Toward a Theoretical Understanding of Why and When Decision Tree Pruning Algorithms Fail

Tim Oates and David Jensen
Experimental Knowledge Systems Laboratory
Department of Computer Science
Box 34610 LGRC
University of Massachusetts
Amherst, MA 01003-4610
{oates, jensen}@cs.umass.edu

Abstract

Recent empirical studies revealed two surprising pathologies of several common decision tree pruning algorithms. First, tree size is often a linear function of training set size, even when additional tree structure yields no increase in accuracy. Second, building trees with data in which the class label and the attributes are independent often results in large trees. In both cases, the pruning algorithms fail to control tree growth as one would expect them to. We explore this behavior theoretically by constructing a statistical model of reduced error pruning. The model explains why and when the pathologies occur, and makes predictions about how to lessen their effects. The predictions are operationalized in a variant of reduced error pruning that is shown to control tree growth far better than the original algorithm.

Introduction

Despite more than three decades of intense research on decision trees, arguably the most commonly used learning mechanism in implemented AI systems, existing characterizations of their behavior are overwhelmingly empirical rather than theoretical. There is currently a large gap between the algorithms and representations that appear to be amenable to theoretical analysis on the one hand, and decision trees on the other. Empirical studies have identified solutions to various parts of the overall process of building decision trees that work well in a broad set of circumstances. However, making precise statements about when and, perhaps more importantly, why those solutions are either appropriate or inappropriate remains difficult.

This paper attempts to narrow the gap between theory and practice by presenting a statistical model that explains one particularly surprising pathology of several common pruning algorithms that occurs with data devoid of structure. The pathology is illustrated in Figure 1, which plots tree size as a function of dataset size for three common pruning techniques – error-based (EBP) (Quinlan 1993), reduced error (REP) (Quinlan 1987), and minimum description length (MDL) (Quinlan & Rivest 1989).[1] All trees were built with C4.5. The datasets contained 30 binary attributes and a binary class label, all with values assigned randomly from a uniform distribution. There was no relationship between the attributes and the class label. Given such datasets, one would expect pruning algorithms to emit trees with a single node – a leaf labeled with the majority class. This does not happen. Trees built with these data exhibit an almost perfectly linear relationship between the amount of structureless data used to build the tree and the size of the final pruned tree.

Although the phenomenon depicted in Figure 1 is most clearly demonstrated with structureless artificial data, it occurs in a broad range of real world datasets (Oates & Jensen 1997; 1998) because they contain subsets of instances with no structure (or structure that cannot be identified by tree growing algorithms). Decision tree growing algorithms typically do not stop splitting the data precisely when all of the structure in the data has been captured. Instead, they push past that point, splitting subsets of the data wherein the attributes and the class label are either totally or nearly independent, leaving it to the pruning phase to find the "correct" tree. The result is that some number of subtrees in the unpruned tree are constructed through recursive invocations of the tree growing algorithm on structureless data, such as that used in Figure 1. The question that remains to be answered is why trees (and subtrees) built from such data escape pruning.

To better understand why several well-studied pruning algorithms leave large amounts of excess structure in trees, we developed a statistical model of one particular algorithm – REP. Analysis of the model provides insights into why and under what conditions REP fails to control tree growth as it should. For example, we identify two properties that hold for almost every deci-

[1] The horizontal axis is the total number of instances available to the tree building process. Each point in the plot reflects the result of 10-fold cross validation, so the actual number of instances used to build the trees is 9/10 of the corresponding position on the horizontal axis. In addition, REP further split the data into a growing set (2/3 of the data) and a pruning set (1/3 of the data).

sion node rooting a subtree that fits noise. They are:

- The probability of pruning such nodes *prior to pruning beneath them* is close to 1. That is especially true for large pruning sets.
- Pruning that occurs beneath such nodes often has the counterintuitive effect of reducing the probability that they will be pruned to be close to 0.

Insights gleaned from the model led to the development of a novel variant of REP that yields significantly smaller trees with accuracies that are comparable to those of trees pruned by the original algorithm. Rather than considering all of the available pruning data when making pruning decisions, the new algorithm selects a randomly sampled subset of that data prior to making each decision. The degree of overlap between samples is a user controlled parameter, with 100% overlap corresponding to standard REP, and 0% overlap (which is feasible when large amounts of data are available) virtually eliminating the effect shown in Figure 1. Other degrees of overlap can be chosen depending on the amount of data available.

The remainder of the paper is organized as follows. The next section presents the statistical model of REP, and the following section discusses implications of the model, including an explanation for the behavior shown in Figure 1. We then present the variant of REP based on the theoretical model. The final section concludes and points to future work.

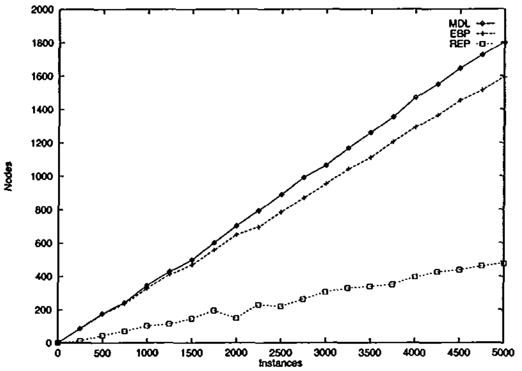

Figure 1: Tree size as a function of dataset size for three common pruning techniques when the class labels are independent of the attributes.

A Statistical Model of Reduced Error Pruning

This section presents a statistical model of REP. The goal is to model pruning decisions at nodes for which the class label and the attribute values of instances are independent, i.e. where there is no structure in the data that, if found, would make it possible to predict the class label better than always guessing the majority class. Independence holds at node N when all of the attributes with any utility in predicting the class label have already been used to split the data on the path from the root of the tree to N. Ideally, such nodes will always be pruned, but as we saw in the previous section, that is often not the case. The model will make it possible to make probabilistic statements about the likelihood of pruning under various conditions.

REP was chosen for analysis primarily because of its simplicity. The algorithm takes as input an unpruned tree and a set of instances, called the pruning set, drawn from the same population as the set of instances used to build the tree, but disjoint from that set. To begin, the unpruned tree is used to classify all of the instances in the pruning set. Let D_N be the subset of the pruning instances that pass through decision node N on their way to leaf nodes. The subtree rooted at N, denoted T_N, commits some number of errors on the instances in D_N. Let that number be r_T. If T_N is pruned back to a leaf and assigned as a class label the majority class in D_N, then, assuming a binary class label, it will commit a number of errors equal to the number of instances in the minority class.[2] Let that number be r_L. In a bottom-up manner, r_T and r_L are computed for each decision node, and T_N is pruned when the number of errors committed by the tree is greater than or equal to the number of errors committed by the leaf, i.e. when $r_T \geq r_L$.

The intuition behind REP is appealing. The number of errors that a subtree commits on the training data is clearly biased downward because the tree was constructed to minimize errors on this set, but the number of errors committed on an independent sample of data, the pruning set, is unbiased. Where the unpruned tree fits noise in the training data (i.e. is overfitting those data) the tree should perform poorly on the pruning set, making it likely that pruning will occur. (We will prove this assertion in the following section.) Where the tree is fitting structure in the data, pruning back to a leaf should result in more errors than retaining the subtree. Given unbiased error estimates, the behavior shown in Figure 1 seems inexplicable.

To model pruning decisions, and thus to explain Fig-

[2] The original formulation of REP as described in (Quinlan 1987) uses the majority class in the training set rather than the pruning set to assign class labels when pruning. The analysis in the remainder of the paper makes the simplifying assumption that the pruning set is used to assign class labels. In practice, there is very little difference between the two approaches. To verify that assertion, we took 19 different datasets (the same ones used in (Oates & Jensen 1998)) and built trees on 10 different splits of the data, with 2/3 of the data being used to build the tree and 1/3 used to prune the tree. For every one of those 190 trees, we counted the number of nodes for which the class label assigned by the training set was the same as the label assigned by the pruning set. To avoid spurious differences near the leaves of the trees, nodes with fewer than 10 pruning set instances were ignored. Over those 19 datasets, 94% of the nodes were labeled identically by the training and pruning sets.

ure 1, we must characterize r_T and r_L because they are the only quantities that enter into pruning decisions. Given D_N, determining the value of r_L is straightforward. Let $n = |D_N|$ be the number of pruning set instances that arrive at N. Assuming a binary class label, which will be the case for the remainder of the paper, let p_C be the probability that an instance in D_N is labeled with class '+'. (That probability is simply the number of instances in D_N labeled '+' divided by n.) Then the value of r_L is given by the following equation:

$$r_L = n \min(p_C, 1 - p_C)$$

If $p_C < 1 - p_C$ the majority class in D_N is '-'. Pruning T_N will result in a leaf labeled '-', and all of the np_C instances in D_N labeled '+' will be misclassified. If $1 - p_C < p_C$ the majority class in D_N is '+', and after pruning all of the $n(1 - p_C)$ instances in D_N labeled '-' will be misclassified.

Characterization of r_T is more difficult because its value depends on the tree structure beneath N. Without exact knowledge of the instances in D_N and of T_N, the exact value of r_T cannot be determined. However, the distribution of values from which r_T is drawn can be characterized. Let R_T be a random variable that represents the number of errors committed by T_N on D_N. Let the probability that a randomly selected instance in D_N will arrive at a leaf labeled '+' be p_L.[3] The probability that the subtree will misclassify an instance in D_N, which we will denote $p_{C \neq L}$, is the probability that an instance labeled '+' will arrive at a leaf labeled '-' plus the probability that an instance labeled '-' will arrive at a leaf labeled '+'. Because the class label and attribute values are independent, that quantity is given by the following equation:

$$\begin{aligned} p_{C \neq L} &= p_C(1 - p_L) + (1 - p_C)p_L \\ &= p_C + p_L - 2p_C p_L \end{aligned}$$

We can think of assigning a class label to an instance in D_N as a Bernoulli trial in which the two outcomes are an incorrect classification (which occurs with probability $p_{C \neq L}$) and a correct classification (which occurs with probability $1 - p_{C \neq L}$). Therefore, the number of errors committed by T_N on D_N has a binomial distribution with mean $\mu_T = np_{C \neq L}$ and standard deviation $\sigma_T = \sqrt{np_{C \neq L}(1 - p_{C \neq L})}$ which, for large n, can be approximated by a normal distribution. There is no way to precisely specify when n is large enough to use the normal approximation, but one commonly used rule of thumb is that it can be used when both $np_{C \neq L}$ and $n(1 - p_{C \neq L})$ are greater than five (Olson 1987).

The model is shown graphically in Figure 2. Given n, p_C and p_L, an exact value of r_L can be determined. However, we can only say that r_T is drawn from a normal distribution with known mean and standard deviation. The node will be pruned if the value drawn from that distribution is greater than or equal to r_L.

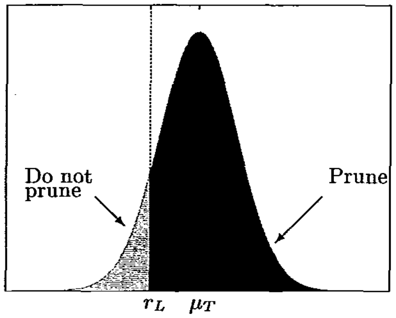

Figure 2: Given n, p_C and p_L, an exact value for r_L can be determined, and r_T is drawn from a normal distribution with know mean and standard deviation. The node will be pruned if $r_T \geq r_L$.

Implications of the Model

This section uses the model just presented to derive several results that provide insight into the behavior of REP. At a high level, there are two important conclusions, both concerning decision nodes rooting subtrees that fit noise. First, the probability of pruning such nodes *prior to pruning beneath them* is close to 1. Second, pruning that occurs beneath such nodes often has the counterintuitive effect of reducing the probability that they will be pruned to be close to 0.

The Probability of Pruning a Node Prior to Pruning Beneath It

It is easy to show that the expected number of errors committed by subtree T_N on D_N is greater than the number of errors committed by the leaf that results from pruning T_N. The expected number of errors for the subtree is $E(R_T)$, which is simply μ_T. In terms of Figure 2, the fact that $E(R_T) > r_L$ means that r_L is always to the left of the mean of the distribution.

Theorem 1 *For $p_C \neq 0.5$, $E(R_T) > r_L$.*

Proof: There are two cases to consider. Either $1/2 > p_C$ or $1/2 < p_C$.

Case 1:

$$\begin{aligned} 1/2 &> p_C \\ 1 &> 2p_C \\ p_L &> 2p_C p_L \\ p_L - 2p_C p_L &> 0 \\ p_C + p_L - 2p_C p_L &> p_C \\ n(p_C + p_L - 2p_C p_L) &> np_C \\ np_{C \neq L} &> np_C \\ \mu_T &> np_C \\ E(R_T) &> r_L \end{aligned}$$

[3] Although p_L depends on T_N and D_N just as r_T does, we will see later that the actual value of p_L has little qualitative effect on the predictions of the model.

Case 2:

$$\begin{aligned}
p_C &> 1/2 \\
2p_C &> 1 \\
2p_C(1-p_L) &> 1-p_L \\
2p_C - 2p_Cp_L &> 1-p_L \\
2p_C + p_L - 2p_Cp_L - 1 &> 0 \\
2p_C + p_L - 2p_Cp_L &> 1 \\
p_C + p_L - 2p_Cp_L &> 1-p_C \\
n(p_C + p_L - 2p_Cp_L) &> n(1-p_C) \\
np_{C\neq L} &> n(1-p_C) \\
\mu_T &> n(1-p_C) \\
E(R_T) &> r_L
\end{aligned}$$

■

Deriving an expression for the probability that T_N will be pruned is straightforward as well. It is simply the probability that $r_T \geq r_L$, which is the area under the normal distribution to the right of r_L in Figure 2. Let $\Phi(\mu, \sigma, x)$ be the cumulative density up to x of the normal distribution with mean μ and standard deviation σ. $\Phi(\mu_T, \sigma_T, r_L)$ is the area to the left of r_L in Figure 2, so the area to the right of that point is:

$$p_{T\rightarrow L} = 1 - \Phi(\mu_T, \sigma_T, r_L) \qquad (1)$$

Figure 3 shows plots of $p_{T\rightarrow L}$ for all possible values of p_C and p_L at various levels of n. When the class labels of instances in D_N are distributed evenly ($p_C = 0.5$) the probability of pruning is 0.5 ($p_{T\rightarrow L} = 0.5$) regardless of the value of p_L. However, that probability rapidly approaches 1 as you move away from the $p_C = 0.5$ line, with the steepness of the rise increasing with n. That is, for all values of p_C and p_L, you are more likely to prune a subtree that fits noise the more pruning instances are available. For example, numerical integration of the curves in Figure 3 shows that the average of $p_{T\rightarrow L}$ when $n = 100$ is 0.926. That number is 0.970 when $n = 1000$ and 0.988 when $n = 10,000$. Unless p_C is very close to 0.5, pruning of subtrees that fit noise in the data is virtually assured *given that no pruning has occurred beneath them.* Note that most decision tree splitting criteria either explicitly or implicitly choose the split that maximizes purity of the data. Said differently, they attempt to move p_C as far away from 0.5 as possible.

The intuition that $p_{T\rightarrow L}$ increases with n, all other things being equal, is now made rigorous. Let $p_{T\rightarrow L}(p_C, p_L, n)$ be the probability of pruning given the specified values of p_C, p_L and n.

Theorem 2 *For* $p_C \neq 0.5$ *and* $\delta > 0$, *it holds that* $p_{T\rightarrow L}(p_C, p_L, n) < p_{T\rightarrow L}(p_C, p_L, n+\delta)$.

Proof: First, we manipulate Equation 1 so that Φ refers to the standard normal:

$$\begin{aligned}
p_{T\rightarrow L} &= 1 - \Phi(\mu_T, \sigma_T, r_L) \\
&= 1 - \Phi(0, 1, \frac{r_L - \mu_t}{\sigma_T}) \\
&= 1 - \Phi(0, 1, z)
\end{aligned}$$

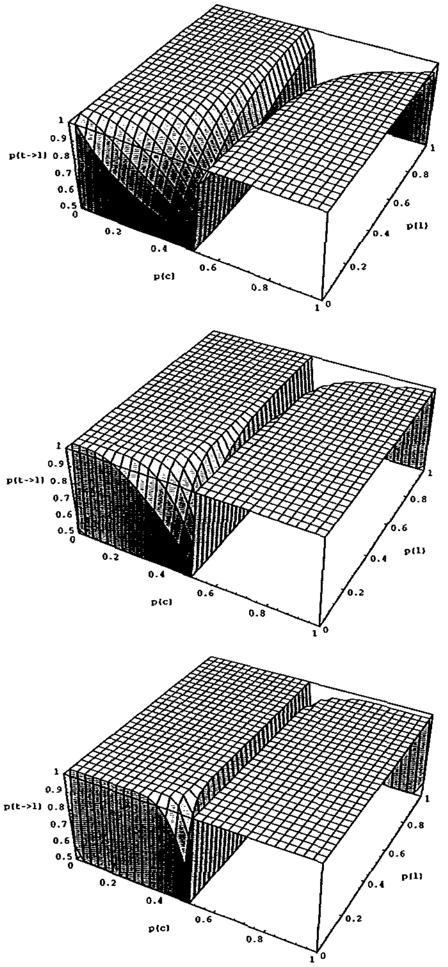

Figure 3: Plots of $p_{T\rightarrow L}$ for p_C and p_L ranging from 0 to 1 and $n = 100$ (the top graph), $n = 1000$ (the middle graph), and $n = 10,000$ (the bottom graph).

Φ increases monotonically as z increases, so $p_{T\rightarrow L}$ decreases monotonically as z increases. That is, the probability of pruning is inversely related to z. The question then becomes, what is the effect of changing n on z?

$$\begin{aligned}
z &= \frac{r_L - \mu_t}{\sigma_T} \\
&= \frac{n\min(p_C, 1-p_C) - np_{C\neq L}}{\sqrt{np_{C\neq L}(1-p_{C\neq L})}} \\
&= \sqrt{n}\frac{(\min(p_C, 1-p_C) - p_{C\neq L})}{\sqrt{p_{C\neq L}(1-p_{C\neq L})}} \\
&= \sqrt{n}K
\end{aligned}$$

Because $r_L < \mu_T$ (see Theorem 1) and σ_T is non-negative (from the definition of the standard deviation), the quantity $(r_L - \mu_t)/\sigma_T$ is always negative.

It follows that K is always negative as well. Because K does not depend on n, $\sqrt{(n+\delta)}K < \sqrt{n}K$. Coupling that fact with the previous observation that $p_{T\to L}$ increases monotonically with decreasing z, we conclude that $p_{T\to L}(p_C, p_L, n) < p_{T\to L}(p_C, p_L, n+\delta)$. ■

The Probability of Pruning a Node After Pruning Beneath It

How does pruning that occurs beneath a decision node affect the probability that the node itself will ultimately be pruned? Recall that the number of errors committed by T_N is defined recursively to be the sum of the errors committed by N's children. Because pruning occurs when the leaf commits the same number or fewer errors than the subtree, if any of the descendants of N are pruned, r_T (the number of errors committed by T_N) must either stay the same or decrease. In effect, there are two values of r_T in which we are interested. There is the value that exists prior to pruning beneath N, and there is that value that exists when all of the descendants of N have been visited by the pruning procedure and a decision is about to be made concerning N. Denote the latter value r'_T.

Let r_{Ndi} be the number of errors committed by the i^{th} descendant of N at depth d after a pruning decision has been made concerning that descendant. If no leaf nodes occur in T_N until depth $d+1$, then r'_T is simply $\sum_i r_{Ndi}$. Assume that N and all of its descendants at depth d share the same values of p_C and p_L.[4] If each of the subtrees rooted at those descendants is pruned, then r'_T becomes the following:

$$\begin{aligned} r'_T &= \sum_i r_{Ndi} \\ &= \sum_i n_i \min(p_C, 1-p_C) \\ &= n \min(p_C, 1-p_C) \\ &= r_L \end{aligned}$$

That is, the number of errors committed by the subtree rooted at N will be the same as the number of errors committed when that subtree is pruned back to a leaf, and so the subtree will be pruned.

Now consider what happens if just one of the descendants at depth d is not pruned. If descendant k is not pruned, then $r_{Ndk} < n_k \min(p_C, 1-p_C)$, so the sum above becomes:

$$\begin{aligned} r'_T &= \sum_{i\neq k} r_{Ndi} + r_{Ndk} \\ &= \sum_{i\neq k} n_i \min(p_C, 1-p_C) + r_{Ndk} \\ &= (n-n_k)\min(p_C, 1-p_C) + r_{Ndk} \\ &< (n-n_k)\min(p_C, 1-p_C) + n_k \min(p_C, 1-p_C) \\ &< n \min(p_C, 1-p_C) \\ &< r_L \end{aligned}$$

If just one of the descendants of N at depth d is not pruned, T_N will be retained. If more than one descendant is not pruned, T_N will still be retained as that can only decrease r'_T. Said differently, N will be pruned only if all of its descendants at depth d are pruned.

We can now derive an expression for the probability of pruning a subtree given that pruning has occurred beneath it. Let $p'_{T\to L}$ denote that probability. (As with r_T and r'_T, the prime indicates the value of the variable at the time that a pruning decision is to be made for the node.) Let the number of descendants at depth d be m, and let p_i be the probability that the i^{th} descendant will be pruned. Then the following holds:

$$p'_{T\to L} = \prod_{i=1}^{m} p_i$$

The value of d has two effects on $p'_{T\to L}$ that may not be immediately obvious. First, as d increases, m increases exponentially, leading to an exponential decrease in $p'_{T\to L}$. Second, as d increases, the number of pruning set instances that reach each of the descendants decreases exponentially. As Theorem 2 makes clear, that decrease leads to a decrease in the value of p_i, and thus to a dramatic decrease in $p'_{T\to L}$.

Figure 4 shows plots of $p'_{T\to L}$ for all possible values of p_C and p_L. The top graph assumes a subtree rooted at a node with 10 descendants, the middle graph assumes 20 descendants, and the bottom graph assumes 50 descendant. All three graphs assume that each descendant has 5 pruning instances. In each case, there are large regions in which the probability of pruning is virtually zero. Only when p_C and p_L are very different (i.e. where $|p_C - p_L| \approx 1$) is the probability of pruning close to one. Numerical integration of the curves in Figure 4 shows that the average value of $p'_{T\to L}$ is 0.33 when the number of descendants is 10. The average is 0.25 and 0.18 when the number of descendants is 20 and 50 respectively. As the number of descendants of a node increases, the probability of pruning that node decreases.

There are two important observations to make about Figure 4. First, as a practical matter, combinations of p_C and p_L yielding a value of $|p_C - p_L|$ close to one are rare. Such a combination would indicate that the class distributions in the training and pruning sets are vastly different, and should not be the case when the two samples are drawn from the same population. The implication is that the effective probability of pruning in any scenario that is likely to occur is much lower than the averages mentioned above. Second, Figure 4 involves exactly the same quantities as Figure 3, only the latter is plotted prior to pruning beneath a node and the former is plotted after pruning beneath a node.

[4] To a first approximation, that is a good assumption when there is no structure in the data and n is large.

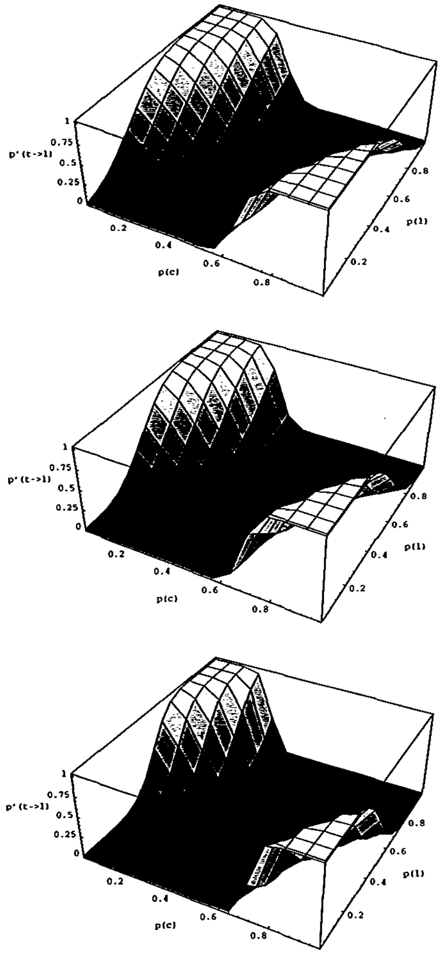

Figure 4: Plots of $p'_{T\to L}$ for p_C and p_L ranging from 0 to 1, where N has 10 descendants (top), 20 descendants (middle) and 50 descendants (bottom).

An Improved REP Algorithm

As demonstrated in the previous section, the probabilities of pruning node N before and after pruning has occurred beneath it are often quite different (i.e. $p_{T\to L} \gg p'_{T\to L}$). This phenomenon exists because pruning decisions are made so as to minimize errors on the pruning set, and the number of errors committed by the subtree rooted at N is a sum of the errors committed by N's descendants at any given depth (assuming no leaf nodes exist above that depth). This observation suggests that excess tree structure might be avoided if a pruning set different from the one used to prune N's descendants was available to prune N itself. The reason is that although N's descendants were pruned so as to minimize errors on the original pruning set, that has no effect on whether N will be pruned when a new pruning set is obtained just for N.

The above intuition was tested on three artificial datasets: the `rand` structureless dataset that was used to create Figure 1, and the `led-24` and `waveform-40` datasets taken from the UC Irvine repository. Figure 5 shows tree size and accuracy as a function of training set size given that a *completely new pruning set* is generated prior to making each pruning decision. Creating new pruning sets is possible with these particular artificial datasets, and may be possible with very large real-world datasets as are common in the KDD community. Note that over all training set sizes, the trees pruned with non-overlapping pruning sets are smaller and just as accurate (determined by a t-test at the 0.05 significant level) as the trees pruned with a single pruning set.

Generating completely new pruning sets for each node is often impractical. However, it is possible that part of the benefit of totally independent pruning sets can be obtained by drawing a random sample of the available pruning instances for each node. Specifically, given m pruning instances, a random sample of size αm where $0 < \alpha \leq 1$ can be drawn. Note that $\alpha = 1$ corresponds to standard REP; the same pruning set (the full pruning set) is used at each node. Smaller values of α lead to less overlap in the pruning sets used at each node and make those pruning sets smaller.

We ran this variant of REP with $\alpha = 0.5$ on the 19 datasets used in (Oates & Jensen 1998) and found that sampling led to significantly smaller trees in 16 cases, and in only one of those cases was accuracy significantly less. (Significance was measured by a t-test comparing means of 10-fold cross-validated tree sizes and accuracies with a significance level of 0.05.) Accuracy was lower on the `tic-tac-toe` dataset because it is noise-free and all of the attributes are relevant (much like the parity problem). Pruning trees built on that dataset almost invariably leads to lower accuracy, with more aggressive pruning leading to additional losses.

Discussion and Future Work

Despite the use of pruning algorithms to control tree growth, increasing the amount of data used to build a decision tree, even when there is no structure in the data, often yields a larger tree that is no more accurate than a tree built with fewer data. A statistical model of REP led to a theoretical understanding of why this behavior occurs, and to a variant of REP that results in significantly smaller trees with the same accuracies as trees pruned with the standard REP algorithm.

Future work will include generalizing the results in this paper to other pruning algorithms, such as MDL and EBP (Oates & Jensen 1997), and to further exploration of the idea of using partially overlapping samples of pruning sets to minimize excess structure in trees.

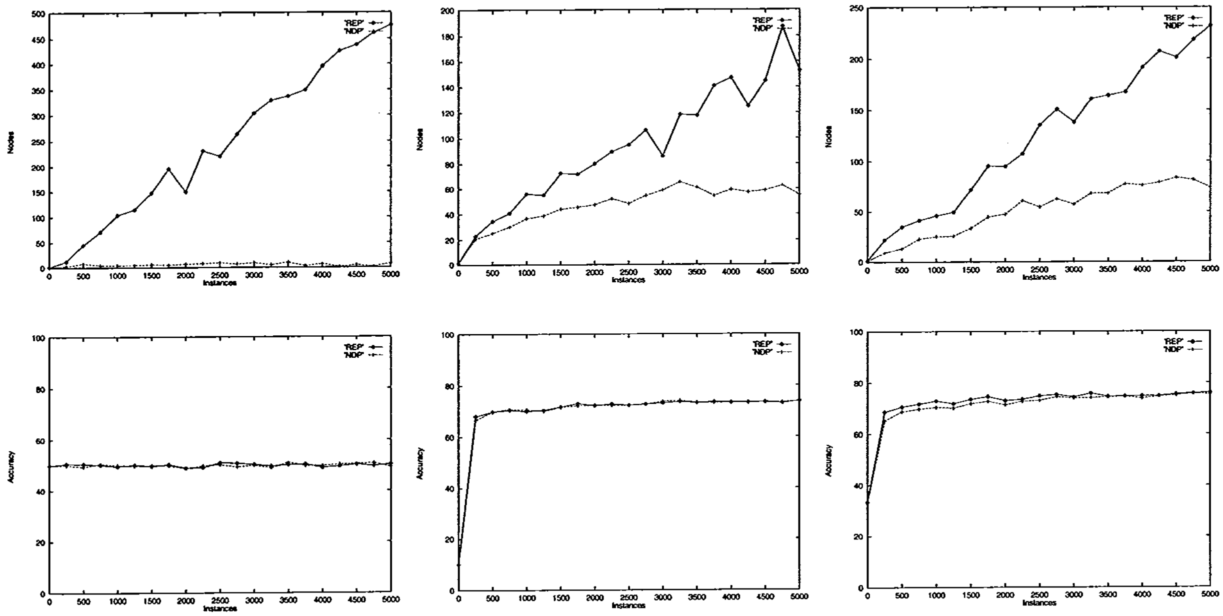

Figure 5: Tree size (top row) and accuracy (bottom row) as a function of training set size for the following datasets: rand (first column), led-24 (middle column) and waveform-40 (last column). In each plot, the REP curve corresponds to standard REP, and the NDP curves refers to a variant of the algorithm in which a completely new pruning set is generated and classified by the tree prior to making each pruning decision.

Acknowledgments

This research is supported by DARPA/AFOSR under contract number F49620-97-1-0485. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Office of Scientific Research or the U.S. Government.

References

Oates, T., and Jensen, D. 1997. The effects of training set size on decision tree complexity. In *Proceedings of The Fourteenth International Conference on Machine Learning*, 254–262.

Oates, T., and Jensen, D. 1998. Large datasets lead to overly complex models: an explanation and a solution. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 294–298.

Olson, C. 1987. *Statistics: Making Sense of Data.* Allyn and Bacon.

Quinlan, J. R., and Rivest, R. 1989. Inferring decision trees using the minimum description length principle. *Information and Computation* 80:227–248.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27:221–234.

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