Exploring the Decision Forest

Patrick M. Murphy & Michael J. Pazzani

Department of Information & Computer Science
University of California, Irvine, CA 92717
pmurphy@ics.uci.edu
pazzani@ics.uci.edu

Abstract

We report on a series of experiments in which all decision trees consistent with the training data are constructed. These experiments were run to gain an understanding of the properties of the set of consistent decision trees, and the factors that affect the error rate of individual trees. The experiments were performed on a massively parallel Maspar[1] computer. The results of the experimentation on two artificial and two real world problems indicate that for three of the four problems investigated, the smallest consistent decision trees tend to be less accurate than the average accuracy of those slightly larger.

Presented at:

Computational Learning and Natural Learning Workshop, Provincetown Massachusetts, 10-12 Spetember 1993.

[1]The research reported here was supported in part by NSF infrastructure grant number MIP-9205737

1 Introduction

The top-down induction of decision trees is an approach to machine learning that has been used on a variety of real world tasks. Decision trees are well-suited for such tasks since they scale fairly well with the number of training examples and the number of features and can represent complex concepts in a representation that is fairly easily for people to understand.

Decision tree induction algorithms (e.g., Breiman, Friedman, Olshen & Stone, 1984; Quinlan, 1986; Fayyad, & Irani,1992) typically operate by choosing a feature that partitions the training data according to some evaluation function (e.g., the purity) of the resulting partitions. Partitions are then further partitioned recursively until some stopping criterion is reached (e.g., the partitions contain training examples of a single class). Nearly all decision tree induction algorithms create a single decision tree based upon local information of how well a feature partitions the training data. However, this decision tree is only one of a set of decision trees consistent with the training data. In this paper, we experimentally examine the properties of the set of consistent decision trees. We will call the set of decision trees that are consistent with training data a *decision forest.*

Our experimentation will be done on two artificial concepts for which we know the correct answer, and two naturally occurring databases from real world tasks available from the UCI Machine Learning Repository (Murphy & Aha) in which the correct answer is not known. The goal of this experimentation is to gain insight into the quality of the decision trees produced by an existing algorithm, ID3 (Quinlan, 1986) and to understand how factors such as the size of a consistent decision tree are related to the error rate on classifying unseen test instances.

For the purpose of this paper a consistent decision tree is one that correctly classifies every training example[2]. We also place two additional constraints on decision trees. First, no discriminator can pass all instances down a single branch. This insures that the test made by the decision tree partitions the training data. Second, if all of the training instances at a node are of the same class, no additional discriminations are made. In this case a leaf is formed with class label specified by the class of the instances at the leaf. These two constraints are added to insure that the decision trees analyzed in the experiments correspond to those that could be formed by top down induction of decision tree algorithms. In this paper, we will not investigate domains that have continuously-valued features or missing feature values.

In Section 2, we will report on some initial experimentation on four data sets. In these

[2] The artificial and natural domains we study here have consistent training sets.

experiments, the smallest consistent decision trees tend to be less accurate than the average accuracy of those slightly larger. Section 3 provides results of additional experiments that address this issue. Section 4 shows where the decision trees that ID3 builds fit in the population of consistent decision trees.

2 Initial Experiments

We will investigate the relationship between various tree characteristics and error. In particular, we will look at node cardinality (i.e., the number of internal nodes in a tree) and leaf cardinality (i.e., the total number of leaves in a tree).

It should be noted that even when using a powerful massively parallel computer, the choice of domains is severely constrained by the computational complexity of the task. The number of trees of any node cardinality that might be generated is $O(number\_of\_discriminators^{node\_cardinality})$. This precluded the use of domains with many features or any continuously-valued features.

2.1 $XYZ \vee AB$

In the first experiment, we consider learning from training data in which there are 5 boolean features. The concept to be learned is $XYZ \vee AB$. This concept was chosen because it was of moderate complexity, requiring a decision tree with at least 8 nodes to represent correctly. We ran 100 trials of creating a training set by randomly choosing 20 of the 32 possible training examples and using the remaining 12 examples as the test set. For each trial, every consistent decision tree was created and we computed the average error rate made by trees with the same node cardinality. Figure 1 plots the mean and 95% confidence interval of these average errors as a function of the node cardinality. Figure 1 also plots the number of trials on which at least one decision tree of a given node cardinality is consistent with the training data.

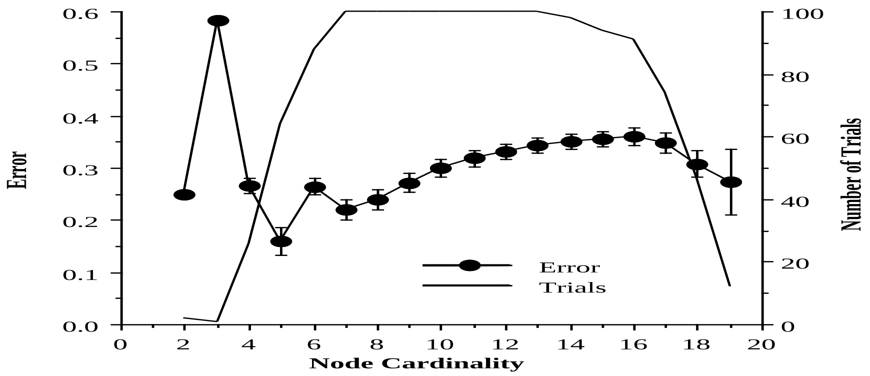

Figure 1. Average Error of 100 train/test trials as a function of Node Cardinality, Number of Trials for each Node Cardinality

From node cardinality 7 to node cardinality 16, there is a monotonic increase in error with increasing node cardinality. For the range from 2 to 3 nodes, the error is varied, but there is little evidence for these error values because they are based on only 2 and 1 trials, respectively. For the range of node cardinalities between 4 and 7, average error is definitely not a monotonically increasing function of node cardinality. From the curve, 5 node trees are on the average more accurate than 4 node trees, and 7 node trees are on the average more accurate than trees with 6 nodes. This last result is somewhat surprising since one gets the impression from reading the machine learning literature (e.g., Muggleton, Srivivasan & Bain, 1992) that the smaller hypothesis (i.e., the one that provides the most compression of the data (Rissanen, 1978)) is likely to be more accurate. We will explore this issue, in further detail in Section 3. First, we'll present data showing that this result is not unique to this particular concept.

Table 1 lists the average number of consistent trees for each node cardinality, and the average number of correct trees (i.e., those trees consistent with the training data that make no errors on the unseen test examples). There are no correct trees with fewer than 8 nodes, since at least 8 nodes are required to represent this concept.

Nodes	*Number of Consistent Trees*	*Number of Correct Trees*
2	2.0	0.0
3	4.0	0.0
4	3.3	0.0
5	12.3	0.0
6	27.6	0.0
7	117.1	0.0
8	377.0	17.8
9	879.4	37.8
10	1799.9	50.2
11	3097.8	41.6
12	4383.0	95.4
13	5068.9	66.6
14	4828.3	37.7
15	3631.5	31.3
16	1910.6	14.8
17	854.4	4.0
18	308.6	3.6
19	113.8	0.0

Table 1. Average Number of Trees

2.2 Mux6

The next problem we consider, MUX6, has a total of 8 binary features. Six features represent the functionality of a multiplexor, while unlike the previous experiment, 2 features are irrelevant. On each trial, we selected 20 examples randomly and tested on the remaining examples. Since most of the computational cost of building consistent trees is for larger node cardinalities, and the initial experiment showed that comparisons among the smaller node cardinalities was most interesting, we only computed consistent trees with up to 10 nodes for 10 trials and up to size 8 for 340 trials. Figure 2 presents the average error as a function of the node cardinality for these trials.

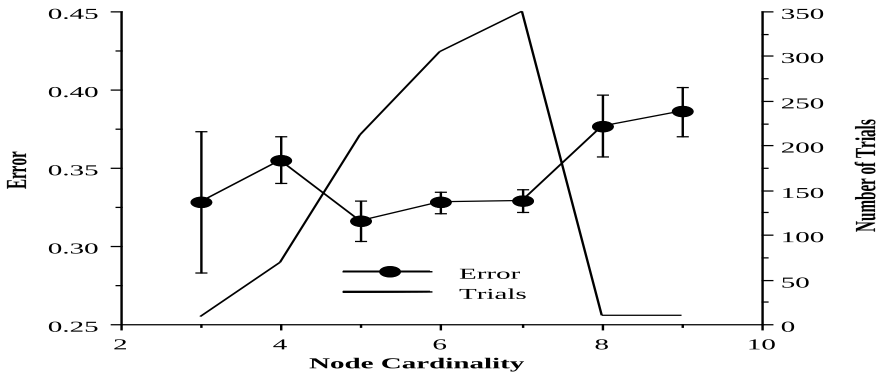

Figure 2. Average Error on MUX6.

This graph again shows that average error does not monotonically increase with node cardinality. Trees of 4 nodes are on the average 4% less accurate than trees of 5 nodes.

2.3 Lenses

The lenses domain has one 3-valued and three binary features, three classes and 24 instances. Since the lenses domain has one non-binary feature, trees with a range of leaf cardinalities are possible for a particular node cardinality. Therefore, we will perform separate analyses for leaf and node cardinalities. We used training set sizes of 8, 12 and 18 for this problem, built all consistent trees and measured the error rate on all unseen examples.

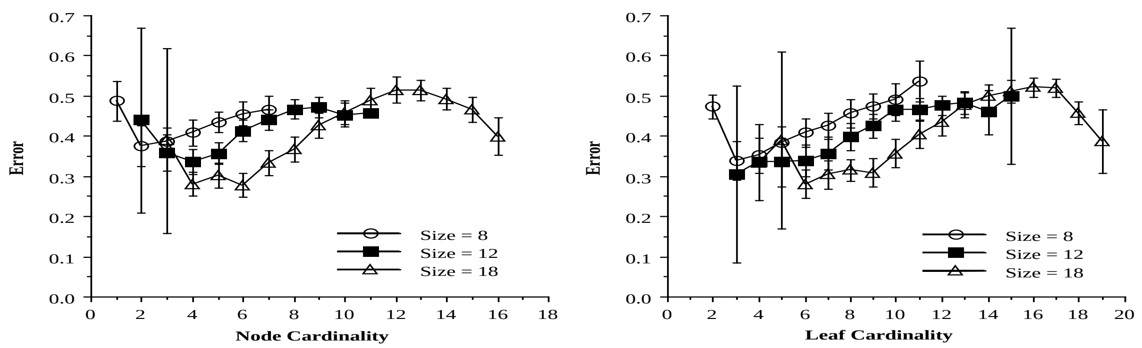

Figure 3. (a) Error as a function of Node Cardinality (b) Error as a function of Leaf Cardinality

Figure 3a shows the error as a function of the node cardinality for the 3 training set sizes averaged over 50 trials. These curves indicate that the smallest consistent trees are not always the most accurate. When observing the larger node cardinalities for the training set sizes 12 and 18, error monotonically decreases with increasing node cardinality. Similar statements can be said for the curve in Figure 3b which relates average error as a function of leaf cardinality.

2.4 Shuttle Landing

The shuttle landing domain has four binary and two 4-valued features, two classes and 277 instances. We used training sets of size 20, 50 and 100 for the shuttle domain, generating all consistent decision trees with fewer than 8, 10 and 12 nodes, and measured the error of these trees all unseen examples. Figure 4 presents the error as a function of leaf cardinality, averaged over 10 trials. For this domain, there is a monotonically increasing relationship between node cardinality and error.

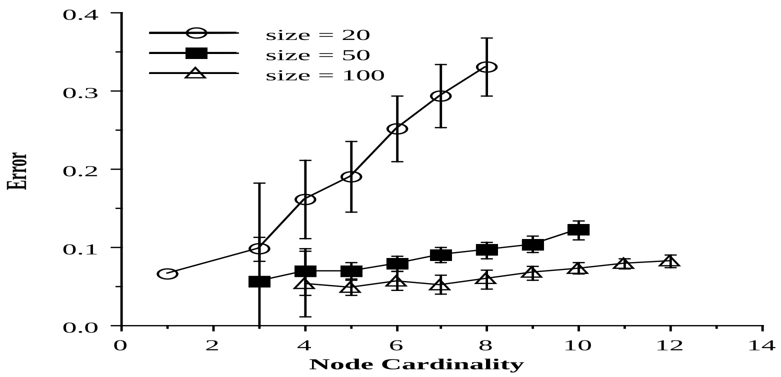

Figure 4. Error as a function of Node Cardinality for the Shuttle domain

2.5 Summary

On three of the four domains studied, we found that on average, the smallest decision trees consistent with the training data had more error on unseen examples than those trees slightly larger. In the next section, we will run additional experiments to make sure that this result is not an artifact of the experimental methodology that we have used.

3 Further Experimentation

One possible explanation for the finding of the previous section is that the smaller decision trees are formed from nonrepresentative samples. For example, there are 11 positive and 21 negative examples of the concept $XYZ \vee AB$. If all or most of the examples in the training set are negative, then a very small tree may be learned which would probably do very poorly on the mostly positive test set. To further insure that the results are not caused by unrepresentative training sets, we eliminated all training data that was not reasonably representative.[3] In particular, since there is a $\frac{11}{32}$ probability that a training instance is positive, a representative training set of size 20 would have about 7 positive instances. Since one standard deviation would be $\sqrt{20 * \frac{11}{32} * (1 - \frac{11}{32})}$, we eliminated from analysis those training sets with greater than 8 or fewer than 5 positive instances. Similarly, there is a 0.5 probability that each binary feature takes on a true value, so we eliminated from analysis any training data which has any feature that is true in greater than 13 or fewer than 7 instances. Figure 6 is based on the 69 of 100 trials of the $XYZ \vee AB$ concept that meet this representative test. Notice that the two trials that formed the only 2 and 3 node trees were removed. Even when only the more representative training sets are considered, the average error of trees of size 4 is greater than the average error of size 5 trees.

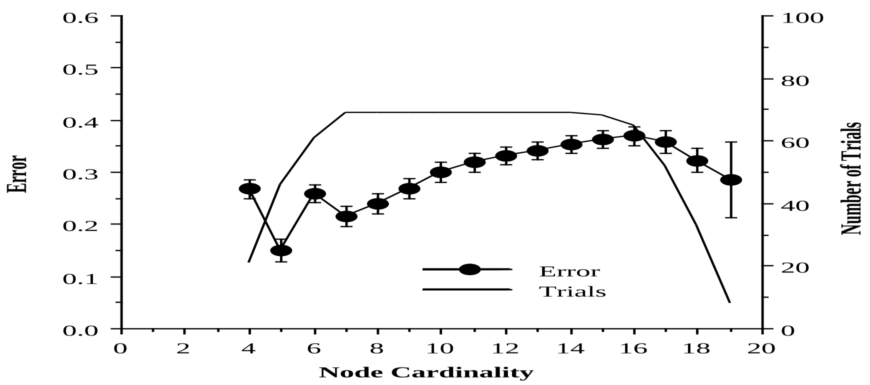

Figure 5. Error Rate of consistent trees from representative training sets as a function of Node Cardinality

The minimum sized decision tree for the concept $XYZ \vee AB$ has 8 tests and 9 leaves. Since the correct trees does not provide much compression[4] of a set of 20 examples used to induce the tree, one might argue that the sample used was too small for this complex a concept.[5] Therefore, we increased the number of training examples to the maximum possible. Figure 6 plots the average error of 32 trials in which we formed all decision trees consistent with 31 examples, and evaluating on the single unseen test example. Figure 6 shows that the smaller trees formed from samples of size 31 have more error than the slightly larger trees. Since the minimum correct decision tree has 8 nodes, and the consistent trees classify all 31

[3] We thank Ross Quinlan for suggesting the method for finding representative samples.

[4] The exact amount of compression provided depends upon the particular scheme chosen for encoding the training data. See Quinlan & Rivest (1989) and Wallace & Patrick (1993) for two such schemes.

[5] We thank Geoffrey Hinton for suggesting this possibility.

training examples correctly, any decision tree with less than 8 nodes classifies the test example incorrectly.

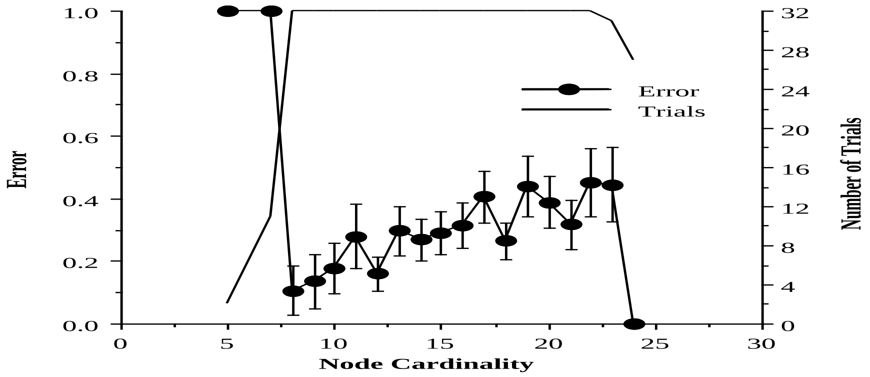

Figure 6. Error Rate of consistent trees with leave-one-out testing as a function of Node Cardinality

Schaffer (1992, 1993) presents a series of experiments on overfitting avoidance algorithms. Overfitting avoidance algorithms prefer simpler decision trees over more complex ones, even though the simpler decision trees are less accurate on the training data in hopes that the trees will be more accurate on test data. Schaffer shows that these overfitting avoidance algorithms are a form of bias. Rather than uniformly improving performance, the overfitting avoidance algorithms improve performance on some distribution of problems and hinder performance on other problems.

The results of our experimentation go a step further than Schaffer's. We have shown that for some concepts the preference of simpler decision trees does not result in an increase in predictive accuracy on unseen test data, even when the simple trees are consistent with the training data. Like Schaffer, we do not dispute the theoretical papers on Occam's razor (Blumer at al., 1987), minimum description length (Quinlan & Rivest, 1987; Muggleton et al, 1992 1987), or minimizing the number of leaves of a decision tree (Fayyad & Irani, 1990). Rather, we point out that for a variety of reasons, the assumptions behind these theoretical papers mean that the results of these papers do not apply to the experiments reported here. For example, Blumer et al. (1987) indicates that if ones finds a hypothesis in a sufficiently small hypothesis space (and simpler hypotheses are one example of a small hypothesis space) and this hypothesis is consistent with a sufficiently large sample of training data, then one can be fairly confident that it will be fairly accurate on unseen data drawn from the same distribution of examples. However, it does not say that on average this hypothesis will be more accurate than other consistent hypotheses not in this small hypothesis space.

The Fayyad & Irani (1990) paper explicitly states that the results on minimizing the number of leaves of decision trees are worst case results and should not be used to make absolute statements concerning improvements in performances. Nonetheless, in informal arguments in the paper, the authors state that "... one method for inducing decision trees is better than another by proving that one algorithm always produces a tree with a smaller number of leaves, given the same training data." Furthermore, in other informal arguments they imply that result is probabilistic because of the existence of "pathological training sets." However, as we have shown in Figure 5 (as well as reanalysis of the MUX6 data) eliminating pathological (i.e., nonrepresentative) training sets does not change the qualitative result that on three of the four domains the smaller trees are less accurate predictors than those slightly larger.

Fayyad & Irani assumes that the number of errors of a decision tree with a given number of leaves is uniformly distributed from 0 to a worst case upper bound. For each leaf cardinality, we computed the observed proportion of consistent trees for each possible number of errors (from 0 to 12) on the $XYZ \vee AB$ concept with 20 training examples. As Figure 7 shows, the errors for the $XYZ \vee AB$ concept do not appear to be distributed uniformly. The shuttle concept which had average errors monotonically increase with leaf cardinality also does not appear to have a uniform distribution of errors.

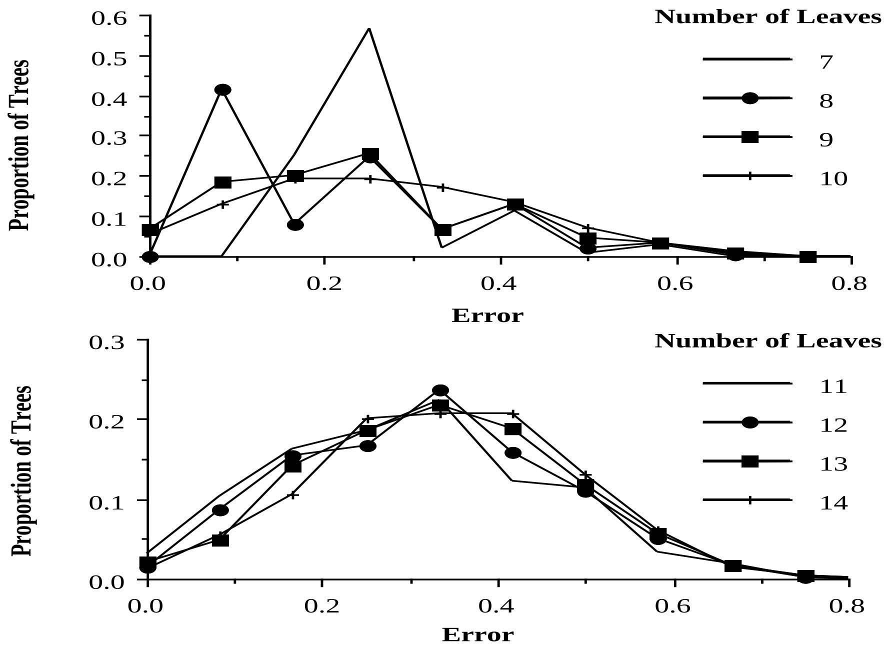

Figure 7. Proportion of trees with a given error rate for trees
with 7 to 10 leaves (upper) and 11 to 14 leaves (lower) for XYZ $\vee$ AB

4 ID3

ID3 (Quinlan, 1986) finds a single decision tree consistent with the training data. Here, we address the issue of how the tree found by ID3 compares to the minimum size tree, and to other consistent trees with the same number of nodes as ID3's tree. Table 2 provides the relevant data.

Domain	*ID3 Node Cardinality*	*ID3 Error (%)*	*Min. Node Cardinality*	*Min. Node Error (%)*	*Average Error (%)*
$XYZ \vee AB$	6.11	20.2	5.18	19.8	21.2
mux6	7.12	36.6	5.30	30.0	34.5[6]
lenses, size = 8	2.53	36.8	2.44	34.9	35.8
lenses, size = 12	3.90	32.2	3.66	30.6	31.9
lenses, size = 18	4.62	19.1	4.60	18.9	19.0
shuttle, size = 20	2.25	9.4	2.20	9.6	9.7
shuttle, size = 50	4.34	7.4	4.20	6.8	7.1
shuttle, size = 100	5.74	5.4	5.10	5.1	5.0

The values in this table are averages across all trials for a particular domain and training set size. For each trial, ID3 was run 50 times[7] with node cardinality and error recorded for each tree built. The average of these measurements over the 50 trials formed average node cardinalities and average error rates for ID3. Columns 2 and 3, respectively, represent the average of these averages over all trials for a particular domain and training set size. Column 4 is the average size of the smallest consistent decision tree for a training set. Column 5 is the average error of the set of smallest decision trees. Column 6 averages an interpolation of the average error of the set of decision trees slightly larger and slightly smaller than the average node cardinality of the trees ID3 built. In short, the columns from left to right represent the average node cardinality and error of ID3's trees, the average node cardinality and error rate of the smallest possible trees and the average error rate of all trees the size of ID3's trees.

Except for the mux6 domain, it appears that ID3 seems to do fairly well at generating near minimal node cardinality decision trees. Similarly, ID3's error rate is competitive with the average error rates of the smallest trees and the average error rates of all trees with similar node cardinality to ID3's trees.

[6] Only those average errors corresponding to ID3 trees of size less than or equal to the maximum node limit were included in this average.

[7] When faced with multiple equally evaluated discriminators, ID3 chooses among them randomly. Because of this nondeterminism, it seemed reasonable to build and evaluate many decision trees for each training set.

The results for the mux6 domain were somewhat different. ID3 did not build near minimal trees, and its error rate was greater than the average error rate of the minimal cardinality trees and the average error rate of all ID3-sized trees. Quinlan (1993) notes that the information-gain heuristic of ID3 rarely chooses an "address line" of the multiplexor as the root node of a decision tree and therefore rarely finds a near minimal size.

5 Conclusion

We have reported on a series of experiments in which we generated all decision trees on two artificial problems and two naturally occurring data sets. A somewhat unexpected result is that on three of the problems, those consistent decision trees that had the fewest number of nodes were less accurate on unseen data than those slightly more accurate. Although unexpected, the results do not contradict existing theoretical results. Rather, they serve to remind us to be cautious when informally using the intuitions derived from theoretical results on problems that are not covered by the theorems, or when using intuitions derived from worst-case results to predict average case performance.

We have also evaluated ID3, a commonly used decision tree induction algorithm that has been used on a wide variety of real world task. We have shown that for the most part, ID3's trees are only slightly larger than the minimum sized tree. In addition, the tree found by ID3 have error rates similar to the average error rate of all consistent trees with the same node cardinality.

Acknowledgments

We'd like to thank Ross Quinlan, Geoffrey Hinton, Michael Cameron-Jones, Cullen Schaffer, Dennis Kibler and Steve Hampson for commenting on various aspects of this research.

References

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1987). Occam's Razor. *Information Processing Letters 24*, 377–380. North-Holland.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Pacific Grove,CA: Wadsworth & Brooks.

Fayyad, U. M & Irani, K. B. (1990). What Should be Minimized in a Decision Tree? In *Proceedings of the Eighth National Conference on Artificial Intelligence AAAI-90*, 749–754. Cambridge, MA: MIT Press.

Fayyad, U. M & Irani, K. B. (1992). The Attribute Selection Problem in Decision Tree Generation. In *Proceedings of the Tenth National Conference on Artificial Intelligence AAAI-92*, 104–110. Cambridge, MA: MIT Press.

Muggleton S., Srinivasan A. and Bain M. (1992). Compression, Significance and Accuracy. In *Machine Learning: Proceedings of the Ninth International Workshop.* Aberdeen, Scotland. Morgan Kaufmann.

Murphy, P. M., & Aha, D. W. *UCI Repository of machine learning databases* [Machine-readable data repository]. Irvine, CA: University of California, Department of Information and Computer Science.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 81–106. Kluwer.

Quinlan J. R. and Rivest R. L. (1989). Inferring Decision Trees Using the Minimum Description Length Principle. *Information and Computation, 80, pp. 227–248.*

Quinlan, J. R. (1993). *C4.5 Programs for Machine Learning.* San Mateo,CA: Morgan Kaufmann.

Rissanen J. (1978). Modeling by Shortest Data Description. *Automatica, 14.*

Schaffer, C. (1992). Sparse Data and the Effect of Overfitting Avoidance in Decision Tree Induction. In *Proceedings of the Tenth National Conference on Artificial Intelligence AAAI-92*, 147–152. Cambridge, MA: MIT Press.

Schaffer, C. (1993). Overfitting Avoidance as Bias. *Machine Learning*, 153–178. Kluwer.

Wallace, C. & Patrick. (1993). Coding Decision Trees. *Machine Learning*, 7–22. Kluwer.