Sparse Data and the Effect of Overfitting Avoidance in Decision Tree Induction

Cullen Schaffer
Department of Computer Science
CUNY/Hunter College
695 Park Avenue, New York, NY · 10021
212-772-4283 · schaffer@marna.hunter.cuny.edu

Abstract

Overfitting avoidance in induction has often been treated as if it statistically increases expected predictive accuracy. In fact, there is no statistical basis for believing it will have this effect. Overfitting avoidance is simply a form of bias and, as such, its effect on expected accuracy depends, not on statistics, but on the degree to which this bias is appropriate to a problem-generating domain. This paper identifies one important factor that affects the degree to which the bias of overfitting avoidance is appropriate—the abundance of training data relative to the complexity of the relationship to be induced—and shows empirically how it determines whether such methods as pessimistic and cross-validated cost-complexity pruning will increase or decrease predictive accuracy in decision tree induction. The effect of sparse data is illustrated first in an artificial domain and then in more realistic examples drawn from the UCI machine learning database repository.

Introduction

It is easy to get the impression from the literature on decision tree pruning techniques [Breiman *et al.*, 1984; Cestnik and Bratko, 1991; Mingers, 1987; Mingers, 1989; Quinlan, 1987; Quinlan and Rivest, 1989] that these techniques are statistical means for improving predictive accuracy. In fact, overfitting avoidance methods in general—and pruning techniques in particular—have an indeterminate effect on expected accuracy. Overfitting avoidance constitutes an a priori prejudice in favor of certain models and, like other forms of bias, it is inherently neither good nor bad. Whether bias will improve or degrade performance depends purely on whether it is appropriate to the domain from which induction problems are drawn.

These points, demonstrated empirically and analytically in [Schaffer, 1992a; Schaffer, 1991; Schaffer, 1992b], suggest an important shift in research focus. Attempts to develop "good" pruning techniques or to compare techniques to determine which is "best" [Mingers, 1989] make no sense if the effect of each technique depends on where it is applied. Rather than looking for a single, universally applicable pruning method, we need a selection of useful alternatives and—most crucially—an understanding of the factors that determine when each implicit bias is appropriate.

This paper takes a first step toward providing insight of this kind. The paper is organized in two parts. In the first, a well-known pruning method—cross-validated cost-complexity pruning—is pitted against a no-pruning strategy in an artificial domain. This domain is designed to show as clearly as possible how the effect of pruning depends on the abundance of training data relative to the complexity of the true structure underlying data generation. In particular, when the training data is relatively *sparse* in this domain, either because few observations are available for learning or because the underlying structure is complex, the bias inherent in popular pruning methods is inappropriate and they have a negative effect on predictive accuracy.

The second part of the paper illustrates the same sparse data effect using examples drawn from the UCI machine learning data repository [Murphy and Aha, 1992]. This part shows that understanding what determines the effect of overfitting avoidance strategies is a matter of *practical* concern. In extreme cases of sparse training data in the UCI examples pruning degrades predictive accuracy by three to five percent; in moderate cases the negative effect is smaller, but still comparable to the gains often reported in *support* of proposed pruning methods.

Poisson Generator Experiments

The Poisson Generator

In experiments described in this section, data for induction is generated artificially. Instances consist of the value of a continuous attribute x that ranges from 0 to 1 and a class value in {A,B}. How the class depends on x is determined by a data generation model; Figure 1 shows a sample model. The class value is A for x values between 0 and the first vertical mark, B between the first and second marks, and so on alternately. To generate data, x values are chosen at random over

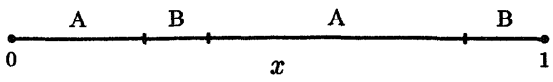

Figure 1: A sample data generation model

the interval [0,1] and paired with the associated class values. Then, to simulate the effect of classification noise, class values are complemented with probability e.

The data generator creates a new data generation model each time it is called and then uses this model to produce training and test data. Models vary only in the number and placement of the vertical cutpoints, which are determined according to a Poisson process with parameter λ. That is, the number of cutpoints is chosen at random over $\{0,1,2,\ldots\}$, the expected number of cutpoints is λ, and, once the number is chosen, cutpoints are placed at random in the interval [0,1]. The value of λ is a parameter of the data generator set by the user.

Since the number of cutpoints in a model is one less than the number of leaves in the corresponding optimal decision tree, λ is a measure of the expected complexity of the underlying model. The *Poisson generator* just described thus provides a way to generate induction problems at varying levels of complexity. By paring induction down to essentials—one attribute and two classes—the generator helps to focus attention on the effect of sparse data. Complex examples representative of induction in real application domains are discussed in below.

Algorithms for Comparison

Two tree induction algorithms are compared in these experiments, each implemented by choosing appropriate options of the CART program. CV uses the Gini splitting criterion to build trees and cost-complexity optimization through tenfold cross-validation to prune them. It differs from the default version of CART [Breiman *et al.*, 1984] only in that (1) it replaces the one-standard-error rule described in [Breiman *et al.*, 1984, p. 78] with a zero-standard-error rule that has performed better in related experiments [Schaffer, 1992a] and (2) it allows nodes of any size to be split. The default version of CART will not split nodes of less than five training cases. A second algorithm, NP, is identical to CV except that it does not carry out any pruning.

Hypothesis

The basic hypothesis underlying this paper is that the effect of overfitting avoidance depends, among other things, on the amount of training data relative to the complexity of the data generation model. Under certain conditions, characteristic both of the artificial examples of this section and the real-data examples of the next, this factor may be critical. As we consider increasingly complex models or smaller training sets in such cases, pruning methods will, according to the hypothesis, perform more and more poorly until, eventually, they *decrease* predictive accuracy. For data sufficiently sparse, in this sense, the bias inherent in techniques like pessimistic and cross-validated cost-complexity pruning is inferior to the bias inherent in a no-pruning strategy and we should expect the latter to perform better.

Experimental Methodology

To investigate this hypothesis, the algorithms CV and NP were tested on problems produced by the Poisson generator. Three parameters were varied in the course of these tests: the size of the training set, n, the average model complexity, λ, and the classification error rate, e. For each of 168 combinations of n, λ and e, the three algorithms were tested on 50 problems produced by the Poisson generator. In each case, a test set of 100 fresh instances was used to measure predictive accuracy. After 50 trials the overall accuracy of CV and NP was compared.

Experimental Results

The results of these experiments are summarized in Figure 3. Each gridpoint is marked either with "CV," if that algorithm attained the highest overall predictive accuracy for the associated n-λ-e combination, or with a dot (·) if NP is superior.[1] Note that the size of the training set decreases along the vertical axis. As the schematic diagram in Figure 2 indicates, this means that data grows sparser in each grid as we move upward or rightward.

These grids very neatly illustrate the sparse data effect: In each, pruned trees prove more predictive at the lower left and unpruned trees more predictive at the upper right. At the top and right, where data is sparsest, avoidance of overfitting through pruning degrades performance, as predicted, rather than improving it.

In each grid, a downward sloping boundary separates a domain region in which the biases of the tested pruning methods are appropriate from one in which they are inappropriate. Note that the increasing levels of noise have the effect of moving this boundary upward and to the right. Other things being equal, in this domain, noise increases the value of pruning.

Additional experiments have been carried out by the author for an induction strategy using information gain and pessimistic pruning and by Ross Quinlan [personal communication] for C4.5 with qualitatively similar results.

[1] At one gridpoint a dash (-) indicates that these algorithms performed identically.

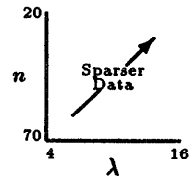

Figure 2: Schematic guide to the Poisson generator results

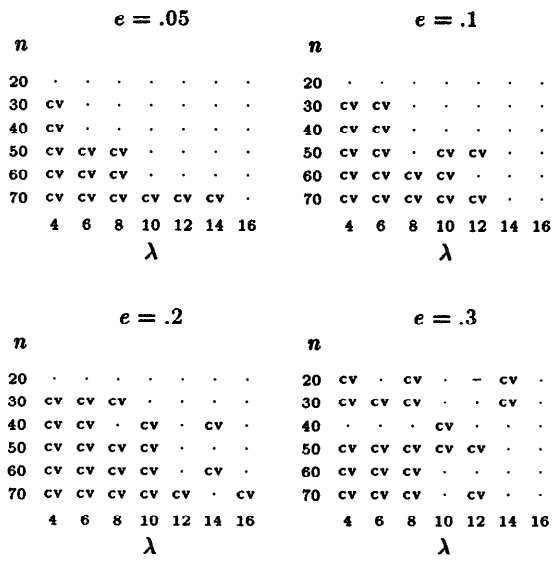

Figure 3: Results for the Poisson generator experiments

	Leaves		Accuracy			
n	CV	NP	CV	NP	Δ	Signif.
50	3.8	10.2	71.38	70.40	−.97	.12
30	2.9	6.2	69.89	68.21	−1.68	.02
15	2.0	3.4	63.79	66.68	2.89	>.99
10	1.6	2.7	58.15	61.87	3.72	>.99
5	1.5	2.0	58.45	61.71	3.25	.99

Table 1: Results for heart disease data experiments

Examples from the UCI Repository

In this section, examples drawn from the UCI repository of machine learning databases are used to illustrate the sparse data effect.[2] Except where noted, results are averaged over 50 trials in which training data is selected at random and remaining data is used for testing.

Cleveland Heart Disease Data

Table 1 summarizes the results of experiments with the Cleveland heart disease data.[3] With real data of this kind, we have no control over the complexity of the true relationship governing data generation, but we can investigate the effect of sparse data by incrementally decreasing the size of the training set. The table shows that, as we do, the predictive accuracy of trees produced by NP climbs from about one percent below those produced by CV to more than three percent above. As in the artificial domain, the effect of overfitting avoidance is negative for sufficiently sparse training data.

The last column of the table gives the statistical significance of NP's superiority, according to a paired t test; in the higher rows, this figure may be subtracted from 1 to get the significance of CV's superiority instead. A figure near .5 in the last column would indicate that there is no statistically significant difference between the algorithms.

Similar results have been obtained for the Cleveland heart disease data in experiments comparing PS, an IND-based [Buntine and Caruana, 1991] tree induction strategy using information gain to build trees and Quinlan's pessimistic method [Quinlan, 1987] to prune them, with PS-NP, a strategy that builds trees the same way, but does not prune.

[2] All data is available by anonymous FTP from the directory pub/machine-learning-databases at ics.uci.edu.

[3] CART requires training sets to include at least one representative of each class for each fold of cross-validation. Thus, even if a full training set includes such representatives and NP produces a tree as usual, CART may refuse to produce a tree using the options defining CV. Cases of this kind are excluded from the analysis in this section. All statements that one of NP or CV is superior to the other ought, for precision, to be prefaced by the phrase "when both algorithms produce trees..."

	Leaves		Accuracy			
n	CV	NP	CV	NP	Δ	Signif.
3000	608.0	659.5	74.34	74.20	−.14	<.01
2000	430.5	489.9	70.60	70.58	−.02	.37
1000	258.6	296.8	63.00	63.06	.06	.76
500	142.1	176.3	54.79	55.06	.27	.94
250	77.4	105.2	44.03	44.32	.29	.98
125	45.9	62.6	32.57	33.95	1.38	.96

Table 2: Results for letter recognition data experiments

Letter Recognition Data

The Cleveland heart disease data illustrates the fact that NP may be superior when training sets are sufficiently small. But training data may be sparse, in the sense of this paper, even for large training sets if the true relationship between attributes and classes is sufficiently complex. Results for the letter recognition data, shown in Table 2, are a case in point. This is a second clear example of the sparse data effect in real data, but, here, NP remains superior for training sets of up to 1,000 cases. The complexity of the true underlying relationship is reflected by the enormous, but highly predictive, trees constructed by both algorithms given large amounts of training data.

In experiments with PS and PS-NP, the latter proved superior for the letter recognition data for training sets of up to 16,000 cases. This would appear to confirm past suggestions that the pessimistic method may lead to consistent overpruning [Mingers, 1989].

LED Digit Recognition Data

A third example of the sparse data effect is reported in a study of the digit recognition problem in [Schaffer, 1992a]. In that paper, CV is superior to NP by about one percentage point when the training set contains 200 instances and attribute errors occur with probability .2. When the number of instances is cut to 100, however, NP proves superior by roughly an equal amount, and its advantage rises to about three percentage points for training sets of size 25.

Mushroom Data

Results for the mushroom data are more complex. An initial set of experiments, summarized in Table 3, show NP superior to CV at every tested training set size, but with a deep dip in significance near $n = 20$.[4] Given training sets of this size, both CV and NP normally discover the single attribute that accounts for about 90 percent of potential predictive accuracy and neither regularly discovers anything else of use. For smaller

[4] At this level of n, an additional 150 trials were run to confirm the weak significance. The table shows results for a total of 200 trials.

	Leaves		Accuracy			
n	CV	NP	CV	NP	Δ	Signif.
1600	6.5	7.5	99.86	99.92	.06	>.99
400	3.8	5.4	99.35	99.47	.13	.99
100	2.3	3.2	97.94	98.21	.27	>.99
20	2.0	2.2	90.70	90.73	.03	.55
10	1.9	2.1	76.86	82.28	5.42	>.99

Table 3: Results for mushroom data experiments

	Leaves		Accuracy			
n	CV	NP	CV	NP	Δ	Signif.
200	2.7	6.6	98.34	97.90	−.44	<.01
100	2.3	4.0	97.56	96.91	−.65	<.01
60	2.2	3.9	96.78	95.71	−1.07	<.01
30	2.1	2.7	93.23	92.20	−1.03	<.01
20	2.0	2.3	89.85	89.55	−.30	.05
15	2.0	2.3	85.97	87.49	1.52	.90
10	1.8	2.1	75.27	80.21	4.95	>.99

Table 4: Results for mushroom data with added noise

training sets, NP discovers the single, highly predictive attribute more consistently, and for larger training sets it leads the way in discovering additional predictive structure; in both cases, it proves superior to CV.

A salient feature of the mushroom data is the low level of noise evidenced by the extremely high accuracies achieved by both tested algorithms. Recalling the effect of classification noise in the artificial domain of the previous section, we may hypothesize that this lack of noise is what allows NP to maintain its superiority at every tested level of n. Table 4 confirms this hypothesis by showing the results of additional experiments with the mushroom data in which artificial noise complements the class variable with probability .01. With this modification, we again observe a clear example of the sparse data effect.

Results of experiments with PS and PS-NP were qualitatively much like those reported here. For the original mushroom data, PS-NP proved superior for training sets of 10 to 6,400 cases except for a middle range centered about training sets of 40 cases. With added classification noise, a clear sparse data effect was observed, with the crossover near $n = 25$.

Hypothyroid Data

Results for the hypothyroid data are the least clear of those reported here. In initial experiments comparing PS and PS-NP, this data seemed to yield another clear example of the sparse data effect, as shown in Table 5. For still larger training sets, however, the superiority of PS's pessimistic pruning *decreased* as both algorithms converged to near perfect performance.

Moreover, experiments comparing CV and NP show

	Leaves		Accuracy			
n	PS	PS-NP	PS	PS-NP	Δ	Signif.
375	5.5	10.1	98.4	98.2	−.2	.02
260	5.5	9.0	97.7	97.7	0	.50
180	5.2	7.4	97.3	97.4	.1	.91
90	3.4	6.4	95.2	95.6	.4	.98
45	1.8	4.6	93.1	94.1	1.0	>.99

Table 5: Results for hypothyroid data experiments with PS and PS-NP

	Leaves		Accuracy			
n	CV	NP	CV	NP	Δ	Signif.
2000	6.3	9.8	99.79	99.77	−.02	.19
375	4.5	5.8	98.56	98.76	.20	>.99
260	4.0	5.6	97.57	97.79	.22	.99
180	2.8	4.9	96.24	96.96	.72	>.99
90	2.1	3.9	95.41	95.46	.06	.62
45	1.6	3.2	94.83	94.62	−.21	<.01

Table 6: Results for two-class hypothyroid data experiments with CV and NP

the former superior for training sets of 45 cases—just where pruning performed worst in experiments with PS and PS-NP. Results for these experiments are shown in Table 6. Note though, that these are not directly comparable with results for PS and PS-NP. First, the PS algorithms use information gain as a splitting criterion, while the CV algorithms use Gini. Second, for practical reasons, the three positive classes of the hypothyroid data were merged for experiments with CV and NP.[5] Note also that 1,000 trials were run for $n = 45$ instead of the usual 50 in order to produce a clearer result.

One fact that may account for the difference between experiments with the hypothyroid data and the others reported in this paper is that the class distribution for the hypothyroid data is highly unequal. Negative cases account for about 95 percent of the data. In a last set of experiments, balanced data sets were constructed for each trial using all of the positive cases and an equal number of randomly selected negative cases. Data sets constructed in this manner were split into training and testing portions at random, as usual. Because the size of the test sets was normally quite small, 100 trials were run at each training set size.

The results, given in Table 7, show a uniform pattern, with NP superior at every tested training set size. It would be interesting to know if CV is superior for larger training sets, yielding a complete example of the sparse data effect, but the data available is not sufficient to run these trials.

[5]See footnote 3. Some of the positive classes are so weakly represented that they are almost certain not to appear in one of the training sets used for cross-validation and this causes the CART system to refuse to build a tree.

	Leaves		Accuracy			
n	CV	NP	CV	NP	Δ	Signif.
180	4.3	5.1	98.06	98.30	.25	>.99
135	4.2	5.0	97.40	97.74	.34	>.99
90	3.6	4.6	96.29	96.71	.42	>.99
45	2.8	3.6	94.55	95.10	.55	>.99

Table 7: Results for balanced two-class hypothyroid data experiments with CV and NP

Discussion

The results of the previous section mainly speak for themselves, but two points may be worth adding. First, it was remarkably easy to find examples like the mushroom and letter recognition data for which pruning degrades performance even for large training sets. Data sets were selected for testing more or less at random from the UCI repository and results for all but one of those data sets have been reported here.[6] Thus, there is reason to suspect that the sparse data effect is *often* of practical importance in the induction problems considered by machine learning researchers.

Second, results for the hypothyroid data point up the fact that sparsity of data is only one of several conditions which together determine the effect of overfitting avoidance. In particular, if class prevalences are far from equal, any of the well-known pruning methods may increase predictive accuracy even when training data is sparse.[7]

As argued in the introduction, when we recognize that overfitting avoidance is a form of bias, we naturally turn our attention away from pursuing "good" overfitting avoidance methods and toward a determination of where alternative methods are appropriate. This paper contributes by identifying the abundance of training data relative to the complexity of a target relationship as one important factor in this determination. It stops far short, however, of defining sparsity precisely or of telling us just how sparse data must be for particular pruning methods to degrade performance. This is an important area for future work.

One thing that may be stated emphatically even at this early juncture, however, is that it is hopeless to

[6]Results for the hepatitis data are omitted. In that case, there appears to be very little relationship between attributes and classes. A one-node tree is almost optimally predictive and, as might be expected when the the complexity of the target concept is so low, pruned trees are superior even for very small training sets.

[7]Ross Quinlan [personal communication] has produced a variant version of the Poisson generator which divides the unit interval as usual, but then assigns classes to the subintervals with unequal probabilities. In this case, pruning uniformly increases the performance of C4.5.

expect the training data itself to tell us whether it is sparse enough to make unpruned trees preferable. As argued at length in [Schaffer, 1992b], training data cannot tell us what bias is appropriate to use in interpreting it. In particular, the sparsity of data depends on the complexity of the true relationship underlying data generation; and it is not data but domain knowledge that can tell us how complex a relationship to expect.

Acknowledgements

Special thanks to Ross Quinlan for his efforts to replicate the results reported here and for pointing out important flaws in a draft version of the paper. Thanks also to Wray Buntine and Robert Holte for supporting the ideas advanced here.

References

Breiman, Leo; Friedman, Jerome; Olshen, Richard; and Stone, Charles 1984. *Classification and Regression Trees.* Wadsworth & Brooks, Pacific Grove, California.

Buntine, Wray and Caruana, Rich 1991. Introduction to IND and recursive partitioning. Technical Report FIA-91-28, RIACS and NASA Ames Research Center, Moffett Field, CA.

Cestnik, Bojan and Bratko, Ivan 1991. On estimating probabilities in tree pruning. In *Machine Learning, EWSL-91.* Springer-Verlag.

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

Mingers, John 1989. An empirical comparison of pruning methods for decision tree induction. *Machine Learning* 4(2):227–243.

Murphy, P. M. and Aha, D. W. 1992. UCI repository of machine learning databases [a machine-readable data repository]. Maintained at the Department of Information and Computer Science, University of California, Irvine, CA.

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

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

Schaffer, Cullen 1991. When does overfitting decrease prediction accuracy in induced decision trees and rule sets? In *Machine Learning, EWSL-91.* Springer-Verlag.

Schaffer, Cullen 1992a. Deconstructing the digit recognition problem. In *Machine Learning: Proceedings of the Ninth International Conference (ML92).* Morgan Kaufmann.

Schaffer, Cullen 1992b. Overfitting avoidance as bias. *Machine Learning.*