ID3 Decision Tree

Elijah Dangerfield CSCI 4350

Introduction

In artificial intelligence, supervised learning is the machine learning technique in which a learning agent is given a labeled dataset in order to learn the mapping of features of the data to the labels of the data. Labels, simply put, are the answers to the question the supervised learning algorithm is solving. The idea of the supervised learning approach is that the algorithm will learn the relationship between the input features and the output answer such that it can discern the answers on unseen examples. A decision tree is a method of supervised learning in which the mapping of these inputs to outputs is defined through a tree of discrete questions about the data. These questions effectively filter the training data into smaller and smaller subsets until terminal node is reached at which point a classification is defined. For new data you can simply climb down the tree based on the example features answers to the questions at each node until a terminal node is reached. This idea is exemplified in figure one where the question is whether or not a person is fit based on features of this person. The goal of the tree is to create and order questions about the person in such a way that they lead to the correct classification. In this lab our goal was to build a decision tree in order to classify examples of two labeled datasets: the iris dataset (150 data points) and the cancer dataset (105 data points). In the iris dataset, the goal is to determine the type of iris flower (of 3 possible types) based on sepal width, sepal length, pedal width, pedal length. The goal of the cancer dataset is to determine the class of disease based on breast tissue sample attributes.

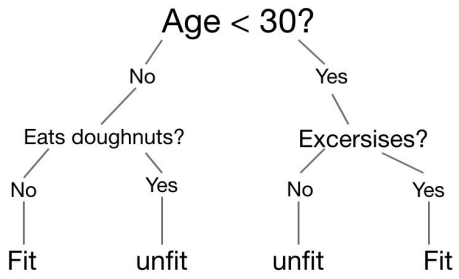

Figure 1. Example Decision Tree

There are different types of decision trees that have differing methods of choosing the order and questions of nodes in the tree. While other types of trees may do it differently, ID3 decision trees selected node order based on the information gained from a particular feature. Simply put information gain represents how well a particular question about a dataset splits up the data labels. For example, In our fitness tree, the first node asks about age being less than thirty. The reason this node would have been chosen first in an ID3 decision tree is because this feature of the data drew a larger distinction between fit and unit than the other questions did. Selecting

nodes based on how much information is gained gives the filtering effect of the decision tree. In order to measure the information gain of a particular question/node we calculate the weighted entropy of the dataset using the Shannon Entropy equation (see figure 1.2). The weighted entropy is calculated by adding the entropy of every value in a feature. To calculate information gained we can simply subtract the weighted entropy of our feature from the preexisting entropy of the labels.

$$H(x) = P(x) * \log(P(x))$$

P(x) - the probability of receiving a certain value (x) in the feature

Log() - Logarithm base 2

Figure 1.2: Entropy calculation

Algorithms

With both data sets being filled with continuous values a choice was made to use binary split points for simplicity. Other options for non-binary split points or for discritizing data would impact the results of the decision tree. With the data being separated on binary split points, each time a node is to be chosen, the information gain (using the formula in figure 1.2) is calculated for every possible split point of every feature, at which point the best is chosen. Possible split points for a given feature are generated by sorting the unique values of the feature and adding the average of every two data points that differ to a list. This process results in a list of every possible split point for the given feature. It is of node that I used the Panas library for all matrix operations which may have an effect on the time and space performance of the training of the algorithm.

Using the ability to find the maximum amount of information gain for any set of features, I recursively built a tree in which each node held the value that was chosen for the split along with the feature that was split on. All left trees were built with training examples below the current split. All right trees were built with training examples above the current split. This is the filtering mentioned earlier where each part of the tree downard deals with smaller and smaller subsets of the original data. When all of the classifications the subset of data are the same, a terminal node was made with the value of the classification. When all examples in the training data held the same values, the mode of the classification labels was taken with smaller integer value labels winning ties.

Cross validation analysis was performed in order to test this algorithm. Both algorithms were ran 100 times for each train/test split shown below (where n denotes the size of the test set)

Iris: n=[1,5,10,25,50,75,100,125,140,145,149]
Cancer: n=[1,5,10,25,50,75,90,100,104]

Data was collected from each run to show the mean and standard error of the percentage of correct classifications of both data sets.

Results

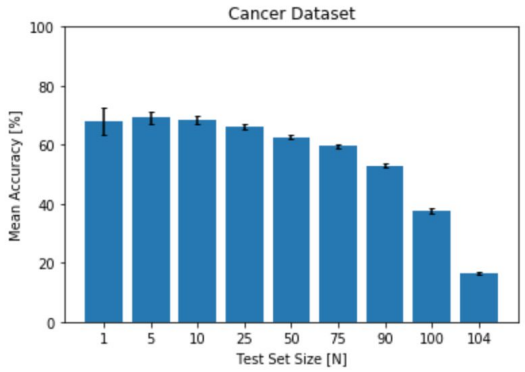

Figure 2: Accuracy of ID3 decision tree on Cancer dataset

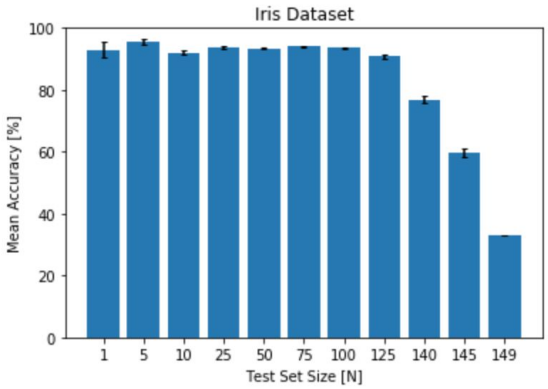

Figure 3: Accuracy of ID3 decision tree on iris dataset

Discussion

The results show the large impact of training/testing split on the accuracy of classification. In this case, the larger the training set, the better the algorithm did on the testing set. This is also shown in the fact that the algorithm performed overall better on the iris data set which had more data points than the cancer data set. It is the case that the bottleneck of supervised learning is data. Supervised learning algorithms are limited by the amount of and quality of data. This is shown very clearly in the results of this decision tree. That being said we would likely see higher accuracy given more training examples. Beyond this there is always the possibility of some hidden feature that would lead to more information gain and thus better classification. Regardless, the results of this lab highlight the impact of data on supervised learning algorithms.