Introduction to Machine Learning, Programming Assignment 4

Decision Trees

Julie P. Garcia

Johns Hopkins University

Abstract

Decision Trees are simple to understand and use hierarchical data structures that can be used as classification or regression models for supervised learning of data. In this paper, we evaluate two methods for classification with decision trees. The decision trees in this experiment were built with the Iterative Dichotomizer 3 (ID3) algorithm that was first introduced by J. R Quinlan in 1979.[3] This classic algorithm is then extend to include reduced-error pruning of the tree to improve the accuracy of classification of the tree, while reducing the space complexity.[1]

Hypothesis

In this paper, we demonstrate that the ID3 algorithm for building decision trees works well as a classification algorithm, especially on larger datasets with fewer parameters. As datasets shrink or parameters get larger, the ID3 algorithm the ability to generalize is lost and the algorithm does not perform as well due to overfitting of the data. We also prove that pruning the tree will improve the accuracy slightly, while reducing the space complexity of the algorithm.

Algorithms Implemented

The ID3 algorithm was used to build a complete decision tree from a set of training data. The algorithm attempts to divide the data into a set of partitions established by evaluating the data for the subsets that will give the most information about a particular dataset. The data is divided by creating a tree of decision nodes, that can be traversed to eventually come to a prediction on a leaf of the tree. Reduced-error pruning, described below, was utilized after the completion of the tree construction to eliminate any subtrees that do not provide any significant data or improve the accuracy of the tree.

ID3 Decision Tree Algorithm

ID3 uses Shannon's Theory to define the entropy of the dataset, which tells us the level of uncertainty in the data. Entropy is calculated on the training dataset as a whole and then Expected Entropy is calculated for each possible subset of the data by feature. Entropy is calculated as follows for k > 2 classification problems, where C = ($c_1, c_2, .. c_k$) is the set of class labels of the dataset:

$$I(\mathrm{c1}, \mathrm{c2}, .. \mathrm{ck}) = \sum_{l=1}^{k} - \frac{c_l}{\mathrm{c1} + \ \mathrm{c2} + .. \mathrm{ck}} * \log \frac{c_l}{\mathrm{c1} + \ \mathrm{c2} + .. \mathrm{ck}}$$

Expected Entropy is calculated for each possible partition of the data, where $c_{\pi,1}^{j}$ is equal to the number of examples in the jth partition when applying the feature f_i:

$$E(f_i) = \sum_{j=1}^{m_i} \frac{c_{\pi,1}^{j} + \cdots + c_{\pi,k}^{j}}{c_{\pi,1} + \cdots + c_{\pi,k}} * I(c_{\pi,1}^{j}, \ldots, c_{\pi,k}^{j})$$

Information Gain is then calculated for each feature, and the feature that supplies the most information gain is chosen for the current node of the tree. Information gain is calculated as:

$$gain(f_i) = I(\mathrm{c1}, \mathrm{c2}, .. \mathrm{ck}) - E(f_i)$$

A gain ratio is used to offset overfitting when a feature has many values.

The algorithm was implemented as follows:

1. Build the decision tree on the current training set
    a. Calculate the entropy for the dataset
    b. Loop through each feature
        i. Calculate expected entropy for the partition of the data for that feature
        ii. Calculate gain ratio the feature
        iii. Choose the feature with the largest gain ratio
2. Make predictions on the current test set using the decision tree
3. Test the class predictions for accuracy against the actual classes
4. If pruning option is selected, run reduced-error pruning and compare the accuracy results with the complete tree results

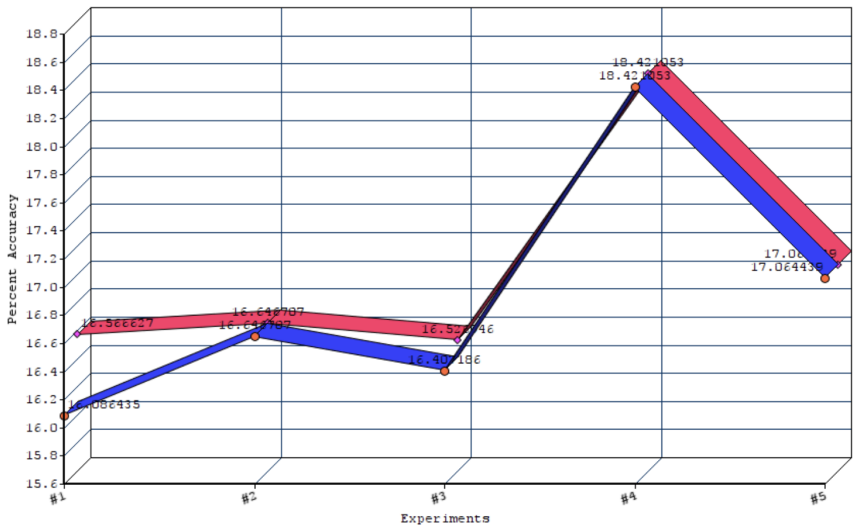
Experimental Approach

All algorithms were run on three classification datasets: Abalone, Image Segmentation, and Car Evaluation. For this a stratified 5-fold cross-validation method was used. Each dataset was divided into five randomly-selected equal datasets. The datasets are stratified to ensure equal representation of classes across each set. Five experiments were run, each time selecting a different partition (20%) as the test set and the remaining 80% as the training set. This allows for each of the 5 sets to rotate as the test set. For classification problems, averages were calculated over the five experiments.

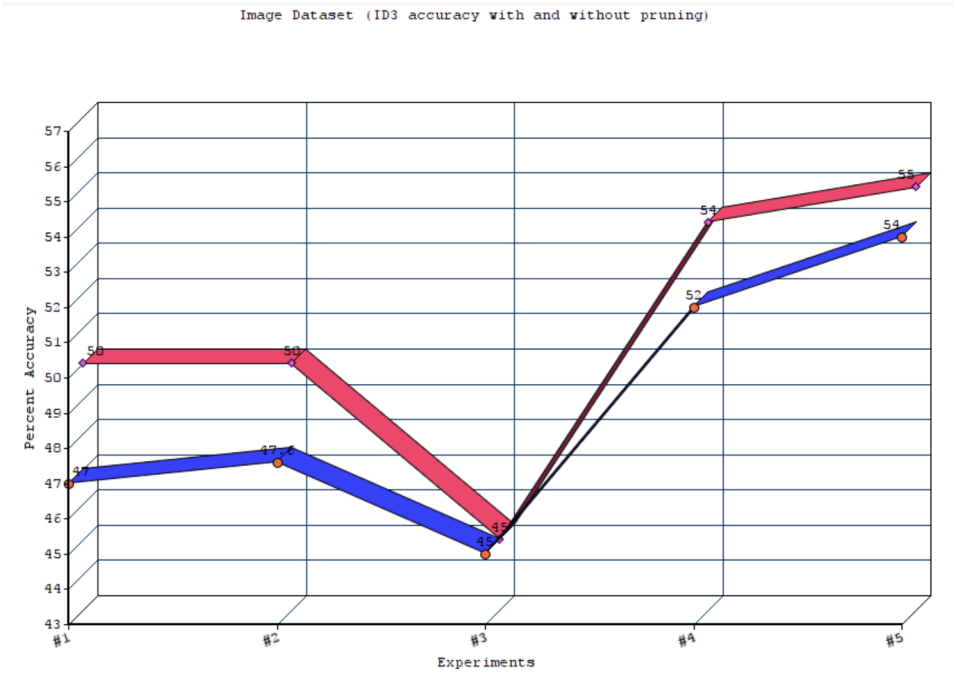
Reduced-error pruning was used in an attempt to avoid overfitting the data. When there are a large number of parameters in a decision tree, the ID3 algorithm may begin to learn noise overtime. This is a result of memorizing of data and building a tree to fit the data exactly. This reduces the models ability to generalize.

Results

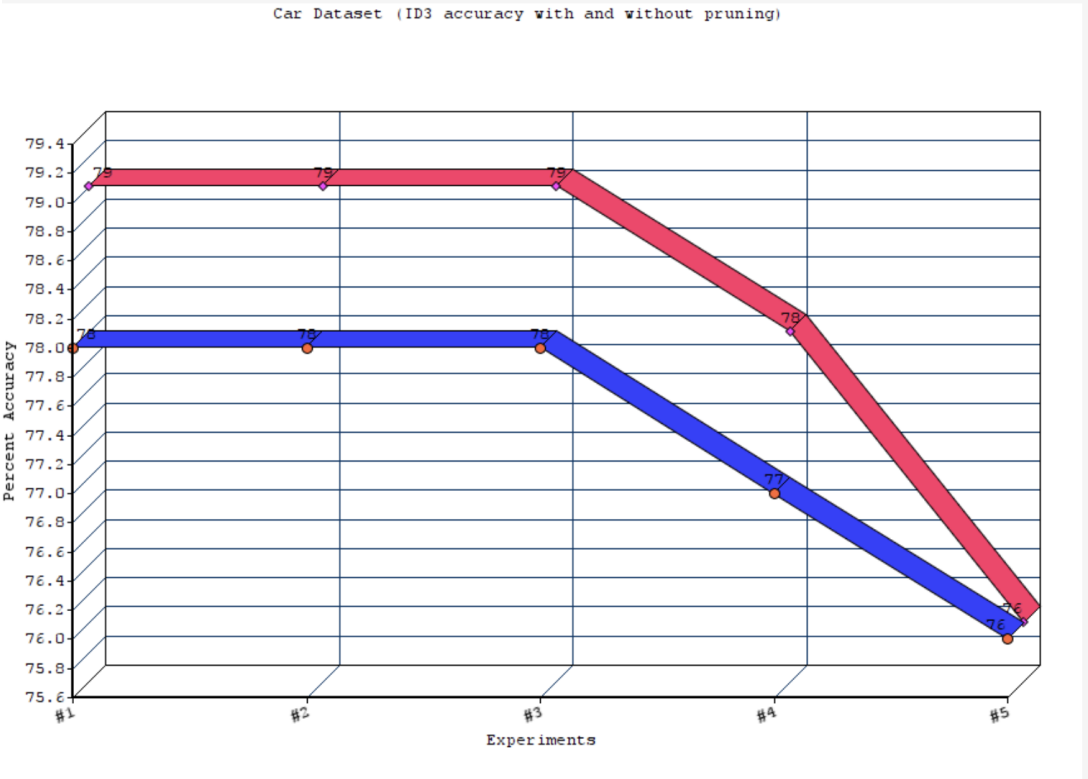
When run on the Abalone dataset (4200 examples, 8 features) the algorithm performed very poorly with an average over the five experiments at ~16%. Pruning only slightly improved this accuracy at ~17%. The figure below shows the percentage accuracy plotted for each of the five experiments with the red line with pruning and blue line without.



The Image Segmentation dataset (210 examples, 19 features) performed better with an average of ~50% accuracy without pruning, and ~50.5% with pruning. For this dataset, pruning did not seem to have much of an effect as shown in the figure below. This may be because of the small size of the dataset at 210 examples. It would be interesting to see how increasing the size of this dataset would affect the outcome.



The Car Evaluation dataset (1728 examples, 6 features) performed the best with an average score of ~77%, with pruning improving that to ~79%.



Summary

The ID3 algorithm performed best when there were a smaller number of features in the dataset, and the feature values were categorical values such as in the car evaluation dataset. When the number of features increased and the number of values for each feature increased the generalization of the model decreased. This is likely due to overfitting of the data. Reduced-error pruning always increased the accuracy of the predictions, however, only by a small percentage in all cases. In general, the ID3 decision tree algorithm will perform well for a dataset with a small number a features with clearly defined categorical values.

References

1. Alpaydin, E. (2014). *Introduction to machine learning*. Cambridge, MA: The MIT Press.
2. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
3. Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014). A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications,4*(2). doi:10.14569/specialissue.2014.040203