Machine Learning, 10, 153–178 (1993)
© 1993 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Overfitting Avoidance as Bias

CULLEN SCHAFFER SCHAFFER@MARNA.HUNTER.CUNY.EDU
Department of Computer Science, CUNY/Hunter College, 695 Park Avenue, New York, NY 10021

Editor: Bruce Porter

Abstract. Strategies for increasing predictive accuracy through selective pruning have been widely adopted by researchers in decision tree induction. It is easy to get the impression from research reports that there are statistical reasons for believing that these *overfitting avoidance* strategies do increase accuracy and that, as a research community, we are making progress toward developing powerful, general methods for guarding against overfitting in inducing decision trees. In fact, any overfitting avoidance strategy amounts to a form of bias and, as such, may degrade performance instead of improving it. If pruning methods have often proven successful in empirical tests, this is due, not to the methods, but to the choice of test problems. As examples in this article illustrate, overfitting avoidance strategies are not better or worse, but only more or less appropriate to specific application domains. We are not—and cannot be—making progress toward methods both powerful and general.

Keywords. Overfitting avoidance, decision tree pruning, inductive bias

1. Introduction

It has been widely noted by researchers studying decision tree induction that performance on training data may be a misleading indication of true predictive accuracy. A complex tree that achieves high accuracy on a training set will often fare worse on fresh data than a simple tree that performed less spectacularly. In this case, researchers say that the complex tree has "overfit" the training data, reflecting not only true underlying relationships, but also patterns arising purely by chance.

Since overfitting in this sense decreases predictive accuracy, a great deal of effort has been expended in developing overfitting avoidance methods for tree induction, generally in the form of pruning strategies. These methods have been reported widely (Breiman et al., 1984; Cestnik & Bratko, 1991; Mingers, 1987; Niblett & Bratko, 1986; Quinlan, 1987; Quinlan & Rivest, 1989); they have been compared empirically (Mingers, 1989); and few researchers would now undertake decision tree induction without relying on some form of overfitting avoidance.

In fact, however, if overfitting avoidance methods have improved the predictive accuracy of induced decision trees it is not because these methods are inherently beneficial. Any overfitting avoidance strategy amounts to a kind of bias and biases are only as helpful as they are appropriate to a domain of application. In particular, as this article is meant to show, the well-documented success of decision tree pruning strategies in published studies is due entirely to the problems on which researchers have chosen to test them. Alternative choices would support the conclusion that it is best to ignore overfitting—even in the presence of noisy data—and simply to choose the tree, of those considered, that best fits the training set.

This is not to deny, of course, that overfitting avoidance methods have often worked well in practice. Apparently, the biases they entail *are* frequently appropriate to the domains from which decision tree induction problems have been drawn. The important point for researchers is to credit observed successes to the appropriate application of bias rather than to construe them as evidence in favor of the bias itself.

Few researchers would suggest that the success of pruning techniques on problems from a number of domains proves by itself that the techniques have a domain-independent value, but the combined weight of empirical successes, intuitive arguments and theoretical references *has* apparently persuaded many of the inherent value of avoiding overfitting, especially in the presence of noise. It is this presumption of *inherent* value that this article is intended to dispel in demonstrating that overfitting avoidance is a form of bias.

To draw an analogy, suppose we are given time to experiment with an unfair coin and then asked to predict whether it will show heads or tails on the next flip. A basic strategy consists of flipping the coin a number of times and predicting for the next flip whichever side of the coin has appeared most often. Consider two variations of this strategy. The first calls for a prediction of heads if heads is flipped in at least a third of the preliminary trials. This is clearly an example of what we mean by the word bias and, just as clearly, it is neither an inherently good nor an inherently bad idea. If we often apply the revised strategy to coins with a high probability of turning up heads, it will outperform the original one; if not, the original strategy will be superior. A bias is as good as it is appropriate.

A second strategy is identical to the basic one except that it doubles the number of preliminary flips. Here, by contrast, a definite statistical improvement has been made. Regardless of the kind of unfair coin we are given, straightforward probabilistic arguments prove that the revised strategy is more likely to make a correct prediction than the original one. In this sense, it is inherently beneficial.

In light of this analogy, we may say that many researchers have concluded that overfitting avoidance is a revision of the second kind; the purpose of this article is to show it is of the first.

A strong indication of the prevalence of the conception of overfitting avoidance as a statistical improvement is the fact that research to date has focused on inventing better pruning techniques or improving old ones. As a form of bias, however, any overfitting avoidance method is only conditionally beneficial and the adjectives "better" and "improved" are meaningless unless these conditions are made explicit. This does not imply, of course, that we ought to abandon accepted pruning techniques. Rather, first, we need to understand what it is about the problems on which they have been demonstrated that accounts for their successes, so that we may delineate a domain of applicability for each technique; and second, since it is fundamentally impossible, we need to abandon the project of attacking decision tree induction in full generality—and with it the subproject of inventing "good" pruning techniques—and concentrate instead on approaches designed expressly for specified conditions that we have observed to hold in important practical domains.

2. Overview

The main body of this article demonstrates, through a series of experiments, that the effect of a particular, widely respected overfitting avoidance technique depends on the problem-generating environment in which it is applied. It illustrates through these same experiments the fact that *all* well-known overfitting avoidance techniques are, likewise, a form of bias rather than a statistical improvement in inducing decision trees. The experiments also suggest some of the domain characteristics that may make overfitting avoidance more or less appropriate and, in particular, refute the common notion that classification noise justifies overfitting avoidance.

Following discussion of the experiments, two meta-points are addressed. First, theoretical, intuitive and empirical arguments in favor of overfitting avoidance are reviewed to see how they can be reconciled with the results presented. Second, the practical significance of the argument embodied in this article is considered.

3. Methodology

3.1. Two strategies for comparison

The experiments reported in the following sections compare two tree induction strategies that differ only in their approach to the problem of overfitting. In an initial phase common to both strategies, a large tree is constructed through recursive splitting. A *naive* strategy selects this tree without modification for use in prediction. A *sophisticated* strategy iteratively prunes to obtain a sequence of trees of increasing simplicity and decreasing accuracy on the training set. It then uses a cross-validation procedure in an attempt to decide which tree in the sequence will yield the best predictive accuracy. The conclusion may be that the original, large tree is best, in which case the naive and sophisticated strategies coincide. If not, however, we may compare the accuracy of the chosen trees to see which is superior.

The sophisticated strategy is based closely on the CART procedure described in detail in Breiman et al. (1984).[1] Trees are built using the Gini splitting rule that measures the impurity of a node as $(n_1/n)(1 - n_1/n)$, where n_1 is the number of class-one instances at the node and n is the total number of instances at the node.[2] Splitting continues as long as impurity can be decreased.

The resulting large tree is pruned iteratively, each iteration eliminating the branch that currently contributes the least gain in accuracy on the training set per leaf node added. This yields a sequence of trees and a *complexity cost* in accuracy gain per leaf node at which each would be acceptable.

To choose an appropriate complexity cost, a tenfold cross-validation strategy is applied, as recommended in Breiman et al. (1984). The procedure just described is repeated ten times, each time using 9/10 of the training cases to build a sequence of trees and the remaining 1/10 to evaluate trees in the sequence and obtain unbiased information about performance at each level of complexity cost. Information from the ten trials is then pooled and the apparent optimal complexity cost is used to select one of the sequence of trees derived from the full training set.

The main difference between this strategy and the one described by Breiman et al. is that the latter, understanding that information on performance at various levels of complexity cost is subject to chance variation, choose a tree based on a complexity cost that yields cross-validated performance within one standard deviation of the apparent optimum. This wrinkle is irrelevant here, since it is intended to promote intelligibility rather than to increase predictive accuracy. In any case, the "one standard error" rule amounts to an additional bias toward simplicity and would only strengthen the results presented.

The CART approach was chosen as representative of the most sophisticated overfitting avoidance strategies that research in decision tree induction has produced (Mingers, 1989). It should be clear from the argument presented in this article that results would be qualitatively similar if another of the well-known algorithms had been employed in comparisons. In particular, it will be demonstrated in Section 5 that one of the key results presented below holds for a broad class of pruning strategies, including all of those best known to machine learning researchers.

3.2. Experiments, reporting and interpretation

In the experiments reported here, five boolean attribute variables, A_1 through A_5, take on the values **T** and **F** independently and with equal probability. A boolean class variable C takes on a value dependent in some way on the attributes, though noise in one form or other always complicates this dependency.

An experiment consists of 25 trials in which the naive and sophisticated strategies are compared. In each trial, a training set of 50 cases is generated and used by both strategies to induce a decision tree. In most of the experiments, the exact, theoretical accuracy of these trees—in predicting the value of the class variable in fresh cases generated in the same way as the training data—is calculated and used in comparisons. Where this calculation would be forbiddingly complex, the fact is noted and trees are tested on 10,000 fresh cases to determine their accuracies empirically.

Trials in which the naive and sophisticated strategies choose identical trees or trees with the same predictive accuracy tell us nothing about which strategy is superior. Hence, reporting focuses on the remaining, *discrepant* trials. For these trials, the average accuracy of trees induced by each strategy is reported along with the number of trials in which each produced the superior tree.

When one strategy has produced superior trees in a large majority of the discrepant trials, the statistical significance of this majority is also reported. If there are 17 discrepant trials in an experiment and the sophisticated strategy produces superior trees in 15 of them, it seems unlikely that the strategies are, in fact, evenly matched. To see how unlikely, we calculate the probability of the sophisticated strategy proving superior in as many as 15 of 17 trials if each strategy had an equal chance of producing a superior tree. This probability is .0011749; hence we report the significance of the majority as .998825.[3]

Although reporting focuses on discrepant cases, it will sometimes be of interest to note the accuracies of trees produced by the naive and sophisticated strategies averaged over all 25 trials of an experiment. To distinguish these from the averages over discrepant cases, they will be referred to as *gross* average accuracies.

A last methodological note concerns what some readers are bound to consider the disturbingly large size of the training set in these experiments relative to the size of the instance space. There are only 2^5, or 32, possible attribute vectors and 50 cases given for learning. In fact, while this relative abundance of training data was convenient for the purposes of this article, it does not at all determine the qualitative results presented below. With less data or more attributes, the superiority of the naive or sophisticated strategy would remain the same in each experiment reported here. This point is discussed at length in Section 5, where it is demonstrated analytically that training set size is irrelevant in one of the key experiments. For now, let it suffice to say that, given the amount of noise and the difficulty of the problems posed, the size of the training set is not excessive. The best simple evidence that it has been chosen appropriately is that both the naive and sophisticated strategies normally operate, in these experiments, far from the extremes of either null or perfect performance.

4. Experiments

4.1. Preliminary experiments

Two preliminary experiments demonstrate that the sophisticated strategy works as expected. In the first, the class variable C takes on the same value as A_1, but this value is logically complemented with probability .1 to simulate the effect of noise. In 22 of 25 trials, the sophisticated strategy produces a tree representing the true relationship between the attribute and class variables, $C = A_1$; the naive strategy recovers this relationship in only four trials. In seven of the 25 trials, the naive and sophisticated strategies choose trees with equal predictive accuracies. In the 18 discrepant trials, however, the tree selected by the sophisticated strategy is always superior. In these trials, the naive strategy chooses trees with an average accuracy of .824 compared with .9—the maximum achievable—for trees selected by the sophisticated strategy.

A second experiment differs only in the definition of the class variable. For these trials, $C = A_1 \vee (A_2 \wedge A_3)$. As before, C is complemented with probability .1 to simulate noise.

Again, the sophisticated strategy handily outperforms the naive one. It recovers the true underlying relationship in 12 of 25 trials while the naive strategy recovers the correct relationship just once. In eight of the 25 trials, the strategies choose trees of equal accuracy. In the 17 discrepant trials, the sophisticated strategy chooses a superior tree in 16 cases, achieving an average accuracy of .854 compared with .796 for trees selected by the naive strategy.

These results are much in line with many previous reports on overfitting avoidance techniques. Using cross-validation to determine how much of a tree should be pruned away eliminates branches that reflect spurious patterns and substantially increases predictive accuracy.

4.2. Overfitting avoidance may decrease predictive accuracy

The catch is that this effect depends critically on the choice of a model for data generation that is simple in the sense that it can be represented by a small tree. A third experiment illustrates this by defining $C = \text{Parity}(A_1A_2A_3A_4A_5)$ that is, $C = \mathbf{T}$ precisely when an odd number of the attributes take on the value **T**. All other features of the experiment are as in the first two.

In this case, the effect of overfitting avoidance is exactly the opposite of what we would hope. The naive and sophisticated strategies choose trees with different accuracies in only five of 25 trials, but in each of these discrepant cases the tree chosen by the naive strategy is superior. The superiority of the naive strategy is significant at the .97 level.

Breiman et al. (1984) have demonstrated their cross-validation pruning strategy on the now well-known digits recognition problem and shown that it outperforms a naive strategy that ignores overfitting. This result, however, depends in part on the fact that the optimal tree for the digits recognition problem is quite small relative to a full tree constructed with all the attributes.[4] As the experiment just reported indicates, if a different problem had been chosen for the demonstration, the effect of overfitting avoidance would have been the opposite.

It is easy to misinterpret the experiment of this section as indicating just that parity is a hard problem for decision tree methods or that it is not always a good idea to prune. Parity *is* a hard problem for decision tree methods—both the naive and sophisticated strategies perform more poorly in the third experiment than in the first two. Even for a hard problem, however, we would like to achieve the best performance possible and, in the case of the parity problem, the naive strategy leads to significantly better performance than the sophisticated one.

As for the second point, the experiment described in this section does not pit pruned trees against unpruned ones. It compares a strategy that never prunes against another that may or may not prune depending on indications in the data. The question is whether it pays to *consider* pruning as a guard against overfitting, whether the sophisticated strategy, in fact, makes more sophisticated use of the training data than the naive strategy in deciding which decision tree to induce. And the answer, taking the first three experiments together, is that the relative standing of the naive and sophisticated strategies is not absolute, but relative to the distribution of test problems. If problems like the parity example predominate, the naive strategy will appear superior; if problems based on simple relationships like $C = A_1$ and $C = A_1 \vee (A_2 \wedge A_3)$ predominate, the sophisticated strategy will prevail.

Returning to the coin flipping analogy, we may say that overfitting avoidance does not confer a definite benefit, like choosing to double the number of flips; it simply favors certain hypotheses, like lowering the threshold for predicting heads. The parity experiment illustrates the fact that overfitting avoidance is a bias in the sense that it is beneficial for certain problem distributions and detrimental for others. In the case of the parity problem, the bias is inappropriate and overfitting avoidance leads to a degradation in performance.

4.3. Classification noise may decrease the value of overfitting avoidance

Since overfitting avoidance is commonly considered a means of coping with noisy data, it might be objected that the level of noise in the parity experiment is too low to favor the sophisticated strategy. In fact, an error rate of .1 corresponds to a noise level of .2—if we define noise as the probability that a random boolean value will be substituted for the true one—and this is quite high by traditional standards.

More to the point, however, it is *not* the case that the sophisticated strategy performs better relative to the naive one in the parity problem as the rate of errors in the class variable

Table 1. Effect of classification noise: Parity problem.

	Naive Strategy Superior		Average Accuracy		
Error Rate	Cases	Significance	Naive Strategy	Sophisticated Strategy	Difference in Gross Accuracy
.05	3 of 3	.88	.762	.631	1.6 percent
.1	5 of 5	.97	.645	.595	1.0 percent
.15	8 of 8	.99	.648	.527	3.9 percent
.2	6 of 8	.86	.570	.519	1.6 percent
.25	17 of 17	.99	.575	.506	4.7 percent
.3	16 of 17	.99	.554	.514	2.7 percent

increases. Table 1 shows the results of a series of variations of the parity experiment in which the error rate takes on values from .05 to .3 (yielding noise levels of .1 to .6). The second row repeats results from the previous section: The strategies choose trees with different accuracies in five of 25 trials; the tree chosen by the naive strategy is superior in all five; and the average accuracies of trees chosen by the two strategies in these cases is .645 and .595.

The table shows clearly (a) that as classification noise increases, the accuracy of trees chosen by the two strategies differs more frequently and (b) that when the strategies disagree, the tree chosen by ignoring the possibility of overfitting is very likely to be superior, regardless of the level of noise. In a sense, then, an increase in classification noise here *decreases* the value of avoiding overfitting; the sophisticated strategy is more likely to produce an inferior tree when noise is high.

Of course, as the two average accuracy columns suggest, the accuracies of the strategies converge as the error rate approaches its maximum of .5. At this maximum, both will produce trees with an average accuracy of .5 and the difference between them will be nil. Nevertheless, over the very broad range of error rates considered here, the difference in the *gross* average accuracies consistently shows an advantage of at least one percentage point for the naive strategy.

4.4. Parity is not a special case

For $C = A_1$ or $C = A_1 \vee (A_2 \wedge A_3)$, the sophisticated strategy outperforms the naive one; for $C = \text{Parity}(A_1A_2A_3A_4A_5)$, the reverse is true. A fourth experiment is designed to show that the latter is not simply an anomaly. For this experiment, conditions are as in the previous three except for the definition of the class variable. In each trial, a relation between C and A_1 through A_5 is chosen at random over the space of all 2^{2^5} boolean functions of five variables.

Table 2 shows a pattern of results much in line with those presented for the parity problem in Table 1. As error rates increase, the naive and sophisticated strategies disagree more often; and at every level of noise the naive strategy proves superior. The accuracy of trees chosen by both strategies approaches chance as the error rate approaches .5, but over the wide range of error rates studied, the difference in gross accuracies consistently shows an advantage of at least one percentage point for the naive strategy.

Table 2. Effect of classification noise: Random boolean functions.

	Naive Strategy Superior		Average Accuracy		
Error Rate	Cases	Significance	Naive Strategy	Sophisticated Strategy	Difference in Gross Accuracy
.05	3 of 3	.88	.884	.725	1.9 percent
.1	4 of 5	.81	.790	.745	.9 percent
.15	10 of 11	.99	.707	.655	2.3 percent
.2	11 of 12	.99	.686	.616	3.4 percent
.25	12 of 13	.99	.609	.554	2.9 percent
.3	13 of 16	.99	.582	.549	2.1 percent

Thus, in a sense, over the space of possible true models, it is $C = A_1$ and $C = A_1 \vee (A_2 \wedge A_3)$ that are anomalous. Most boolean relations are like parity in that, when they underlie data generation as described, it is best to ignore overfitting in inducing decision trees.

This does not mean, of course, that the same is true of problems on which researchers have generally tested decision tree induction methods. Rather, the fact that published methods have often proven effective lends weight to the conclusion that the problems to which decision tree methods have been applied are a small and very special subclass.

4.5. The effect of overfitting avoidance is representation dependent

4.5.1. Random rerepresentations

The experiments of Sections 4.1 to 4.4 rely on a fixed representation scheme for the 32 possible elements of the instance space. Suppose a different scheme had been used. For example, if we define $B_1 = A_1$ and B_i = Parity($A_1 \ldots A_i$) for $i = 2, \ldots, 5$, then we may just as well use the vectors $B_1B_2B_3B_4B_5$ to represent the instance space elements.[5] The element heretofore identified by the $A_1A_2A_3A_4A_5$ vector **TTTFT** would now be identified by the $B_1B_2B_3B_4B_5$ vector **TFTTTF**.

Under this scheme, the relationship C = Parity($A_1A_2A_3A_4A_5$) may be represented succinctly as $C = B_5$. Thus, if decision tree induction for the parity problem were conducted using the B representation, the relationship to be discovered would be quite simple—precisely as simple, in fact, as the relationship $C = A_1$ of Section 4.1—and we could expect both the naive and sophisticated strategies to produce trees with much higher predictive accuracies than they did in Section 4.2 using the A representation. This is a straightforward example of the well-known principle that representation critically influences the power of induction strategies.

For present purposes, however, the key point of this example is not that the new representation increases the performance of both the naive and sophisticated strategies on the parity problem, but rather that it affects their relative standing. Under the A representation, the naive strategy performs best; under the B representation—since the rerepresented parity problem is exactly analogous to the $C = A_1$ problem of Section 4.1—the reverse is true. To rephrase the point, in the first case, it is best to ignore the possibility of overfitting; in the second, it pays to guard against it.

The effect of overfitting avoidance is not just problem-dependent—as previous sections have shown—but representation-dependent as well. Moreover, in the same sense that most boolean functions relating C to the attribute values A_1 through A_5 favor the naive strategy, most representation schemes may favor the naive strategy even when the sophisticated strategy would be superior for a fixed representation.

This may be illustrated through a variation of the first experiment. As in the original, we define $C = A_1$. For purposes of induction, however, a different representation is employed for each trial. Representations are chosen at random from the 32! possible five-bit schemes $B_1B_2B_3B_4B_5$ mapping one-to-one with the vectors $A_1A_2A_3A_4A_5$.[6] The error rate is held constant at .2.

In 18 of 25 trials of this kind, the naive and sophisticated strategies produce trees with identical predictive accuracies. In each of the seven discrepant cases, however, the tree chosen by the naive strategy is superior.[7] In these cases, the trees chosen by the naive strategy attain an average accuracy of .655 compared with .602 for trees chosen by the sophisticated strategy. Thus, while the sophisticated strategy is preferable under the A representation, random representations decisively favor the naive strategy.

Again, the same is evidently *not* true of the representations employed in many published demonstrations. The fact that overfitting avoidance is successful in these cases suggests that the representations employed are a distinguished minority among all those that might have been adopted.

4.5.2. Random rerepresentation vs. random functions

At first glance, it might seem that random rerepresentation of a specified boolean function is equivalent to choosing a boolean function at random and hence that the result just presented merely restates the result of Section 4.4. Consider, however, the results of a new experiment, identical to the previous one except that the class variable is defined as $C = A_1 \vee (A_2 \vee A_3)$, as in the second experiment. In this case, the sophisticated strategy remains superior after random rerepresentation: The two strategies choose trees of differing accuracy in 21 of 25 trials and in 14 of these, the tree chosen by the sophisticated strategy is the better of the two.[8] In the 21 discrepant cases, trees chosen by the naive strategy attain an average accuracy of .651 compared with .672 for trees chosen by the sophisticated strategy.

The explanation for the difference between the $C = A_1$ and $C = A_1 \vee (A_2 \wedge A_3)$ cases is simple. If we represent boolean functions in tabular form as in Table 4, then, for functions of five attributes, the table will consist of 32 rows. When we represent the function $C = A_1$, exactly half of these rows will have the value **T** in the C column. Rerepresentation in terms of new attributes B_i does not alter this property, so long as the vectors $B_1B_2B_3B_4B_5$ correspond one-to-one with the vectors $A_1A_2A_3A_4A_5$. Likewise, in any rerepresentation of the function $C = A_1 \vee (A_2 \wedge A_3)$, the value **T** will appear 24 times in the C column. By contrast, a randomly chosen boolean function may have any number of Ts in the C column.

In general, the greater the purity of the C column—the closer it is to consisting wholly of Ts or Fs—the simpler the smallest corresponding tree will be on average in a randomly chosen representation and the greater the advantage of the sophisticated strategy averaged

Table 3. Effect of purity after rerepresentation.

Number of T Values	Naive Strategy Superior (cases)	Average Accuracy		Difference in Average Accuracy
		Naive Strategy	Sophisticated Strategy	
16	7 of 7	.655	.602	5.3 percent
20	5 of 12	.600	.614	−1.4 percent
24	7 of 21	.651	.672	−2.1 percent
28	0 of 19	.637	.716	−7.9 percent
32	0 of 22	.638	.777	−13.9 percent

over all possible representations. Table 3 illustrates this by comparing results of the two experiments just described—in the first and third rows, respectively—with three more in which the number of **T** values is 20, 28 and 32. In this last case, the class variable is defined as $C = \mathbf{T}$ under any representation and always corresponds to a one-node tree. At this extreme, it is easy to see why the sophisticated strategy is superior.

The results on rerepresentation presented here follow directly from the fact that overfitting, as defined at the outset of the article, depends on the notion of complexity, which itself is representation-dependent. Overfitting occurs when a complex model outperforms a simple one on training data, but not in true accuracy; but the complexity of a tree model depends on the representation of attribute data. An overfitting avoidance scheme that serves as a bias toward models that are considered simple under one representation is at the same time a bias toward models that would be considered complex under another representation. Unless we know that the bias induced by the application of a particular overfitting avoidance scheme under a particular representation scheme is appropriate, we have no reason to expect a sophisticated strategy to outperform a naive one.

An incidental conclusion of this section is that the purity or negative entropy of values taken on by a boolean function might serve as a representation-independent measure of its complexity for decision tree induction. Though we intuitively think of $C = A_1$ as simpler than $C = A_1 \vee (A_2 \wedge A_3)$, the reverse would be true according to the proposed measure. This reflects the fact that $C = A_1 \vee (A_2 \wedge A_3)$ corresponds to a simpler tree in most representations and hence is relatively more conducive to application of a simplicity-biased tree induction strategy like CART. The proposed measure would, in general, need to take into account the likelihood of attribute vectors; in experiments reported here, they are equally likely. This means, for example, that we cannot expect to gauge the simplicity of the underlying function in an applied problem by the prevalence of the various classes in the training set.

5. Effect of the size of the training set and instance space

As noted earlier, the experiments reported here are based on an instance space of only 2^5, or 32, possible attribute vectors that is extensively sampled by training sets consisting of 50 cases. A natural question to ask is whether the degradation of performance due to overfitting avoidance observed in these experiments might be due to this abundance of training data relative to the size of the instance space.

In fact, under the conditions of Section 4.4, where all boolean functions were equally likely, the superiority of the naive strategy holds regardless of the size of the training set or the number of attributes. Moreover, a naive strategy is superior under these conditions, not simply to the particular overfitting avoidance strategy chosen as representative for the purposes of this paper, but to *all* pruning strategies in a broad class that includes most of those best known to machine learning researchers.

The main purpose of this section is to present an argument demonstrating these claims. Since this article is meant to provide insight and bolster intuition, the argument is not fully formal. It should be clear, however, that each stated point could be made perfectly rigorous.

Two general points are worth noting before the argument is presented. First, although only the experiment of Section 4.4 is discussed here, the size of the training set does not affect the relative standing of the naive and sophisticated strategies in *any* of the experiments reported in this article.

Second, while it does not affect relative standing, the size of the training set does affect the ease with which relative standing may be confirmed empirically. With much *less* training data—or equivalently, many more attributes—both the naive and sophisticated strategies would frequently learn nothing; often, both would perform at the null level, the number of discrepant trials would decrease and it would be necessary to run more total trials to prove a significant difference between the strategies. With much *more* training data, or fewer attributes, the strategies would frequently both learn perfectly, inducing the true underlying relationship; again, this would decrease the number of discrepant trials and increase the total number necessary to prove a significant difference. Thus, as noted in Section 3, the size of the training set relative to the number of attributes was *conveniently* chosen in these experiments, allowing significant differences between the two tested strategies to be empirically demonstrated without a large number of trials.

5.1. Overfitting avoidance decreases predictive accuracy when all functions are equally likely: An analytic demonstration

The argument, now, runs as follows. Under the conditions of Section 4.4, boolean functions are chosen at random, each function as likely as any other. Suppose we represent functions as in Table 4, with a row for each possible attribute vector and a final column showing the assigned class C. Then we may implement the random selection of boolean

Table 4. Tabular representation of an n-ary boolean function.

$A_1A_2A_3 \ldots A_{n-1}A_n$	C
TTT ··· TT	
TTT ··· TF	
⋮	*to be filled in*
FFF ··· FT	
FFF ··· FF	

functions by repeatedly flipping a fair coin marked **T** on one side and **F** on the other and using the results of these flips to fill in the C column of the table.

It should be clear that this approach does implement an equiprobable distribution over the 2^{2^n} possible functions and that any other implementation would have to behave identically, that is, *as if* we were flipping a fair coin to determine the C column.

Under the conditions of Section 4.4, training cases consist of an attribute vector together with a class designation. This class designation is normally the true value of C, as listed in the filled-in C column of Table 4, but with some fixed probability it will be the complement of C.

Now suppose the C column of Table 4 has been filled in by someone else, flipping a coin as described, and consider trying to induce the true value of C for any arbitrary row—the first one, for example. Before any training cases have been observed, we clearly know only that the value of C in this row has an equal chance of being **T** or **F**. Suppose the first training case has an attribute vector **TTT**$\cdots$**TF**, matching the vector in the second row of Table 4, and a class designation of **T**. How does this help us to determine the value of C in the first row?

The answer, it should be clear, is that this training case yields no information at all about the value of C in the first row. It gives us a certain amount of probabilistic information about the value in the second row, that is, about the outcome of the second coin flip. But this second coin flip was entirely independent of the one that determined the value of C in the first row. Even if we knew—on the basis of 1000 training cases with the attribute vector **TTT**$\cdots$**TF**—that the value of C in the second row of Table 4 is almost certainly **T**, we still would know nothing about the value of C in the first row.

This is a first key point. The only cases in the training set of relevance in inducing the value of C in the first row are those with the attribute vector **TTT**$\cdots$**TT**. The independence of the coin flips determining the true values of C implies that, instead of a single induction problem, we are really faced with 2^n separate problems, one for each row.

Moreover, it is easy to see what to do for each row. The optimal induction strategy for determining the value of C in the first row, for example, is to guess that it coincides with whichever class has most often been associated with the attribute vector **TTT**$\cdots$**TT** in the training set.[9] The simulated noise makes this strategy fallible, but it is the best we can do. If **TTT**$\cdots$**TT** is designated as class **T** in 500 training cases and **F** in 499, we would not place much weight on our induction that the true value of C is **T**, but we certainly have no reason to guess **F** instead.

In short, the optimal strategy under the conditions of the previous section is rote learning regardless of the size of the training set, the number of attributes or the amount of noise corrupting C. For each attribute vector, we remember the class that has been observed most often and use this for prediction.

Note also, and this is a second key point, that the argument just given can be extended if we are forced to make a single class prediction for any fixed group of attribute vectors. For example, if we must assign the same class to the vectors **TTT**$\cdots$**TT** and **TTT**$\cdots$**TF** for purposes of prediction, the best we can do is to choose the class observed most often among examples of these vectors in the training set. The independence of the coin flips makes the class observed for any other attribute vectors irrelevant.

So far, nothing has been said about decision trees. Suppose, however, that we are considering any tree induction strategy with the following properties:

1. It begins by building a tree T to fit the training set.
2. It may choose to prune T, yielding a smaller tree P.
3. In any case, after a tree is chosen, class predictions are assigned to match the majority class of training cases at each leaf.
4. There is a positive chance that a tree P will be chosen that differs from T in some class predictions.

Consider any leaf of the tree T and the set S of attribute vectors routed to it. By the third property, the class predicted for this set is chosen according the optimal strategy noted in the previous paragraph. If T is pruned to produce P, a single value will still be used as the predicted class for all attribute vectors in S. If this is the same value assigned by T, the expected accuracy of the new tree with respect to the attribute vectors in S is unchanged, but if P assigns the complementary class value to these vectors, the expected accuracy is reduced, since T's assignment was optimal.

This argument holds for every leaf of T. If there is no change in the predicted class for the associated attribute vectors, the effect of pruning on expected accuracy is null; if there is a change, the expected effect is negative.

The expected accuracy of P is thus no better than the expected accuracy of T. By the fourth property, it is sometimes worse. Hence, a strategy that always chooses T will be strictly superior to one that satisfies the four properties given above and sometimes chooses P.

5.2. Related points

A few points are worth noting about the argument just presented. In Section 4.4, the data generation model makes all attribute vectors equally likely and applies a fixed level of noise to the class value C, but neither of these properties is referred to in the argument just given. We may assume any distribution over attribute vectors and allow noise to vary over time or between parts of the instance space without affecting the conclusion that pruning strategies conforming to the description given decrease predictive accuracy.[10] The key assumptions are just that all functions are equally likely and that noise affects only the class variable.

A second point is that, since the argument depends heavily on the independence of the class value chosen for various attribute vectors, it is tempting to conclude that this independence is primarily responsible for the "anomalous" degradation of performance due to overfitting avoidance. In fact, independence is not the culprit.

The experiment of Section 4.4 and the argument of this one show that, if we do not have any prior reason to believe that some functions are more likely than others, most common techniques of overfitting avoidance can be expected to degrade performance when noise affects solely the class variable. That is, in a sense, it is not independence but ignorance that is key. If we do not know in advance that some functions are more likely than others to underlie data generation, we cannot expect overfitting avoidance to help. It is useful only if we have reason to believe that the implicit bias is appropriate.

Moreover, Sections 4.2 and 4.4 suggest clearly when this bias *is* appropriate. Overfitting avoidance helps when the underlying function is simple, in the sense that it may be represented by a small tree, and it hurts when the underlying function is complex. The degradation caused by pruning when all functions are equally likely is due, not to the property of independence that simplifies the argument presented above, but to the fact that a great majority of all possible functions are complex. We can create, literally, any number of distributions over functions that share this property without also implying the property of independence; hence, again, the issue is not independence, but the degree to which the bias of a particular overfitting avoidance technique is appropriate to the distribution of problems.

To reiterate, assuming that class values are chosen independently simplifies the argument, but is not otherwise essential. So long as complex functions predominate sufficiently an analogous argument will show that all overfitting avoidance strategies in the broad class described above decrease expected predictive accuracy.

This section leaves open several important questions. First, what happens if noise affects attributes as well as the class variable? Second, if overfitting avoidance techniques improve performance only inasmuch as they indirectly take into account prior knowledge about the likely relative performance of various models, why not abandon overfitting avoidance and take prior knowledge into account directly? Third, finally, what are the arguments that have led so many researchers to believe that overfitting avoidance increases expected predictive accuracy and how can they be reconciled with the argument of this section? These questions are addressed, respectively, in Sections 6, 9 and 8 below.

6. Attribute noise and the value of overfitting avoidance

In the experiments of Section 4, simulated noise was applied only to the class variable C, attribute values A_1 through A_5 being treated as noise-free. In fact, in many practical application domains, attribute noise may be less common. This is because the *effect* of classification noise is present whenever the observed attributes of an instance do not completely determine its class. In other words, while classification and attribute noise may both arise through measurement and reporting errors, the former is also a consequence of what Mingers (1989) calls *residual variation* in the class variable.

In this section, experiments with attribute noise illustrate two main points: first, that attribute noise has a very different effect than classification noise on the value of overfitting avoidance and, second, that noise may affect attribute values in many different ways and with distinct consequences. In these experiments, the accuracy of trees selected by the naive and sophisticated strategies has been measured empirically on the basis of 10,000 fresh examples generated under the same conditions as the training data. As a result, trees with equal accuracies, which would have been excluded from the analysis in preceding sections, may sometimes appear different and be included here.[11]

A first experiment adds attribute noise to conditions described in Section 4.4. For each trial, the class variable C is defined by a boolean function chosen at random over the space of 2^{2^5} possibilities. The class variable is complemented with probability .2 to simulate the effect of classification noise. In addition, however, individual attribute values are also complemented with probability .2 to simulate the effect of attribute noise.

The result is that, while trees chosen by the naive strategy are almost always superior under the original conditions, attribute noise tips the balance in the opposite direction. In 20 of 25 trials, the two strategies choose trees with different empirical accuracies and, in 14 of these, the tree chosen by the sophisticated strategy performs better. The superiority of the sophisticated strategy is significant at the .94 level.[12] Thus, the presence of attribute noise of this kind substantially increases the value of avoiding overfitting.

Intuitively, as the level of noise in the attribute labelling a tree node increases, the usefulness of the distinction effected by the node decreases. Eventually, the harm done in splitting the training data into two portions outweighs any possible benefit of treating the two parts of the instance space differently.[13]

The experiment just reported provides a possible justification for overfitting avoidance. It is worth noting, however, that the outcome of the experiment depends critically on the particular model of attribute noise employed. This model assumes, among other things, that errors are independent. If errors are correlated, results may be quite different.

In a second experiment, conditions are as in the first except that attribute *vectors* rather than individual values are complemented with probability. 2. Note that the probability that an individual value will be complemented is still .2 and, hence, that the expected number of attribute value errors is unchanged. The only difference is that the probability of an error is now dependent on the occurrence of other errors in the same attribute vector.

Under these conditions, trees with different empirical accuracies are chosen in 15 of 25 trials; the sophisticated strategy picks an apparently superior tree in eight of these 15 and the naive strategy picks the apparent winner in seven. That is, the result is as close as possible to a tie.[14] Since Section 4.4 showed the naive strategy superior in the absence of attribute noise, the effect of adding it is clearly still to increase the value of avoiding overfitting. Comparison with the previous experiment shows, however, that this correlated form of attribute noise is less favorable to the sophisticated strategy than the independent model that has drawn more attention in decision tree research (Quinlan, 1986).

In a third experiment, errors are introduced by replacing attribute vectors with *random* vectors. By setting the probability of such a substitution at .4, we maintain a .2 probability that an individual attribute value will be complemented. In this experiment, the naive strategy proves clearly superior, producing the tree with a higher empirical accuracy in 11 of 14 cases in which different empirical accuracies are observed. This result is significant at the .97 level. The accuracy averaged over 14 discrepant cases is .649 for trees chosen by the naive strategy versus .604 for trees chosen by the sophisticated strategy.

Intuitively, replacing an attribute vector with a random alternative is almost—though not quite—the same as beginning with a new attribute vector and assigning it to a random class.[15] This third type of attribute noise thus acts almost as a kind of classification noise and we ought not to be surprised that it does not tend to produce conditions favorable to overfitting avoidance.

The three models of attribute noise illustrated in this section are all plausible, in the sense that each may accurately reflect the effect of noise in some practical application domains, and yet they have quite different consequences for someone deciding whether to adopt an overfitting avoidance strategy. We conclude only that the effects of attribute noise are distinct from those of classification noise—as far as the problem of overfitting is concerned—and that these effects depend critically on the manner in which noise affects attribute values.

7. Recap of the experiments

To reiterate, overfitting avoidance strategies are a form of bias and their effect on performance is determined by the degree to which this bias is appropriate rather than by any inherent advantage in guarding against overfitting. For data reflecting simple relationships, as in many reported examples, a bias toward simplicity is appropriate and overfitting avoidance does improve performance over a simple, best-fit approach. On the other hand, the reverse is true for data reflecting sufficiently complex relationships.

Classification noise only reinforces the negative effect of bias for data of this kind. Despite the fact that overfitting avoidance is sometimes considered a means of coping with difficulties caused by residual variation and other forms of classification noise, for complex problems it increases the chances that overfitting avoidance will result in the choice of an inferior tree.

Moreover, in two senses, most problems are complex. First, complex relationships make up the overwhelming majority of possible relationships for any fixed representation scheme. Second, many simple problems in any fixed representation scheme are likely to be complex with respect to the overwhelming majority of alternative representations. The reported success of overfitting avoidance strategies in empirical trials thus indicates that these trials were conducted on a special subclass of problems using a special subclass of possible representation schemes.

Finally, the presence of attribute noise may serve to justify overfitting avoidance, if noise affects attribute values independently. Various types of correlated attribute noise, however, may have an inconclusive effect or even act to make overfitting avoidance inappropriate.

8. Misconstruing overfitting avoidance

Machine learning researchers often speak of overfitting avoidance techniques as if their purpose was to distinguish between structure and noise in training data or, equivalently, to determine the complexity appropriate for a model of the data generation process. It ought to be clear, however, that it is impossible to make such a distinction or determination on the basis of the training data. When models of differing complexity account equally well for observed data, nothing in the data can help us choose between them.[16]

To take a simple example, suppose a training set consists of the four observations collected in Table 5. There are a huge number of models that account equally well for these observations—among them $C = A_1$, $C = A_2 \otimes A_5$ and $C = (A_3 \wedge (A_4 \vee A_5)) \vee \neg(A_3 \vee A_4 \vee A_5)$—and the *data* can tell us nothing about which of all these to prefer. To choose, we must rely on knowledge of the inherent plausibility of the various models—domain knowledge, for example, that might suggest that the true model is likely to be quite complicated or that it is likely to involve A_5 or that the attribute vector **FFFFF** must be associated with the class value **F** and so on.

Table 5. A four-observation training set.

Observation	$A_1A_2A_3A_4A_5$	C
1	TTTTF	T
2	TFTFT	T
3	FFFFF	F
4	FFFTF	F

The fact that data must be supplemented in this way when many models are equally effective in accounting for observations might be taken as self-evident, but it also follows formally from a straightforward and standard application of Bayes rule: $P(M_i | D) = P(D | M_i)P(M_i)/P(D)$. If we want to know which of a number of data generation models M_i is most likely to be responsible for observed data D, we should pick the model maximizing $P(M_i | D)$. But $P(M_i | D)$ is proportional to $P(D | M_i)P(M_i)$; hence, if many models are equally plausible in explaining D, yielding equal values of $P(D | M_i)$, we must know the relative values of $P(M_i)$ to decide between them. No matter how sophisticated or ingenious, algorithms for extracting information from the training set D cannot help.[17]

In practice, however, machine learning research on overfitting avoidance has often proceeded as if the reverse were true. This section examines some of the theoretical, intuitive and empirical support for this opposing position to see how it can be reconciled with the arguments and evidence presented here.

8.1 Theoretical considerations

8.1.1. Occam's razor

One reason overfitting avoidance has been construed as an inherent improvement rather than a form of bias is that theoretical results have sometimes been misunderstood or misapplied. The well-known "Occam's Razor" paper (Blumer et al., 1987) provides a case in point.

The gist of this paper is that, by sticking to simple models, a learning algorithm may guarantee that performance on training data will carry over substantially to fresh data. Formal mathematical arguments—using Valiant's PAC model (Valiant, 1984)—demonstrate a probabilistic performance guarantee of this sort for algorithms that choose a restricted model space on the basis of the amount of training data available, considering complex models only when the training set is large.

For present purposes, however, the critical point is what the paper does *not* say. First, it makes no claim that restricting consideration to simple models increases expected predictive accuracy. Suppose algorithm A considers only simple models and finds one, M_A, that performs well on the training data. And suppose algorithm B considers a wider class of models and finds one, M_B, that performs even better on the data. Results from the "Occam's Razor" paper may tell us we can be more *sure* about M_A's performance on fresh data than about M_B's but they say nothing about the relative expected performance of the two models.

Second, the paper depends on no property of simple models other than that there are relatively few of them. Roughly speaking, the authors assign a bit-string identifier to each model and associate the complexity of a model with the length of its identifier. Thus, there may be no more than 2^n models of complexity less than n. The argument depends critically on this fact, but the actual assignment of identifiers to models—the designation of certain models as simpler than others—is arbitrary. Thus, the paper does not claim, and is not intended to claim, any special importance for the models a human expert might naturally think of as simple.[18]

In short, it would be a mistake to interpret the sound theoretical arguments of this paper as demonstrating that small trees and other intuitively simple models are more likely to perform well on fresh data than complex alternatives that do as well or better on the training data. William of Occam's original statement that entities should not be multiplied unnecessarily has sometimes been construed to mean that, other things being equal, simple hypotheses are more likely to be predictive. The "Occam's Razor" paper simply does not address this interpretation of Occam's statement; hence, it provides no justification for eschewing complex models when predictive accuracy is the goal.

8.1.2. The bias-variance argument

Another important example of a theoretical argument that has sometimes been misinterpreted is the bias-variance analysis of Breiman et al. (1984). In this case, the authors must take responsibility for at least a part of the resulting confusion. Although their analysis is correct and useful as far as it goes, it does not go as far as introductory remarks would suggest. Anyone depending on these remarks rather than on details of the theoretical derivations would likely be misled.

The CART program considers a series of increasingly pruned versions of a tree fit greedily to the training data. Breiman et al. note that, in their empirical trials with CART, true error rates often follow the pattern shown in Figure 1 if the complexity of trees is measured by the number of leaf nodes. Error rates first decrease rapidly with increasing complexity and then slowly increase again. Also, the error rate for the most complex trees is never more than twice the lowest achievable rate.

The bias-variance analysis is introduced as a heuristic attempt to understand these points, but, in fact, it concentrates on bounding the error rate rather than on explaining the U shape of observed error rate curves.

Any decision tree T may be viewed as a partition of the attribute space, since its leaves receive mutually disjoint and exhaustive subsets of the possible attribute vectors. Inasmuch as T's accuracy is less than that of the best possible tree, T_{opt}, it must either be because the partition is not as good as it could be or because the wrong class is assigned to one or more partition elements—that is, to the leaves. Breiman et al. call the first factor *bias* and the second *variance*.[19] More precisely, let $T_{bias\text{-}only}$ be a tree with the same structure as T, and hence the same associated partition, but with classes assigned optimally to the leaves. Breiman et al. consider the difference in the error rates of T_{opt} and $T_{bias\text{-}only}$ a

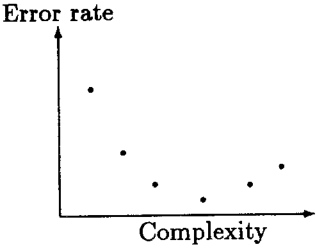

Figure 1. A typical error-rate curve.

measure of the bias, the suboptimality of the partition, and the difference in the error rates of $T_{bias\text{-}only}$ and T a measure of the variance, the suboptimality of the class assignments.

In introducing these terms, Breiman et al. suggest that bias decreases and variance increases along the sequence of trees that CART considers, if these trees are ordered from the simplest to the most complex. Moreover, in paraphrasing their basic claims before entering into theoretical derivations, they state that "the variance term increases slowly as L [the number of leaves or partition elements] increases and is bounded by a slow growth factor in L."

In fact, however, while the mathematical arguments in Breiman et al. (1984) show that bias decreases and that variance is bounded, they do not show—or even attempt to show—that variance increases; nor certainly do they account for the upward sloping portion of the U-shaped error rate curve by showing that variance must increase faster than bias decreases at sufficient levels of complexity. In the sequence of trees considered by CART, partitions associated with complex trees are refinements of partitions associated with complex trees, and this implies that bias must decrease—or at least not increase—with increasing complexity. If the total error rate increases with increasing complexity in some empirical trials, it must be because variance increased in those trials; and if the error rate curve is U-shaped, variance must, at some level of complexity, have increased faster than bias decreased. But it cannot be proved that this is true in general, because the opposite is sometimes true. A simple case in point is when true class values are assigned independently to each element of the instance space. As argued in Section 5, the finer the partition in this case, the lower the expected error rate will be. In other words, this case is a counterexample proving that the expected error rate curve will not always be U-shaped.

When a U-shaped error rate curve is expected, it is clearly not optimal to choose the most complex of the trees CART considers. But the bias-variance analysis does not give us any reason to expect a U-shaped curve. Moreover, even when the curve *is* U-shaped, the analysis does not show—and is not intended to show—that statistical means like those employed by CART locate a low point on it, increasing predictive accuracy by choosing a tree that is suboptimal for the training set. As noted at the beginning of this section, statistical methods, however sophisticated, are inadequate for this purpose.

8.1.3. Large sample results for minimum description length methods

A final theoretical argument sometimes invoked to justify overfitting avoidance is Barron and Cover's (1991) proof that adherence to the minimum description length principle in modeling guarantees that models derived from training data will converge to the true one as more and more data is collected. In conjunction with Quinlan and Rivest's (1989) practical suggestions for using minimum description length as a criterion for inducing decision trees from data, this proof is easily misconstrued as demonstrating that it is better to choose a small tree that minimizes Quinlan and Rivest's measure of description length than a larger one that performs better on the training set.

In fact, Quinlan and Rivest themselves state quite clearly that their application of the minimum description length principle to tree induction amounts simply to the application of a bias. As they point out, the effect is precisely the same as assuming a certain prior distribution over true models and applying straightforward Bayesian modeling techniques. Thus, the minimum description length approach is no more or less effective than its bias is appropriate to the domain of application.

Barron and Cover's result proves a desirable property of the minimum description length approach in the long run, as the size of the training set tends to infinity. It does not demonstrate that the approach is superior to others for training sets of finite size and it could not do so; the bias effected by minimizing description length is neither inherently good nor inherently bad.

8.2. Intuitive arguments in favor of overfitting avoidance

8.2.1. Noisy data and overfitting

Another general reason overfitting avoidance has been construed as an inherent benefit, rather than as the application of a bias, is that it has often seemed intuitively necessary as a means of dealing with noisy data.

Roughly speaking, the intuitive argument runs this way. Some of the patterns to be found in noisy data reflect true underlying structure, but others arise merely by chance as a consequence of noise. An induction procedure that simply strives to achieve optimum performance on the training set will construct a model that captures both kinds of patterns. It will go too far, fitting noise as well as structure and attempting to extract more information from the data than the data really contain. For this reason, we must adopt some mechanism for distinguishing between real and spurious patterns when dealing with noisy data.

There are two basic flaws in this argument. First, even assuming that an induced model includes one component reflecting the true model and another reflecting chance patterns in a training set, it does not follow that we can use the same training data to distinguish between the two. In fact, for reasons set out at the beginning of this section we definitely cannot wring extra information of this sort from the data; to distinguish between real and spurious structure in an induced model, we must either have fresh data—more about this in a moment—or independent information indicating which models are inherently more plausible. In other words, even if it were true that induction in noisy domains necessarily leads to overfitting, it would still be impossible to improve performance by strictly statistical means, using the training data in some sophisticated fashion to avoid the problem. Data cannot tell us which of the patterns in them are real.

A more fundamental flaw, however, is the assumption that a naive fit-as-far-as-possible strategy will lead to overfitting rather than underfitting. On the contrary, if the true underlying structure is complex relative to the size of the training set, even an unbridled induction procedure will stop short, producing a model that is too simple rather than too complex. Given a five-observation training set, none of the best-known decision tree induction methods will produce a tree with more than five leaves; if we apply these methods to training sets of this size in domains for which trees with fifty leaves typically represent the true underlying structure, they will normally underfit, even if the data is noisy. The example is contrived, but it illustrates an important point: Whether we need a procedure for deleting parts of the induced model or one for adding parts depends entirely on the mix of problems our induction methods will face. And again, to restate and expand the first point, nothing in the data can help us to decide either what is superfluous or what is lacking. We must have some other way of knowing whether true structure tends to be simple or complex.

Of course, data *can* be used to distinguish between real and apparent structure if they consist of fresh cases. If we divide available data into two portions and use the first to obtain a series of pruned versions of a large tree, the second portion can be used effectively to choose between these. But the tree *T* that results from using the whole training set in this two-phase fashion cannot be compared effectively with another tree, T^*, obtained directly from the whole training set by fitting it as far as possible. If T^* fits the whole training set as well or better than *T*, nothing in the data can be used to prove that *T* is superior. The data as a whole cannot tell us which of the patterns in them to trust and the two-phase approach yields no inherent benefit.

8.2.2. Cross-validation and fresh data

The same is true of resampling approaches like the cross-validation scheme employed by CART and the sophisticated strategy of this article. Cross-validation has generally been understood to provide the benefit of testing on fresh data without the cost of collecting it, but many of the experiments presented here illustrate the fact that choosing models on the basis of cross-validated performance estimates amounts to applying a bias that may either improve or degrade performance, relative to a naive strategy, depending on the problem distribution. By contrast, choosing models on the basis of performance on true fresh data would yield a provable increase in expected accuracy regardless of the problem distribution. The analytic argument of Section 5 extends the validity of the claim that cross-validation provides no inherent benefit in decision tree induction to many schemes other than the particular one employed by the sample sophisticated strategy. Under the conditions of that section, *any* cross-validation approach that sometimes results in the choice of a tree other than the one performing best on the training cases will be inferior to a simple pick-the-apparent-best alternative. This single example suffices to show that the effect of such a cross-validation approach depends on the problem distribution and that, even if it increases accuracy in some circumstances, we may be sure it will decrease accuracy in others.

8.2.3. Statistical significance

Yet another line of intuitive reasoning about overfitting avoidance depends on the statistician's notion of significance. Inasmuch as statistical significance is meant as an indicator of the reality of apparent or hypothesized empirical patterns, it has often seemed possible to improve performance by eliminating the parts of an induced model that do not yield statistically significant increases in fit. Two immediate problems with this idea are the necessity of fresh data and a correct model of noise for valid significance calculations. But a more fundamental problem is that statistical significance provides the wrong kind of information for someone interested in optimizing performance.

To draw an analogy, the "Occam's Razor" paper offers probabilistic guarantees that performance on the training set will carry over to new cases if only simple models are considered, but it does not say that expected performance is likely to be improved when this restriction is observed. If we consider a wider space of models, we may be less *sure* how well our chosen model will perform on new cases, but it is no more or less likely to perform well.

Likewise, if we rely only on the parts of a model that yield statistically significant increases in performance, we are guaranteed probabilistically that those same parts will yield increases in performance when the model is applied to fresh data. But if we also include in our model parts that yielded statistically *in*significant increases in performance, we may be less sure how these will affect future performance, but we have no reason to think the effect will be negative. For example, suppose we consider a tree *T* and a pruned version *P* and find that, although *T* performs better than *P* on the training set, the increase in accuracy is not statistically significant. This means that the improvement might plausibly be due to the chance effects of noise rather than to any real advantage of *T* and, as a consequence, that we ought not to be surprised to find that *P* performs as well as *T* on fresh cases. It does not, however, tell us we should be surprised if *T does* continue to outperform *P*. In fact, it tells us nothing about which model is likely to achieve higher accuracy in the future.

In short, reliance on statistical significance is a means of reducing uncertainty about future performance, not of optimizing it.

8.3. Empirical evidence

Despite all that has been said here, it may still be objected that in practice overfitting avoidance is necessary and beneficial, that, empirically speaking, overfitting avoidance is a proven success and that, whatever the case with the artificial problems considered in this article, experience with *real* data justifies application of the various well-known pruning techniques.

The problem with this line of argument is that it makes an unfounded leap from the particular practical, real-data problems on which overfitting avoidance techniques have been demonstrated to the general class of all such problems. While it is true that overfitting avoidance has improved performance in many practical problems tackled by decision tree researchers to date, this may have as much to do with the selection of problems as with the techniques applied.

Holte (1991), Jensen (1991) and Weiss et al. (1991) all present evidence that many of the data sets typically employed in testing machine learning induction algorithms reflect very simple relationships. Certainly it would be a mistake to take this as empirical proof that, in practice, real data tend to reflect simple relationships. For the same reason, it would be wrong to conclude from the empirical evidence that techniques favoring simple relationships are generally superior in practical applications. The value of these techniques in practice depends on the problems practitioners choose; and it would be foolhardy to expect that future practitioners will confine their application of overfitting avoidance techniques to problems qualitatively like those already tried.

The key point is that it is not real data that determine if overfitting avoidance will improve or degrade performance, but rather characteristics such as the complexity of the underlying relationship and the manner in which noise affects attribute values. Perhaps some important applications do share characteristics that make overfitting avoidance beneficial, but—because of the widespread assumption that overfitting avoidance provides an inherent benefit—we know little or nothing today about these determining characteristics. As a result, we cannot say when new problems are sufficiently like old ones to justify application of overfitting avoidance techniques. And it seems safe to predict that indiscriminate use of these techniques will sooner or later lead to performance degradation in real-data problems of practical importance.[20]

9. Practical significance

This article will have served a practical purpose if, as just suggested, it causes researchers to study the conditions under which existing overfitting avoidance techniques can be expected to increase predictive accuracy in induced models. But there is a more important and general lesson to be drawn from the observation that overfitting avoidance is a bias.

As already noted, many of the classification data sets employed by machine learning researchers reflect simple relationships. And inasmuch as simple models are appropriate, *any* procedure that effects a bias toward simplicity will improve performance—whether it is based on removing statistically insignificant parts of a model, minimizing description length, limiting consideration to short rules or optimizing cross-validated performance. The improvement is not due to statistical significance, information theory, resampling or overfitting avoidance, however, but to the indirect application of an appropriate bias. All of these techniques will decrease performance—relative to a greedy naive strategy—if we apply them to problems of sufficient complexity.

The intended message of this article is not, however, that there is something wrong with existing induction techniques. Even if their domain of application is less universal than may have been generally understood, it is certainly broad and important. The key point is that our attention has been misplaced, even with regard to the best of these useful techniques. As a community, we have concentrated on statistical significance, information theory and overfitting avoidance, while our practical successes have been due to the appropriate application of bias.

It is important to exercise caution in applying existing overfitting avoidance techniques and urgent for us to understand when they can and cannot be expected to help. But the critical message is that these techniques rely on the *indirect* application of bias and that we can expect to do better by applying it directly. If the successes of overfitting avoidance are due to our implicit reliance on key domain characteristics, then we should be able to improve performance—even in domains where overfitting avoidance has proved quite successful—by relying on those characteristics explicitly and designing techniques to take full advantage of them.

10. Related work

The idea that overfitting avoidance is a form of bias is implicit in Buntine's Bayesian perspective on the problem of inducing decision trees (Buntine, 1990). An early study by Quinlan (1986) of the effect of noise on decision tree induction is also related to the work reported here, though Quinlan was concerned with the effect of noise on a single induction strategy rather than with its effect on the relative value of alternative strategies.

The empirical comparison study by Mingers (1989) has already been cited. The argument of this article should discourage interpretation of the study as indicating that one pruning strategy is better or worse than others; relative performance is determined by the choice of a test suite and does not reflect relative performance on other problems unless these are known to be essentially similar.

This article is based partly on the examination of a minimal overfitting avoidance problem considered in Schaffer (1991). Follow-up work, conducted since this article was submitted for publication, is reported in Schaffer (1992a; 1992b).

Acknowledgments

I thank Tom Ellman, Peter Clark and Francesco Bergadano for discussions that motivated the work reported here and Robert Holte, Wray Buntine and William Cohen for extensive comments on draft versions of the article.

Notes

1. The treatment in Mingers (1989) is much simpler and briefer, but it leaves out key details and does not describe the cross-validation procedure described by Breiman et al. and employed here.
2. All illustrations in this article involve two-class problems, although the qualitative results are more general.
3. For the statistically sophisticated, this is a standard, one-sided binomial sign test as described in Conover (1980). The advantage of this test over alternatives—a paired *t* test, for example—is that it is non-parametric, avoiding unwarranted assumptions of normality. This makes the test more sensitive in some of the experiments reported here, though, it is important to note, the choice of test was made before any data were collected.
4. The nearly optimal tree given in Breiman et al. (1984) has just 10 leaves; a full tree would have 128.
5. Recall that Parity($\vec{V}$) is **T** if and only if $\vec{V}$ contains an odd number of **T** values.
6. To see that there are 32! rerepresentation schemes, consider that any particular scheme can be specified by a two-column table with all the A vectors listed in a fixed order in the first column and the corresponding B vectors in the second. Each of the 32! ways of permuting B vectors in the second column gives one possible rerepresentation.
7. The superiority of the naive strategy is significant at the .99 level.
8. The superiority of the sophisticated strategy is significant at the .90 level.
9. If this attribute vector has not appeared in the training set, the true value of C is still equally likely to be either **T** or **F** from our perspective, and it makes no difference which we guess. We are implicitly assuming that error rates range from 0 to .5, corresponding to noise levels of 0 to 1.0. Higher error rates would simply reverse the meanings of **T** and **F** in the C column.
10. It is necessary, however, to assume that the distribution over attribute vectors is the same in training and test data.
11. For example, a tree that gives the correct prediction for all attribute vectors except **TTTTT** has the same accuracy as one that gives the correct prediction for all attribute vectors except **FFFFF**. But if the test set contains more of one vector than the other, the two trees may appear to have different accuracies.

12. Trees chosen by the naive and sophisticated strategies attain average accuracies of .521 and .528, respectively, for cases in which they differ. The small difference in average accuracy is due to the high error rates—equivalent to noise levels of .4—affecting both the attribute and class variables. It does not weaken the statistical significance reported, which simply reflects the fact that it is unlikely for one of two strategies to appear superior in 14 of 20 trials if, in fact, the two are equally matched.
13. This tradeoff is analyzed by Breiman et al. (1984) and discussed below in Section 8.
14. In fact, trees chosen by the naive strategy attain a higher average accuracy for these 15 cases (.586) than those chosen by the sophisticated strategy (.574).
15. The difference is that attribute vector replacement does not affect the prevalence of **T** and **F** values in the class variable. Classification noise, of the kind applied in this article, adjusts these prevalences toward .5.
16. Of course, *fresh* data can be used to choose between models developed on the basis of previous observations, but this does not invalidate the point just made. This distinction is discussed at length in Section 3.2.1.
17. For clarity, this paragraph focuses on the problem of choosing the most likely model rather than on the problem of choosing the model with the highest expected accuracy. An analogous argument for the latter is more complex, but the gist is the same. We cannot decide between models that perform equally well on the training data without knowledge of the distribution of true models in the problem-generating environment.
18. The "Occam's Razor" paper is based heavily on a much earlier paper by Pearl (1978) that makes this point quite explicitly: "From a philosophical viewpoint it is essential to note that in all cases examined the role of *simplicity* was only incidental to the analysis. We would have gotten identical results if instead of L_c [Pearl's language for describing simple models or functions] being a complexity bounded sublanguage we were to substitute an arbitrary sublanguage with equal number of functions.
19. The meaning of bias here is completely different than elsewhere in this article. Breiman et al., appealing to statisticians rather than machine learning researchers, are using the terms *bias* and *variance* in analogy with their meanings in regression analysis.
20. In fact, in work conducted since this article was written (Schaffer, 1992b), the pruning method of CART (Breiman, et al., 1984) has been shown to decrease predictive accuracy in important, real-data cases drawn from a standard machine learning database repository.

References

Barron, A.R., & Cover, T.M. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory, 37,* 1034–1054.

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

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

Buntine, W. (1990). *A theory of learning classification rules.* Doctoral dissertation, University of Technology, Sydney.

Cestnik, B., & Bratko, I. (1991). On estimating probabilities in tree pruning. In Y., Kodratoff (Ed.), *Machine learning, EWSL-91.* Berlin: Springer-Verlag.

Conover, W.J. (1980). *Practical nonparametric statistics.* New York: John Wiley.

Holte, R.C. (1991). Very simple classification rules perform well on most datasets (Technical Report TR-91-16). Ottawa, Canada: University of Ottawa, Department of Computer Science.

Jensen, D. (1991). *Induction with randomization testing: Decision-oriented analysis of large data sets.* Doctoral dissertation, Washington University, Sever Institute of Technology.

Mingers, J. (1987). Expert systems—rule induction with statistical data. *Journal of the Operational Research Society, 38,* 39–47.

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

Niblett, T., & Bratko, I. (1986). Learning decision rules in noisy domains. In M.A. Bramer (Ed)., *Research and development in expert systems III,* Cambridge: Cambridge University Press.

Pearl, J. (1978). On the connection between the complexity and credibility of inferred models. *International Journal of General Systems, 4,* 255–264.

Quinlan, J.R. (1986). The effect of noise on concept learning. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2). San Mateo, CA: Morgan Kaufmann.

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

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

Schaffer, C. (1992a). Deconstructing the digit recognition problem. *Machine Learning: Proceedings of the Ninth International Conference* (pp. 394–399). San Mateo, CA: Morgan Kaufmann.

Schaffer, C. (1992b). Sparse data and the effect of overfitting avoidance in decision tree induction. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 147-152). Cambridge, MA: MIT Press.

Schaffer, C. (1991). When does overfitting decrease prediction accuracy in induced decision trees and rule sets? In Y. Kodratoff (Ed.), *Machine learning, EWSL-91.* Berlin: Springer-Verlag.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Weiss, S., & Indurkhya, N. (1991). Reduced complexity rule induction. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (pp. 678–684). San Mateo, CA: Morgan Kaufmann.

Received April 29, 1991
Accepted August 1, 1991
Final Manuscriot March 1992