Title: Iterative Dichotomiser 3 with Reduced Error Pruning
Author: Logan Turske
Phone: 989.590.0098

Abstract:

One of the hardest things to do in the field of machine learning is to portray the process of how the algorithms you have implemented work to those people who may not be technical nor comfortable in the field of machine learning. Arguably the easiest algorithm to show others is the decision tree. In this experiment we implement the Iterative Dichotomiser 3 (ID3) decision tree and also perform reduced error pruning on that tree. We tested our ID3 tree on three datasets { abalone.csv[1], car.csv[2], image.csv[3] } for a classification problem. The actual performance of the algorithm varied from dataset to dataset and we explain the reasons below. The algorithm is implemented in Python 2.7 which gives us a very readable implementation. Overall, the results seemed to come out as expected and the algorithm, described below, was very intuitive.

Problem Statement:

We wish to find the average classification accuracy for each of the three datasets { abalone.csv[1], car.csv[2], image.csv[3] } using the Iterative Dichotomiser 3 with and without implementing reduced error pruning.

Hypothesis:

I believe that given a dataset the Iterative Dichotomiser 3 with and without reduced error pruning will have a classification accuracy greater than 50%.

Iterative Dichotomiser 3:

"The basic structure of ID3 is iterative. A subset of the training set is called the *window* is chosen at random and a decision tree is formed from it"[4]. J.R Quinlan produced a paper in 1986 describing the ID3 algorithm along with other algorithms like ID3. As the quote at the beginning of the paragraph stated, ID3 is an iterative process. We begin building the decision tree by iteratively selecting attributes to split the current vertex on and we determine that from the training set and the entropy of those attributes. Entropy lies at the heart of our ID3 algorithm and it can simply be described as the measure of disorder in a system[5]. For each split of the vertex we remove the particular attribute that had the highest entropy at that iteration so that we do not use it further down in the tree. This iterative process will produce a very small tree but not always the smallest tree. The performance of our ID3 tree was also supplemented with the use of reduced error pruning.

Reduced Error Pruning:

When making decision trees we want to generally have smaller trees because larger trees tend to overfit our data. Even though ID3 produces very small trees compared to other algorithms, we still want to attempt to cut down the size of the tree. In this experiment we implemented reduced error pruning to cut down the depth of certain trees in our entire decision tree. The overall process of reduced error pruning can be summed up as "[check] for each internal node, whether replacing it with the most repeated class, that it does not reduce the accuracy of the tree"[6]. In more detail, mark every vertex of your tree to be pruned. One by one prune the vertices and during each iteration attempt to classify your validation set and retrieve the accuracy of the overall tree. If your accuracy did not decrease then you permanently keep that vertex pruned, otherwise you keep that vertex and its subtrees inplace. For this experiment pruning was removing all of the vertices subtrees and taking the majority class that the vertex was classified as. This process improved accuracy overall, however, the pruning took a much longer time due to the fact you had to keep classifying the validation set after each prune.

Experimental Approach:

In this experiment of implementing ID3 and reduced error pruning, we tested the decision tree against three datasets { abalone.csv[1], car.csv[2], image.csv[3] } for classification accuracy. We begin by taking the dataset and using a 5-cross fold validation to partition our dataset appropriately. Then based on the training set we grow the tree to completion using the algorithm ID3 as described above. We then attempt to classify the validation set against the grown ID3 tree and use classification error to measure the accuracy. If selected, we then implement reduced error pruning. Reduced error pruning takes longer due to the fact that we must attempt to classify the tree twice per iteration of the pruning process, once for before the pruning and once after the pruning took place. Below are the results of this experiment.

Results:

Evaluation Metric: Classification error

Data	ID3	ID3 w/Error Prune
abalone.csv	26.15%	26.58%
car.csv	86.93%	89.37%
image.csv	96.51%	96.51%

Table 1.

Behavior of Algorithms:

The results above show that we get a small increase in performance if we use reduced error pruning, however, for these datasets the improvement is very small. One of the surprising outcomes of this experiment was on the abalone.csv dataset. We notice in Table 1 that the accuracy of the experiment was around 26% which seems very low. We believe the reason for such low accuracy is directly tied to the distribution of the classes of the data. There are many classes for this dataset, however, only a handful of them make up the vast majority of the entire dataset. When this distribution of classes occur doing 5-fold cross validation will cause the accuracy to plument. This is not a bad thing because then we can be sure that we are not overfitting our data. It is also our belief that many measurements in this dataset were exactly the same but had different classifiers.

Summary:

After this experiment was conducted further examination of the results and data showed that we created a very generalized ID3 tree with reduced error pruning. Through the use of ID3 we are able to create a very small and readable decision tree that can be explained and walked through by anyone. Reduced error pruning further shrunk the tree in size creating an even more generalized tree. The results of the algorithm, with further analysis of the data, made sense and proved our hypothesis for two out of the three datasets (see Table 1). In conclusion, the ID3 algorithm along with reduced error pruning is a simple way to classify datasets in a easily readable and understandable way.

References:

[1] Warwick. Abalone data set. Retrieved April 1, 2018 from https://archive.ics.uci.edu/ml/datasets/Abalone

[2]Bohanec. Car Evaluation Data set. Retrieved April 1, 2018 from https://archive.ics.uci.edu/ml/datasets/Car Evaluation

[3]Vision Group. Image Segment data set. Retrieved April 1, 2018 from https://archive.ics.uci.edu/ml/datasets/Image Segmentation

[4]J.R. Quinlan. Introduction of Decision Trees.

[5]James P. Sethna. Entropy, Order Parameters and Complexity. , 77–79.

[6]Rinkal Patel. A Reduced Error Pruning Technique for Improving Accuracy of Decision Tree Learning. , 10–11.