The Effects of Training Set Size on Decision Tree Complexity

Abstract

This paper presents experiments with 19 datasets and 5 decision tree pruning algorithms that show that increasing training set size often results in a linear increase in tree size, even when that additional complexity results in no significant increase in classification accuracy. Said differently, removing randomly selected training instances often results in trees that are substantially smaller and just as accurate as those built on all available training instances. This implies that decreases in tree size obtained by more sophisticated data reduction techniques should be decomposed into two parts: that which is due to reduction of training set size, and the remainder, which is due to how the method selects instances to discard. We perform this decomposition for one recent data reduction technique, John's ROBUST-C4.5 (John 1995), and show that a large percentage of its effect on tree size is attributable to the fact that it simply reduces the size of the training set. We conclude that random data reduction is a baseline against which more sophisticated data reduction techniques should be compared. Finally, we examine one possible cause of the pathological relationship between tree size and training set size.

1 Introduction

Data preprocessing is becoming increasingly popular as a way to improve the performance of decision tree algorithms. Often such techniques involve *data reduction*, the removal of training instances prior to tree construction. For example, some techniques identify instances that are "bad" and remove them from the training set, while others actively build a training set from available instances by selecting those that are "good". Whether the explicit goal of any given technique is increased accuracy or smaller trees, the latter is invariably observed. John's ROBUST-C4.5 treats misclassified training instances as outliers, iteratively removing them and building a new tree (John 1995). The result over a large number of datasets is trees that are much smaller than those built by C4.5, but that have roughly equivalent accuracy. Brodley and Friedl developed a method to remove instances deemed mislabeled (e.g. by transcription errors) in an effort to boost accuracy. They observe that such filtering, as an unanticipated side-effect, leads to substantially smaller trees (Brodley & Friedl 1996).

In this paper we argue that, under a broad range of circumstances, all data reduction techniques will result in some decrease in tree size with little impact on accuracy. Section 2 offers detailed empirical evidence for the validity of this claim, but an intuitive feeling for why it might be true can be grasped by looking at Figure 1. The figure shows plots of tree size and accuracy as a function of *training set size* for the UC Irvine **australian** dataset. C4.5 was used to generate the trees (Quinlan 1993) and each plot corresponds to a different pruning mechanism: error-based (EBP – the C4.5 default) (Quinlan 1993), reduced error (REP) (Quinlan 1987), minimum description length (MDL) (Quinlan & Rivest 1989), cost-complexity with the 1SE rule (CCP1SE) (Breiman *et al.* 1984), and cost-complexity without the 1SE rule (CCP0SE). On the left-hand side of the graphs, no training instances are available and the best one can do with test instances is to assign them a class label at random. On the right-hand side of the graph, the entire dataset (excluding test instances) is available to the tree building process. Movement from the the left to the right corresponds to the addition of randomly selected instances to the training set. Alternatively, moving from the right to the left corresponds to removing randomly selected instances from the training set. (See Section 2 for a detailed description of how the graphs were generated.)

In all five graphs in Figure 1, accuracy peaks with small numbers of training instances, thereafter remaining almost constant. Surprisingly, tree size continues to grow nearly linearly in three of the graphs. Growth continued despite two important facts: (1) accuracy has ceased to increase; and (2) C4.5 is pruning the trees to avoid overfitting. The graphs clearly show that overfitting is occurring, and it gets worse as the size of the training set increases. For example, with EBP, accuracy peaks after only 25% of the available training instances are seen. The tree at that point contains 22 nodes. When 100% of the available training instances are used in tree construction, the resulting tree contains 64 nodes. Despite a 3-fold increase in size over the tree built with 25% of the data, the accuracies of the two trees

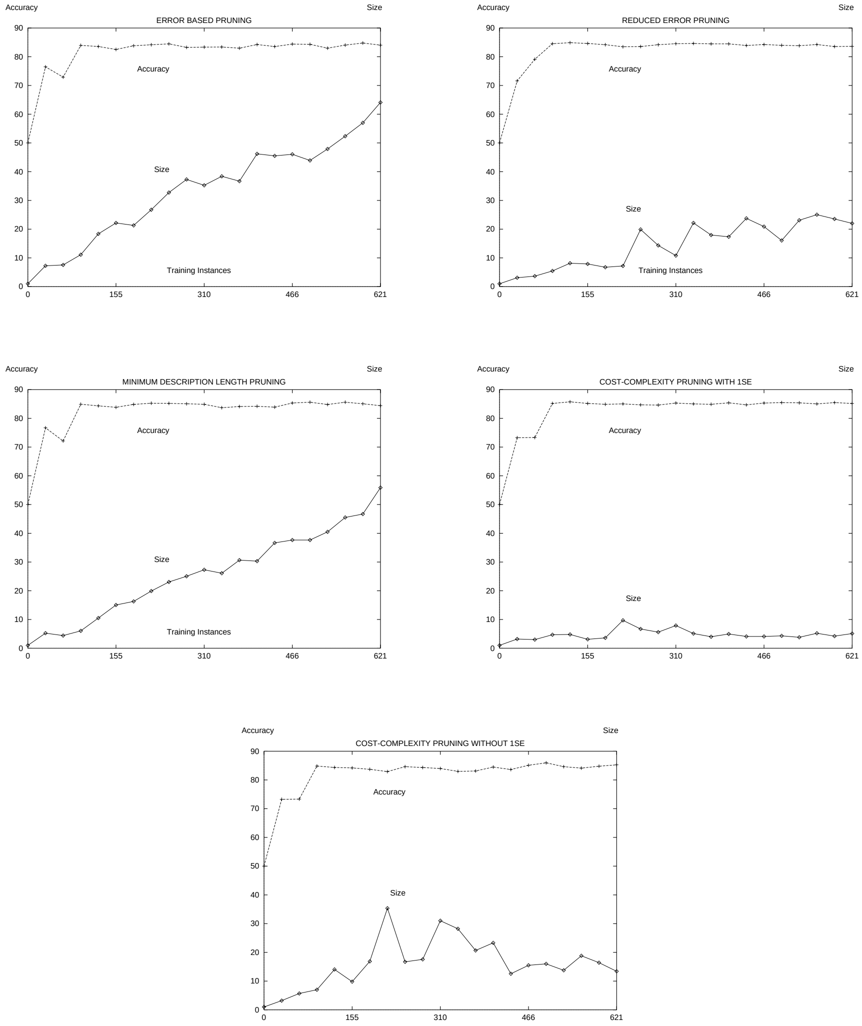

Figure 1: Plots of tree size and accuracy as a function of training set size for the **australian** dataset. All trees were generated by C4.5, and each plot corresponds to a different pruning mechanism: error-based, reduced error, minimum description length, cost complexity with the 1SE rule, and cost complexity without the 1SE rule (0SE).

are statistically indistinguishable.

One clear implication of the strong relationship between training set size and tree size is that almost any scheme for removing training instances prior to tree construction will, on this dataset, yield smaller trees with accuracies roughly equivalent to that obtainable from the full training set. Also, the size of the resulting tree will depend strongly on the fraction of instances that are discarded. The reason is that removing any instances, even randomly selected instances (which corresponds to moving from the right-hand side of the graphs in Figure 1 to the left), has just that effect, and the magnitude of the effect increases with the number of training instances that are discarded. Therefore, it seems likely that at least part of the reduction in tree size observed by those studies cited earlier is attributable to the nearly linear relationship between training set size and tree size as exhibited in Figure 1. Manipulating training set size will have an impact on tree size, regardless of the method used to rule training instances in or out. This suggests that random data reduction is a baseline against which more sophisticated data reduction techniques should be compared. The magnitude of the reduction in tree size that such techniques obtain by discarding training instances should be decomposed into two components: that which is due to reduction of training set size (i.e. the reduction that would result from removing the same number of randomly selected instances), and the remainder, which is directly attributable to how the method selects instances to remove.

The rest of the paper is organized as follows. Section 2 explores the relationship between tree size and accuracy and training set size for 5 different pruning methods on 19 datasets taken from the UC Irvine repository. Section 3 performs the decomposition mentioned above for one data reduction technique, and shows that a substantial percentage of the gains achieved by that technique are due to reduction of training set size. Finally, Section 5 concludes with a discussion of additional implications of this work and future directions.

2 Empirical Results

The experiments in this section test the hypothesis that, under a broad range of circumstances, there is a nearly linear relationship between training set size and tree size, even after accuracy has ceased to increase. The experiments generate plots of tree size and accuracy as a function of training set size for a given dataset and pruning algorithm, find the training set size at which accuracy ceases to increase, and run a linear regression on the points in the tree size curve to the right of that training set size. In general, additional tree structure is welcome as long as it improves classification accuracy, and it is unwelcome otherwise. Ideally, there will be no correlation between tree size and training set size once classification accuracy peaks. The linear regression of tree size on training set size indicates the probability, p, of making an error in rejecting the null hypothesis that there is no such correlation (that the slope of the regression line is zero), and the amount of variance in tree size accounted for by training set size, r^2. When p is significant and r^2 is high, changes in training set size have strong and predictable effects on tree size.

The relationship between training set size and tree size was explored with 5 pruning methods and 19 datasets taken from the UC Irvine repository.[1] The pruning methods are error-based (EBP – the C4.5 default) (Quinlan 1993), reduced error (REP) (Quinlan 1987), minimum description length (MDL) (Quinlan & Rivest 1989), cost-complexity with the 1SE rule (CCP1SE) (Breiman *et al.* 1984), and cost-complexity without the 1SE rule (CCP0SE). The majority of extant pruning methods seem to take one of four general approaches: deflating accuracy estimates based on the training set (e.g. EBP); pruning based on accuracy estimates from a pruning set (e.g. REP); creating a set of pruned trees based on different values of a parameter and then selecting the appropriate parameter value using a pruning set or cross-validation (e.g. CCP1SE and CCP0SE); and managing the tradeoff between accuracy and complexity (e.g. MDL). The pruning methods used in this paper were selected to be representative of these four approaches. CCP0SE was included to determine the impact of the 1SE rule in cost-complexity pruning.

The plots of tree size and accuracy as a function of training set size were generated for each combination of dataset and pruning algorithm as follows. Typically, k-fold cross-validation is used to obtain estimates of the true performance of decision tree algorithms. A dataset, D, with n instances is divided into k disjoint sets, D_i, each containing n/k instances. Then for

[1] The datasets are the same ones used in (John 1995) with two exceptions. The `crx` dataset was omitted because it is roughly the same as the `australian` dataset, and the `horse-colic` dataset was omitted because it was unclear which attribute was used as the class label. Note that the `vote1` dataset was created by removing the `physician-fee-freeze` attribute from the `vote` dataset.

$1 \leq i \leq k$, a tree is built on the instances in $D - D_i$ and tested on the instances in D_i, and the results are averaged over all k folds (Cohen 1995). That procedure was augmented for this paper by building trees on subsets of $D - D_i$ of various sizes, and testing them on D_i. Specifically, 20 subsets were created by retaining from 5% to 100% of the instances in $D - D_i$ in increments of 5%; standard k-fold cross-validation corresponds to the case in which 100% of the instances in $D - D_i$ are retained. The order of the instances in D was permuted prior to creating the $k = 10$ folds, and the instances to be retained were gathered sequentially starting with the first instance in $D - D_i$ for each level of data reduction. In this way, 10-fold cross-validated estimates of tree size and accuracy as a function of training set size were obtained. (Cohen calls this incremental cross-validation.) This procedure was performed twice for each combination of dataset and pruning method, generating complete size and accuracy curves for two different permutations of the data, and the results were averaged. The goal was to reduce the inherent variability of cross-validated estimates of size and accuracy. Note that the same divisions of a given dataset were used for all of the pruning methods. With 19 datasets, 5 pruning methods, 20 levels of training set size, and 2 runs of 10-fold cross-validation at each level of training set size, the results reported in this paper involved running C4.5 38,000 times.

For each plot generated according to the procedure outlined above, the training set size at which accuracy ceased to grow was found by scanning the accuracy curve from left to right, stopping when the mean of three adjacent accuracy estimates was no more than 1% less than the accuracy of the tree based on all available training data (the right-most point on the accuracy curve, which data reduction techniques typically use as the standard for comparison). Averaging three adjacent accuracies makes the stopping criterion robust against random variations in the accuracy curve.[2] Bounding the absolute change in accuracy from below by 1% ensures that any reduction in tree size costs very little in terms of accuracy. Then, as described above, a linear regression of tree size on training set size was performed on the points in the tree size curve to the right of the training set size at which accuracy ceased to grow.

[2] We did not use the mean of the final three points on the accuracy curve minus 1% as the accuracy threshold because those points represent different training set sizes, and their mean is therefore not an estimate (robust or otherwise) of the accuracy of trees built on all available training instances.

The results for EBP are summarized in Table 1. We report the percentage of available training instances at which accuracy ceased to grow (% Kept), results of the linear regression of tree size on training set size (p and r^2), the percentage decrease in tree size (Δ size) and the absolute difference in accuracy (Δ accuracy) between the tree built from all available training instances and the tree built from the number of instances at which accuracy ceased to grow. Given tree T_f built from the full training set and tree T_r built from the reduced training set, Δ size $= 100 * (size(T_f) - size(T_r))/size(T_f)$, and Δ accuracy $= accuracy(T_f) - accuracy(T_r)$. Linear regression requires at least 3 data points, so no results are reported for a dataset if accuracy continued to grow with training set sizes larger than 90% of the available data. Also, if there is no relationship between tree size and training set size (i.e. if $p > 0.10$), then p is listed as *ns* (not significant) and no other results are given for that dataset. The final row of the table gives the number of datasets for which accuracy peaked prior to seeing 100% of the available training instances, the number of datasets for which the relationship between tree size and training set size is significant, and the means of r^2, Δ size and Δ accuracy for those datasets with significant p values.

Summary information for all of the pruning methods (i.e. the information contained in the last row of Table 1) in given in Table 2.

Consider Table 1, which shows the results for EBP. Accuracy peaked prior to seeing 100% of the available training instances for 16 of the 19 datasets. Every one of those 16 datasets exhibited a significant relationship between tree size and training set size beyond the point at which accuracy stopped growing, and 12 of them were highly significant (at the 0.001 level). In spite of the fact that accuracy remains basically constant, tree size continues to grow as training set size does (the slope of the regression line is positive in all cases). The most remarkable feature of the table is the r^2 column. Recall that $100{*}r^2$ is the percentage of variance in tree size accounted for by training set size. Across 16 datasets, the average r^2 is 0.90. This result is interesting for two reasons. First, it says that training set size has an extremely strong and predictable effect on tree size. Increasing training set size invariably leads to larger trees; decreasing training set size invariably leads to smaller trees. Second, this effect is robust over a large group of datasets with widely varying characteristics. Regardless of the default accuracy, the number and types of attributes, the presence or absence of class and attribute noise, and differences in a number of other features along which the datasets vary, EBP

Dataset	% Kept	p	r^2	Δ size	Δ accuracy
australian	25	0.001	0.93	65.44	1.50
breast-cancer	100				
breast-cancer-wisc	50	0.001	0.90	32.72	0.36
kr-vs-kp	45	0.001	0.77	19.18	0.58
cleveland	40	0.001	0.92	39.45	-0.81
diabetes	30	0.001	0.99	71.38	-1.92
german	50	0.001	0.98	47.86	-1.53
glass	45	0.001	0.99	50.76	-0.22
heart	100				
hepatitis	40	0.001	0.84	38.93	-1.06
hypothyroid	20	0.001	0.64	36.00	0.45
iris	85	0.061	0.88	16.48	0.31
labor-neg	100				
lymphography	85	0.061	0.88	16.70	-0.60
segment	75	0.001	0.94	16.71	0.61
sick-euthyroid	20	0.001	0.88	55.87	0.43
tic-tac-toe	85	0.017	0.97	8.04	-0.67
vote	20	0.001	0.85	32.38	0.45
vote1	20	0.001	0.97	64.81	-0.11
	16	16	0.90	38.29	-0.14

Table 1: The effects of random data reduction on C4.5 with error-based pruning (C4.5's default pruning method).

does not appropriately limit tree size as training set size increases.

The Δ size column of Table 1 shows the percent reduction in size from trees built on all available training instances to trees built on the number of instances in the % Kept column. The Δ accuracy column shows the absolute difference in accuracy between those same trees. In Table 1 the mean reduction in tree size for the 16 datasets with significant p values is 38.29%, and the mean difference in absolute accuracy is -0.14%. By reducing training set sizes through the removal of randomly selected instances, it is possible, on average, to obtain trees that are 38.29% smaller, with a sacrifice in accuracy of less than two tenths of one percent. Note that accuracy was higher with reduced training sets in 8 cases, and it was lower in 8 cases.

The results for REP and MDL (Table 2) are qualitatively the same as those for EBP. For REP, 17 datasets show a significant relationship between tree

Pruning Method	% Kept < 100	$p < 0.1$	Mean r^2	Mean Δ size	Mean Δ accuracy
EBP	16	16	0.90	38.29	-0.14
REP	17	17	0.75	39.32	-0.32
MDL	18	17	0.88	44.03	-0.37
CCP1SE	19	10	0.62	30.11	-0.06
CCP0SE	17	11	0.58	47.84	-0.06

Table 2: Summary of the effects of random data reduction for all of the pruning methods.

size and training set size (12 at the 0.001 level) and the mean r^2 is 0.75. The average reduction in tree size obtainable via random data reduction is 39.32% with an average loss in accuracy of less than four tenths of one percent. Accuracy was higher with reduced training sets in 12 of the 17 cases. For MDL, 17 datasets had significant p values (14 at the 0.001 level), the average r^2 was 0.88, and trees based on reduced training sets were on average 44.03% smaller and less than four tenths of one percent less accurate. Note that for one dataset, **hypothyroid**, there is no significant relationship between tree size and training set size past the point at which accuracy stopped growing. In this one case, MDL appropriately limits tree size by not adding structure to the tree unless a concomitant increase in classification accuracy occurs.

The results for CCP1SE and CCP0SE (Table 2) indicate that they appropriately limit tree growth much more frequently than the previous three pruning methods. Consider CCP1SE. Accuracy peaked for all 19 datasets prior to seeing 100% of the available training instances. However, only about half of the time (10 out of 19 datasets) was there a significant relationship between tree size and training set size after accuracy stopped growing. CCP1SE appropriately limits tree growth for 9 datasets, whereas EBP and REP never did so, and MDL did so once. For the 10 datasets that exhibited significant relationships between tree size and training set size, random data reduction still leads to substantially smaller trees (30.11% on average) with little loss in accuracy (less than one tenth of one percent on average). The results for CCP0SE are qualitatively the same.

3 A Case Study

The results of the previous section show that there is often a strong relationship between tree size and training set size, even when there is no such relationship between accuracy and training set size. Furthermore, reducing tree size by randomly removing training instances costs little or nothing in terms of accuracy over some (often large) range of training set sizes. This suggests that all data reduction techniques will see some decrease in tree size simply because they are reducing the size of the training set. Clearly, one would like to know how much of the decrease in tree size obtained by a given data reduction method is due to how the method selects instances to remove, and how much of the decrease is due to the fact that the method is reducing the size of the training set. In this section, we investigate that question for one of the data reduction methods mentioned earlier, John's ROBUST-C4.5 (RC4.5) (John 1995).

The idea behind RC4.5 is that when a pruning algorithm turns a test node into a leaf, it is in effect making a local decision to ignore those instances that don't belong to the majority class. John reasoned that if those instances are not informative locally, at the node where the decision to prune is made, they may also be uninformative globally, higher up in the tree. This insight is incorporated into the RC4.5 algorithm by removing training instances that the pruned tree misclassifies, and rebuilding the tree on the new, reduced training set. This procedure is repeated, removing additional instances and rebuilding the tree, until a tree is created the correctly classifies all of the remaining training instances. The result over a large number of datasets (using C4.5 with EBP to build and prune trees) is trees that are much smaller than those built by the standard C4.5 algorithm, but that have roughly equivalent accuracy.

To determine how much of RC4.5's effect on tree size for a given dataset is due to reduction of training set size, we need to know four items of information: the size of the tree that C4.5 builds on the entire dataset (C4.5 Size); the size of the tree that C4.5 builds on the reduced dataset generated by RC4.5 (RC4.5 Size); the percentage of training instances retained by RC4.5 (% Kept); and the size of the tree that C4.5 builds when the same percentage of randomly selected training instances are retained (RDR Size). The percentage of RC4.5's effect on tree which is due to reduction in training set size can then be computed as $100 * (\text{C4.5 Size} - \text{RDR Size})/(\text{C4.5 Size} - \text{RC4.5 Size})$.

To obtain estimates of C4.5 Size, RC4.5 Size and % Kept for a given dataset, we generated 10-fold cross-validated estimates of those quantities on 20 different permutations of the data, and averaged the results over the 20 permutations. The goal of averaging the results over multiple runs of cross-validation was to reduce the variance in our estimates. Given an estimate of the number of training instances that RC4.5 can be expected to discard for a dataset, RDR Size was estimated via 10-fold cross-validation on 20 new permutations of the data where each of the 10 training sets in each run of cross-validation were reduced by randomly discarding the same number of instances that RC4.5 would discard.

					% of RC4.5 Effect
Dataset	C4.5 Size	RC4.5 Size	% Kept	RDR Size	Due to RDR
australian	61.58	48.48	92.19	58.89	20.53
breast-cancer-wisc	20.25	18.25	97.48	20.08	8.5
cleveland	44.61	35.13	88.58	41.70	30.70
diabetes	124.96	65.99	83.11	107.24	30.05
german	157.37	108.65	84.01	131.11	53.90
glass	50.21	41.33	89.34	46.02	47.18
heart	44.26	36.28	90.68	41.31	36.97
hepatitis	14.02	11.5	90.32	14.27	-9.92
lymphography	26.10	23.98	90.14	23.62	116.98
segment	83.05	78.47	98.48	82.48	12.45
tic-tac-toe	131.55	119.67	89.44	119.35	102.69
vote1	21.96	18.32	93.17	20.14	50.00

Table 3: A decomposition of the effect of RC4.5 on tree size into components attributable to reduction in training set size and to the method for choosing which training instances to discard.

Table 3 shows the results for datasets for which RC4.5 achieved a 5% or greater reduction in tree size over C4.5. On the **hepatitis** dataset, random data reduction actually results in a larger tree than the one that C4.5 builds on the full dataset. Reduction of training set size accounts for only about 10% of RC4.5's effect on two of the datasets (**breast-cancer-wisc** and **segment**), and it accounts for 100% of RC4.5's effect on two other datasets (**lymphography** and **tic-tac-toe**). On average, 41.67% of the decrease in tree size that RC4.5 obtains is attributable to the fact that it is simply reducing the size of the training set.

What do these results mean? First, it is clear that tree sizes obtained through random data reduction should serve as a baseline against which other data reduction techniques measure their success, much as default accuracy or Holte's one-rules serve as a baseline for classification accuracy (Holte 1993). If a data reduction technique improves accuracy, or obtains smaller trees relative to trees built by eliminating a comparable number of randomly selected instances, then our confidence in that technique's ability to identify "bad" instances is boosted. Second, these results by themselves do not shed any additional light on the merits of RC4.5. We know that for the 12 datasets listed in Table 3, 42% of RC4.5's effect is due to reduction of training set size, and 58% is due to RC4.5's method of selecting instances to remove. Clearly, substantial reductions in tree size are directly attributable to the method. RC4.5's approach to selecting training instances is highly effective in some cases (e.g. **segment**), and highly ineffective in others (e.g. **tic-tac-toe**). Note that the algorithm's lack of success with the **tic-tac-toe** dataset is not unexpected because that dataset is noise-free, and anything removed as an "outlier" is probably an infrequent pattern rather than an anomalous instance. We cannot judge whether decreases in tree size achieved by RC4.5 after accounting for the effect of reducing training set size are better or worse than those achieved by other data reduction techniques until those other techniques undergo experiments similar to the one reported in this section.

4 Bias in Reduced Error Pruning

Why would any of the five pruning methods explored earlier exhibit a pathological relationship between tree size and training set size? This section examines one form of bias inherent in several pruning methods that explains part of that relationship. This bias is discussed in the context of REP for concreteness and clarity, and an empirical demonstration of its effects is presented.

Recall that REP builds a tree with a set of growing instances, and then prunes the tree bottom-up with a disjoint set of pruning instances. The number of classification errors that a subtree rooted at node N makes on the pruning set, $E_T(N)$, is compared to the number of errors made when the subtree is collapsed to a leaf, $E_L(N)$. If $E_T(N) \geq E_L(N)$, then N is turned into a leaf.

Note that $E_L(N)$ is independent of the structure of the subtree rooted at N. To compute $E_L(N)$, all of the instances in the pruning set that match the attribute tests on the path from the root of the tree to N are treated as a set. The number of instances in this set that do not belong to the majority class of the set is the number of errors that the subtree would make as a leaf. For a given pruning set, $E_L(N)$ depends only on the structure of the tree above N, and therefore does not depend on how pruning set instances are partitioned by additional tests below N. As a consequence, $E_L(N)$ remains constant as the structure beneath N changes due to the effects of bottom-up pruning.

In contrast to E_L, $E_T(N)$ is highly dependent on the structure of the subtree rooted at N. $E_T(N)$ is defined to be the number of errors made by that subtree on the pruning set, and its value can change as pruning takes place beneath N. Consider a subtree rooted at N', where N' is a descendant of N. If $E_T(N') < E_L(N')$ then N' is not pruned, and because the structure beneath N remains unchanged, $E_T(N)$ also remains unchanged. The alternative is that $E_T(N') \geq E_L(N')$, in which case N' is turned into a leaf. This structural change either causes $E_T(N)$ to remain unchanged (when $E_T(N') = E_L(N')$) or to decrease (when $E_T(N') > E_L(N')$).

E_L and E_T are estimates of the accuracy of a subtree, as a leaf and as a tree respectively, on the population of instances from which the pruning set was drawn. Each time pruning occurs beneath N, $E_L(N)$ remains invariant and $E_T(N)$ usually decreases. This systematic deflation of E_T, a statistical bias inherent in REP, produces two effects: (1) pruning beneath N increases the probability that $E_T(N) < E_L(N)$ and that N will therefore not be pruned; (2) E_T for the final pruned tree tends to be an overestimate. These effects should be larger for large unpruned trees, because they afford many opportunities to prune and to deflate E_T. These effects should also be larger for small pruning sets because they increase the variance in estimates of E_L and E_T. Highly variable estimates make it more likely that, by random chance, $E_T(N) \geq E_L(N)$ and the subtree rooted at N will be pruned, thereby boosting E_T for all parents of N. Note that it is also more likely that, by random chance, $E_T(N) < E_L(N)$, resulting in no change in E_T for the parents of N and the retention of the structure beneath N. In either case, the net result is larger trees, either from the explicit retention of structure or increases in the bias of E_T which often leads to the retention of structure higher in the tree.

We empirically explored the effects of the bias described above in a series of experiments with artificial data. All datasets contained instances with 30 binary attributes with uniformly distributed values. The class label for each instance was the value of the first attribute, and the class label was complemented to simulate noise with probability $p \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Trees were built by C4.5 and pruned with REP, and their accuracy on the pruning set was measured. The size of the growing set was varied from 100 to 1000 instances in increments of 100, as was the size of the pruning set. This led to 100 distinct experimental conditions (a fully factorial experiment). For each experimental condition, 15 replicates were performed. As might be expected from the experiments in Section 2, training set size is an almost perfect proxy for the size of the unpruned tree. The correlation between the two for all levels of noise exceeded 0.99.

With no noise in the data ($p = 0.0$), all trees contained 3 nodes (a test on the value of the first attribute and two leaves) and were 100% accurate on the pruning set, regardless of the size of the growing and pruning sets. However, for all other noise levels, a two-way analysis of variance (ANOVA) revealed main effects of growing set size and pruning set size, and an interaction effect, all significant at the 0.01 level. The theoretical limit on the accuracy of the trees is $1 - p$, which was always attained with the smallest growing set (100 instances) and the largest pruning set (1000 instances). To the extent that accuracy exceeds $1 - p$, E_T is a biased estimate of the accuracy of the tree on the population. Inspection of the group means generated by the ANOVA revealed that increasing the size of the growing set increased bias, as did decreasing the size of the pruning set (the main effects found by the ANOVA). Said differently, larger trees and smaller pruning sets each lead to increased bias. Bias peaked (i.e. the overestimate of E_T was the greatest) with the largest training set size and the smallest pruning set size (the interaction effect found by the ANOVA).

5 Discussion

Experiments with 5 pruning methods and 19 datasets demonstrated that tree size is strongly dependent on training set size. As the percentage of available instances used to build the tree is increased from 0% to 100%, accuracy often peaks quickly. Despite the fact that adding more training instances has little effect on accuracy, doing so has a large effect on tree size. Trees built with 100% percent of the available training instances are often much larger, and no more accurate, than trees built on a small subset of the training instances. Given the strong relationship between tree size and training set size, any technique that removes training instances prior to tree construction could result in smaller trees just because it is reducing the size of the training set.

The realization that small numbers of training instances suffice to build small, accurate trees, in addition to yielding a useful tree-simplification tool, frees data previously used in tree construction for other purposes. For example, many pruning techniques divide the training set into two disjoint subsets, one for building the tree and another for pruning (Quinlan 1987; Cestnik & Bratko 1991; Mingers 1989). Larger pruning sets result in better estimates of classification accuracy and, therefore, more effective pruning. Random data reduction simultaneously produces smaller trees and makes mores data available for pruning.

Random data reduction can also serve as a method for evaluating new pruning techniques. Continued growth in tree size with no associated increase in accuracy points to a problem with overfitting, and experiments such as the one described in Section 2 can be used to determine the extent of the problem for a given pruning method. In addition, random data reduction can be used to estimate the size of the "right" tree. One can assess whether a pruning method results in trees of appropriate size on artificial datasets by comparing the trees to tree-based representations of the function used to compute the class label. However, that approach is not possible for real-world data, where the function used to assign class labels is unknown (thus the need to construct decision trees). Random data reduction can be used to find the smallest tree that results in accuracy equivalent to that possible with the full dataset, yielding an estimate of the size of the "right" tree.

Future research will include additional investigation of why three of the pruning methods tested in this paper do not avoid overfitting as training set size increases. One of the authors has identified multiple testing in tree construction and pruning as one source of problems, and has implemented a promising solution (Removed for blind review. 1997). Also, decision trees are but one type of model, and we intend to investigate the extent to which other model construction algorithms fall victim to a pathological relationship between model complexity and the amount of data used to build the model.

References

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

Brodley, C. E., and Friedl, M. A. 1996. Identifying and eliminating mislabeled training instances. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence.*

Cestnik, B., and Bratko, I. 1991. On estimating probabilities in tree pruning. In *Proceedings of the Fifth European Working Session on Learning*, 138–150.

Cohen, P. R. 1995. *Empirical Methods for Artificial Intelligence.* The MIT Press.

Holte, R. C. 1993. Very simple classification rules perform well on most commonly used dataset. *Machine Learning* 11:63–90.

Removed for blind review. 1997. Adjusting for multiple testing in decision tree pruning. In *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics.*

John, G. H. 1995. Robust decision trees: Removing outliers from databases. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining.*

Mingers, J. 1989. An empirical comparison of pruning methods for decision tree induction. *Machine Learning* 4:227–243.

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.