looks to be zero. This means that there is no change in the algorithm's answers. This adds weight to the theory that if it only knows one classification, then it will always pick that classification, and in the case of the iris dataset, will be 33% correct every time.

Wisconsin Breast Cancer Dataset (WDBC)



Here we have the data from the runs that I did with the WDBC data. We can see right off the bat that it performed significantly worse on this data set than the iris data set. We see a continued trend from the iris data when we see the uncertainties at both ends of the spectrum. When we had smaller validation sets, the algorithm performed well, but had a higher deviation from the mean. When we look at the middle, we see a more stable deviation from the mean. Towards the end, we see more deviation until we hit the last column where the testing data was only one example, and we are completely certain because the algorithm only knows of one class label.

The reason why the algorithm didn't perform as well on the WDBC data could be because the WDBC has 9 attributes versus the iris set only having 4 attributes. Just by looking at the dataset of the WDBC you can observe that the numbers are much larger and vary from attribute to attribute. In my opinion I think the reason why it didn't perform as well is because it's a more complicated dataset than the iris dataset. Tree pruning could be added to this to optimize the strategies here for better results.

Work Cited

https://en.wikipedia.org/wiki/ID3_algorithm

Looked at a quick overview of the page to get a base idea and checked the images to try and visualize what the tree looks like.

https://numpy.org/devdocs/reference/index.html#reference

NumPy reference guide to look up what functions I could use in my code.

https://www.youtube.com/watch?v=coOTEc-0OGw&t=243s

YouTube video on ID3

https://www.datacamp.com/tutorial/decision-tree-classification-python

Great webpage on decision trees

https://scikit-learn.org/stable/modules/tree.html

Also just a good page

https://www.youtube.com/watch?v=IJDJ0kBx2LM

Skipped around this video on recursion

https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

extracting information of the values behind the cancer dataset

https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/

Article on decision trees

https://www.youtube.com/watch?v=j3lRFLPPwCQ

Video on decision trees, looked through it and was just trying to visualize what each step looked like.