Decision Tree Induction: How Effective is the Greedy Heuristic?

Sreerama Murthy and **Steven Salzberg**

Department of Computer Science
Johns Hopkins University
Baltimore, Maryland 21218
lastname@cs.jhu.edu

Abstract

Most existing decision tree systems use a greedy approach to induce trees — locally optimal splits are induced at every node of the tree. Although the greedy approach is suboptimal, it is believed to produce reasonably good trees. In the current work, we attempt to verify this belief. We quantify the goodness of greedy tree induction empirically, using the popular decision tree algorithms, C4.5 and CART. We induce decision trees on thousands of synthetic data sets and compare them to the corresponding optimal trees, which in turn are found using a novel map coloring idea. We measure the effect on greedy induction of variables such as the underlying concept complexity, training set size, noise and dimensionality. Our experiments show, among other things, that the expected classification cost of a greedily induced tree is consistently *very* close to that of the optimal tree.

Introduction

Decision trees are known to be effective classifiers in a variety of domains. Most of the methods developed have used a standard top-down, greedy approach to building trees, which can be summarized as follows. Recursively do the following until no more nodes can be split: choose the best possible test at the current node according to some *goodness measure* and split the current node using that test; after a complete tree is grown, prune it back to avoid overfitting the training data (Breiman *et al.* 1984; Quinlan 1993). The choice of a "best" test is what makes this algorithm greedy. The best test at a given internal node of the tree is only a locally optimal choice; and a strategy choosing locally optimal splits necessarily produces suboptimal trees (Goodman & Smyth 1988).

Optimality of a decision tree may be measured in terms of prediction accuracy, size or depth. It should be clear that it is desirable to build optimal trees in terms of one or more of these criteria. Maximizing classification accuracy on unseen data (within the constraints imposed by the training data) is obviously desirable. Smaller, shallower decision trees imply better comprehensibility and computational efficiency. Shallow trees are also more cost-effective, as the depth of a tree is a measure of its classification cost. However, because the problem of building optimal trees is known to be intractable (Hyafil & Rivest 1976; Murphy & McCraw 1991), a greedy heuristic might be wise given realistic computational constraints.

The goal of this paper is to examine closely the consequences of adopting a greedy strategy. We ask the question, if we had unlimited resources and could compute the optimal tree, how much better should we expect to perform? An alternative way of asking the same question is, what is the penalty that decision tree algorithms pay in return for the speed gained by the greedy heuristic?

Setting up the Experiments

Our experimental framework is quite simple — we use C4.5 (Quinlan 1993) and CART (Breiman *et al.* 1984) to induce decision trees on a large number of random data sets, and in each case we compare the greedily induced tree to the optimal tree. The implementation of this framework raises some interesting issues.

Optimal Decision Tree for a Training Set. The problem of computing the shallowest or smallest decision tree for a given data set is NP-complete (Hyafil & Rivest 1976; Murphy & McCraw 1991), meaning that it is highly unlikely that a polynomial solution will be found. Previous studies that attempted comparisons to optimal trees (e.g., (Cox, Qiu, & Kuehner 1989)) used approaches like dynamic programming to generate the optimal trees. Because it is slow, this option is impractical for our study, in which we use hundreds of thousands of artificial data sets. Our solution is to first generate a random decision tree D, and *then* generate data sets for which D is *guaranteed* to be the optimal tree. The main idea behind ensuring the optimality of a random decision tree is coloring its leaf nodes with appropriate class labels.

An *instance* is a real valued vector $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ plus a class label c_i. x_is are the *attributes* of X_i, and d is its dimensionality. Consider a binary decision tree D in two attributes. (The ensuing argument applies to arbitrary dimensions.) D induces

a hierarchical partitioning of the attribute space, which can be drawn as a map M. The boundaries of M are the splits (test nodes) in D, and the regions of M are the leaf nodes in D. Assuming that each leaf node of D contains instances of only one class, we can color M by assigning a distinct color to each class in D. Now consider a data set S consistent with D, which has the additional property that S requires every leaf node of D, i.e., every leaf node of D contains at least one instance of S.

It should be clear that D is the smallest binary decision tree consistent with S, *provided* no two neighboring regions of M have the same color. Informally, any decision tree that has fewer leaves than D needs to either ignore some decision regions of D, or merge (parts of) two or more regions into one. The former possibility is ruled out because S requires all decision regions in D. The latter is impossible because no decision regions of the same color are adjacent, so no two regions can be merged. Hence, *any* decision tree consistent with S has to have at least as many leaf nodes as D. Moreover, if D was a perfectly balanced tree to start with, then any decision tree consistent with S has to be at least as deep as D.

In our experiments, we start with perfectly balanced, empty trees. We then generate random tests at the decision nodes, ensuring that no leaf region is empty. Finally we color the leaves to ensure optimality with respect to size, using the following procedure. We first compute the adjacency information of the leaf nodes. After initializing the class labels at all leaf nodes to k ($\geq$ number of leaves), we go back and change the label of each leaf to be the smallest number in $[1, k]$ that is not yet assigned to any neighbor. This heuristic procedure worked quite well in all our experiments. (For instance, decision trees of 64 leaf nodes in the plane were colored with 5 classes on average.) A sample random decision tree in 2-D, along with the class labels assigned by the above coloring procedure, is shown in Fig. 1.

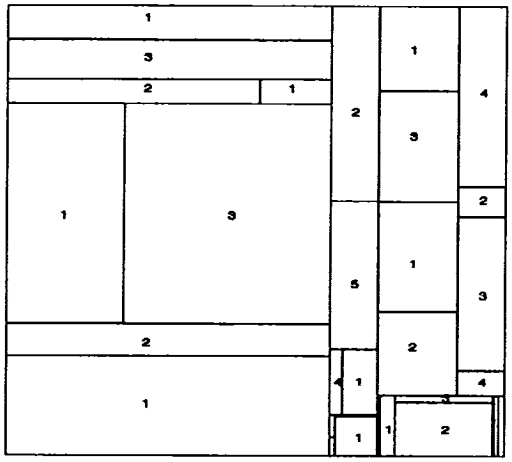

Figure 1: The partitioning induced by a random decision tree of 32 leaf nodes. Class labels assigned by our coloring procedure are shown (for most nodes).

Tree Quality Measures. In all our experiments, we report tree quality using six measures:

- Classification accuracy: accuracy on the training data;
- Prediction accuracy: accuracy on an independent, noise-free testing set;
- Tree size: number of leaf nodes;
- Maximum depth: distance from the root to the farthest leaf node; (distance from A to B is the number of nodes between, and including, A and B)
- Average depth: mean distance from the root to a leaf node in the tree;
- Expected depth: number of tests needed to classify an unseen example. We compute expected depth by averaging, over all the examples in the testing set, the length of the path that the example followed in the tree.

Control Variables. The effectiveness of greedy induction can not be measured independently of training data characteristics. For instance, if the training data is very noisy, it is likely that no induction method will be able to generate accurate trees. In this paper, we study the effectiveness of greedy induction in controlled settings with respect to the following parameters:

- concept complexity (measured as the size of the optimal decision tree),
- size of the training set,
- amount and nature of noise in the training data (noise in class labels versus noise in attributes), and
- dimensionality (number of attributes).

Tree Induction Methods Used. The tree induction methods we use are C4.5 (Quinlan 1993) and CART (Breiman *et al.* 1984). One main difference between C4.5 and CART is the *goodness criterion*, the criterion used to choose the best split at each node. C4.5 uses the information gain [1] criterion, whereas CART uses either the Gini index of diversity or the twoing rule. All the experiments in this paper were repeated using information gain, Gini index and twoing rule. *In no case* did the results show statistically significant differences between goodness measures—the differences in accuracies, sizes *and* measurements of depth were always much less than one standard deviation. For brevity, we report only the results with information gain (i.e., C4.5) in the rest of this paper. We implemented all the goodness measures using the OC1 system (Murthy,

[1] Quinlan suggested gain ratio as an improvement over information gain. However the two measures are equivalent in our experiments as all our decision trees are binary.

Optimal Size	Training Set	Classification Accuracy	Prediction Accuracy	Tree Size	Depth		
					Maximum	Average	Expected
8	1000	100.0	99.5±0.1	9.8±1.7	4.8±0.7 (3)	3.6±0.3 (3)	2.9±0.3 (3)
16	1000	100.0	98.7±0.3	20.7±3.3	7.2±1.0 (4)	5.0±0.4 (4)	3.9±0.4 (4)
32	1000	100.0	97.2±0.6	40.4±6.8	9.3±1.0 (5)	6.3±0.5 (5)	5.0±0.5 (5)
64	1000	100.0	94.3±0.9	71.7±10.3	11.5±1.2 (6)	7.4±0.5 (6)	5.8±0.5 (6)

Table 1: Effects of concept complexity. No noise in data. Numbers in parentheses are for the optimal trees.

Kasif, & Salzberg 1994). Although C4.5 and CART differ in respects other than the goodness measures, we have not implemented these differences. In the experiments in which the training data is noise-free, no pruning was used with either method. In the experiments using noisy training sets, we augment both methods with cost complexity pruning (Breiman *et al.* 1984), reserving 10% of the training data for pruning.

Experiments

This section describes five experiments, each of which is intended to measure the effectiveness of greedy induction as a function of one or more control variables described earlier. The procedure is more or less the same for all experiments.

For each setting of the control variables:
  generate 100 random trees with no class labels;
  for each tree D_{opt} generated in the above step:
    color D_{opt} with class labels;
    generate a large, noise-free testing set for which D_{opt} is optimal;
    generate 50 training sets using D_{opt};
    for each training set T:
      greedily induce a tree D on T;
      record D and D_{opt};
  report the mean and std. dev. of the quality measures for the 5000 trees.

The instances in the training and testing sets are always generated uniformly randomly, and are labeled using the optimal decision tree. The size of the testing set is linearly dependent on the complexity of the optimal tree and the dimensionality of the data, whereas the size of the training set is a control variable. More precisely, $|T| = C * (D - 1) * 500$, where $|T|$ is the size of the testing set, C is the size of the optimal tree and D is the number of attributes. For instance, for a size 16 concept in 4 dimensions, we use a testing set of size $16 * (4 - 1) * 500 = 24,000$. We ensure that no subtree of the optimal decision tree is consistent with the testing set.

In all the tables in this paper, each entry comprises of the average value of a tree quality measure over 5000 trees and the standard deviation (one σ). Numbers in parentheses correspond to the optimal trees. The σ's are omitted when they are zero. Optimal values are omitted when their values are obvious. The optimal trees always give 100% prediction accuracy in our experiments, because the testing set has no noise. In addition, they give 100% classification accuracy when the training data is noise-free.

Experiment 1: The purpose of this experiment is to evaluate the effectiveness of greedy induction as a function of the size of the optimal tree. All training sets comprise of 1000 random 2-D instances. There is no noise in the training data. Table 1 summarizes the results.

Observations: The prediction accuracy of greedily induced trees decreases as the size of the optimal tree for the data increases. This can be either be due to the inadequacy of greedy search or due to inadequate training data. (The training set size remained at 1000 though the concept complexity increased from 8 to 64.) In Experiment 2, we increase the size of the training set in proportion with the size of the optimal tree, in order to better isolate the effects due to greedy search.

The difference between the sizes of greedily induced and optimal trees increases with the size of the optimal tree in Table 1. However, it can be seen on closer observation that the variances, not just the differences in size, are increasing. Greedily induced tree sizes are just more than one σ away from the optimal in 3 out of 4 rows, and less than one std. dev. away for concepts of size 64.

The maximum depth measurements in Table 1 show that greedily induced trees can have decision paths which are about twice as long as those in the optimal trees, even for moderately complex concepts. However, the average depth measurements show that the decision paths in greedily induced trees only have about one test more than those in the optimal trees. In terms of the third depth measurement, the expected depth, greedily induced trees are almost *identical* to the optimal ones, for all the concept sizes considered in this experiment. This is a very desirable, although somewhat counterintuitive, trend which is seen consistently throughout our experiments. (Note that no pruning was used in this experiment.)

Experiment 2: The purpose of this experiment is to isolate the effects of concept complexity, from those of the training set size. The size of the training sets now grows linearly with the concept complexity—25 training points on average are used per each leaf of the optimal tree. There is no noise. Table 2 summarizes

Optimal Size	Training Set	Classification Accuracy	Prediction Accuracy	Tree Size	Depth		
					Maximum	Average	Expected
8	200	100.0	97.5±0.7	8.5±1.5	4.4±0.6 (3)	3.4±0.3 (3)	2.8±0.3 (3)
16	400	100.0	97.1±0.7	17.5±3.3	6.6±0.9 (4)	4.7±0.5 (4)	3.8±0.5 (4)
32	800	100.0	96.6±0.7	38.0±7.1	9.1±1.0 (5)	6.2±0.5 (5)	4.8±0.5 (5)
64	1600	100.0	96.4±0.6	76.3±12.3	11.6±1.2 (6)	7.5±0.6 (6)	5.8±0.6 (6)

Table 2: Effects of concept complexity and training set size. No noise. Numbers in parentheses are for the optimal trees.

Class Noise	Classification Accuracy	Prediction Accuracy	Tree Size	Depth		
				Maximum	Average	Expected
0%	100.0 (100.0)	93.9±1.4	31.1±6.2	8.2±0.9	5.6±0.4	4.8±0.5
5%	92.1±1.3 (95.1±0.01)	89.5±2.4	21.9±5.1	7.0±0.8	4.9±0.5	4.4±0.4
10%	87.7±1.3 (90.5±0.02)	88.2±2.6	22.2±5.1	7.0±0.8	4.9±0.4	4.4±0.4
15%	83.5±1.3 (86.1±0.05)	86.6±2.9	22.4±5.4	7.0±0.8	4.9±0.5	4.4±0.4
20%	79.7±1.4 (81.9±0.05)	84.9±3.1	22.7±5.2	7.1±0.8	4.9±0.5	4.4±0.4
25%	76.1±1.4 (77.8±0.03)	83.1±3.4	23.3±5.7	7.1±0.8	4.8±0.5	4.4±0.4

Table 3: Effects of noise in class labels. Numbers in parentheses are for the optimal trees. Optimal trees are of size 32.

the results.

Observations: It is interesting to note that the prediction accuracy does *not* drop as much with increase in optimal tree size here as it does in Experiment 1. In fact, when the optimal trees grew 8 times as large (from 8 to 64), the accuracy went down by just more than one standard deviation. In addition, none of the differences in tree size between greedily induced and optimal trees in Table 2 are more than one σ. This is surprising, considering no pruning was used in this experiment. In terms of the three depth measures, the observations made in Experiment 1 hold here also.

Comparing the entries of Tables 1 and 2, line by line, one can see the effect of the training set size on prediction accuracy. When the training set size increases, the prediction accuracy increases and its variance goes down. In other words, the more (noise-free) training data there is, the more accurately and reliably greedy induction can learn the underlying concept.

Experiment 3: This experiment is intended to evaluate the effectiveness of greedy induction in the presence of noise in class labels. The training sets are all in 2-D, and consist of 100 instances *per class*, uniformly randomly distributed in each class. k% noise is added into each training set by incrementing by 1 the class labels of a random k% of the training points. All concepts are of size 32, so all optimal tree depth values are equal to 5.0. Pruning was used when noise level is greater than 0%. Table 3 summarizes the results.

Observations: As is expected, the classification and prediction accuracies decrease when the amount of noise is increased. The tree size and and depth measurements vary significantly when the first 5% of noise is introduced (obviously because pruning is started), and remain steady thereafter.

One needs to be careful in analyzing the results of experiments 3 and 4, in order to separate out the effects of noise and the effect of the greedy search. What we want to investigate is whether the greedy heuristic becomes less and less effective as the noise levels increase, or if it is robust. For instance, the fact that the classification accuracy decreases linearly with increase in noise in Table 3 is perhaps *not* as significant as the fact that the prediction accuracy decreases more slowly than classification accuracy. This is because the former is an obvious effect of noise whereas the later indicates that greedy induction might be *compensating* for the noise.

Several patterns in Table 3 argue in favor of the effectiveness of pruning, which has come to be an essential part of greedy tree induction. Classification accuracies of the greedy trees are close to, and *less than*, those of the optimal trees for all the noise levels, so overfitting is not a problem. Prediction accuracies of greedily induced trees with pruning are *better* than their classification accuracies, again indicating that there is no strong overfitting. Tree size and depth measurements remained virtually unchanged in the presence of increasing noise, certifying to the robustness of pruning.

Experiment 4: This experiment is similar to the previous one, in that we measure the effectiveness of greedy induction as a function of noise in the training data. However, this time we consider noise in attribute values. The training sets again comprise 100 2-D instances per class, uniformly randomly distributed in each class. k% noise is introduced into each training

Attribute Noise	Classification Accuracy	Prediction Accuracy	Tree Size	Depth		
				Maximum	Average	Expected
0%	100.0 (100.0)	93.9±1.4	31.1±6.2	8.2±0.9	5.6±0.4	4.8±0.5
5%	95.2±1.3 (98.0±0.4)	90.0±2.3	22.2±5.3	7.0±0.8	4.9±0.5	4.4±0.4
10%	93.5±1.4 (96.0±0.7)	88.7±2.6	22.6±5.5	7.0±0.8	4.9±0.5	4.4±0.4
15%	92.1±1.6 (94.1±1.0)	87.4±2.8	23.3±5.6	7.0±0.8	5.0±0.5	4.4±0.4
20%	90.7±1.8 (92.2±1.3)	86.2±3.1	23.7±5.6	7.0±0.8	4.9±0.5	4.4±0.4
25%	89.4±2.0 (90.6±1.6)	85.0±3.4	23.7±5.5	7.0±0.8	4.9±0.5	4.3±0.4

Table 4: Effects of noise in attribute values. Numbers in parentheses are for optimal trees. Optimal trees are of size 32.

#Dim.	Classification Accuracy	Prediction Accuracy	Tree Size	Depth		
				Maximum	Average	Expected
2	100.0	98.7±0.3	20.7±3.3	7.2±1.0	5.0±0.4	3.9±0.4
4	100.0	98.3±0.7	23.9±6.0	6.6±0.9	5.0±0.5	4.0±0.4
8	100.0	98.0±0.8	24.5±6.5	6.3±0.9	4.9±0.5	4.1±0.2
12	100.0	97.9±0.9	25.4±6.8	6.3±0.9	4.9±0.5	4.1±0.2

Table 5: Effects of dimensionality. Training set size=1000. No noise. Optimal trees are of size 16.

set by choosing a random $k\%$ of the instances, and by adding an $\epsilon \in [-0.1, 0.1]$ to each attribute. All the concepts are of size 32, so all the optimal depth measurements are equal to 5.0. Cost complexity pruning was used in cases where the noise level was greater than 0%. The results are summarized in Table 4.

Observations: There results with attribute noise (Table 4) and noise in class labels (Table 3) are very similar, except for the classification accuracies. The values for prediction accuracy, tree size and depth measurements in the presence of $k\%$ noise are almost the same whether the noise is in attribute values or class labels. The classification and prediction accuracies decrease with increasing noise. The tree size and depth measurements decrease when the first 5% of the noise is introduced (due to pruning) and remain steady thereafter.

However, introducing $k\%$ attribute noise is not equivalent to introducing $k\%$ class noise. Changing the attributes of an instance by a small amount affects the classification of only those instances lying near the borders of decision regions, whereas changing the class labels affects the classification of all the instances involved. This can be seen from the classification accuracies of the optimal trees in Tables 3 and 4. The classification accuracy of the greedy trees is quite close to, and less than that of the optimal trees in both tables. All the prediction accuracy values in Table 4, unlike those in Table 3, are less than the corresponding classification accuracies.

Experiment 5: Our final experiment attempts to quantify the effect of dimensionality on the greedy heuristic. All the training sets consist of 1000 uniformly randomly generated instances, with no noise, as in Experiment 1. No pruning was used. All concepts are of size 16, so the optimal tree depths are 4.0. Table 5 summarizes the results.

Observations: The changes in all tree quality measures are quite small when dimensionality is increased from 2 to 12. This result is surprising because, intuitively, higher dimensional concepts should be much more difficult to learn than lower dimensional ones, when the amount of available training data does not change. Our experiments indicate that the effects due to dimensionality do not seem to be as pronounced as the effects due to concept complexity (Table 1) or noise. The quantity that does increase with increasing dimensionality is the variance. Both prediction accuracy and tree size fluctuate significantly more in higher dimensions than in the plane. This result suggests that methods that help decrease variance, such as combining the classifications of multiple decision trees (see (Murthy 1995) for a survey), may be useful in higher dimensions.

Discussion and Conclusions

In this paper, we presented five experiments for evaluating the effectiveness of the greedy heuristic for decision tree induction. In each experiment, we generated thousands of random training sets, and compared the decision trees induced by C4.5 and CART to the corresponding optimal trees. The optimal trees were found using a novel graph coloring idea.

We summarize the main observations from our experiments below. Where relevant, we briefly mention related work in the literature.

- *The expected depth of greedily induced decision trees was consistently very close to the optimal.* Garey

and Graham (1974) showed that a recursive greedy splitting algorithm using information gain (*not* using pruning) can be made to perform arbitrarily worse than the optimal in terms of expected tree depth. Goodman and Smyth (1988) argued, by establishing the equivalence of decision tree induction and a form of Shannon-Fano prefix coding, that the average depth of trees induced by greedy one-pass (i.e., no pruning) algorithms is nearly optimal.

- *Cost complexity pruning* (Breiman *et al.* 1984) *dealt effectively with both attribute and class noise.* However, the accuracies on the training set were overly optimistic in the presence of attribute noise.
- Greedily induced trees became less accurate as the concepts became harder, i.e., as the optimal tree size increased. However, *increasing the training data size linearly with concept complexity helped keep the accuracy stable.*
- Greedily induced trees were not much larger than the optimal, even for complex concepts. However, *the variance in tree size was more for higher dimensional and more complex concepts.* Dietterich and Kong (1995) empirically argued that even in terms of prediction accuracy, variance is the main cause for the failure of decision trees in some domains.
- *For a fixed training set size, increasing the dimensionality did not affect greedy induction as much as increasing concept complexity or noise did.* Several authors (e.g., (Fukanaga & Hayes 1989)) have argued that for a finite sized data with no *a priori* probabilistic information, the ratio of training sample size to the dimensionality must be as large as possible. Our results are consistent with these studies. However, with a reasonably large training set (1000 instances), the drop in tree quality was quite small in our experiments, even for a 6-fold (2 to 12) increase in dimensionality.
- *The goodness measures of CART and C4.5 were identical in terms of the quality of trees they generated.* It has been observed earlier (e.g.,(Breiman *et al.* 1984; Mingers 1989)) that the differences between these goodness criteria are not pronounced. Our observation that these measures consistently produced identical trees, in terms of six tree quality measures, in a large scale experiment (involving more than 130,000 synthetic data sets) strengthens the existing results. [2]

Many researchers have studied ways to improve upon greedy induction, by using techniques such as limited lookahead search and more elaborate classifier representations (e.g., decision graphs instead of trees). (See (Murthy 1995) for a survey.) The results in the current paper throw light on why it might be difficult to improve upon the simple greedy algorithm for decision tree induction.

[2]The fact that we only used binary splits in real-valued domains may be one reason why information gain, Gini index and twoing rule behaved similarly.

References

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Trees.* Wadsworth International Group.

Cox, L. A.; Qiu, Y.; and Kuehner, W. 1989. Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Annals of Operations Research* 21(1):1–30.

Dietterich, T. G., and Kong, E. B. 1995. Machine learning bias, statistical bias and statistical variance of decision tree algorithms. In *Machine Learning: Proceedings of the Twelfth International Conference.* Tahoe City, CA: Morgan Kaufmann Publishers Inc., San Mateo, CA. to appear.

Fukanaga, K., and Hayes, R. A. 1989. Effect of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11:873–885.

Garey, M. R., and Graham, R. L. 1974. Performance bounds on the splitting algorithm for binary testing. *Acta Informatica* 3(Fasc. 4):347–355.

Goodman, R. M., and Smyth, P. J. 1988. Decision tree design from a communication theory standpoint. *IEEE Transactions on Information Theory* 34(5):979–994.

Hyafil, L., and Rivest, R. L. 1976. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters* 5(1):15–17.

Mingers, J. 1989. An empirical comparison of selection measures for decision tree induction. *Machine Learning* 3:319–342.

Murphy, O. J., and McCraw, R. L. 1991. Designing storage efficient decision trees. *IEEE Transactions on Computers* 40(3):315–319.

Murthy, S. K.; Kasif, S.; and Salzberg, S. 1994. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research* 2:1–33.

Murthy, S. K. 1995. Data exploration using decision trees: A survey. In preparation. http://www.cs.jhu.edu/grad/murthy.

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