Error-Based Pruning of Decision Trees Grown on Very Large Data Sets Can Work!

Lawrence O. Hall,[1] Richard Collins,[1] Kevin W. Bowyer,[2] and Robert Banfield[1]

1. Department of Computer Science & Engineering
University of South Florida
Tampa, Florida 33620-5399

2. Computer Science & Engineering
384 Fitzpatrick Hall
Notre Dame, IN 46556
hall@csee.usf.edu, rbanfiel@csee.usf.edu, kwb@cse.nd.edu

Abstract

It has been asserted that, using traditional pruning methods, growing decision trees with increasingly larger amounts of training data will result in larger tree sizes even when accuracy does not increase. With regard to error-based pruning, the experimental data used to illustrate this assertion have apparently been obtained using the default setting for pruning strength; in particular, using the default certainty factor of 25 in the C4.5 decision tree implementation. We show that, in general, an appropriate setting of the certainty factor for error-based pruning will cause decision tree size to plateau when accuracy is not increasing with more training data.

1. Introduction

In general, a decision tree can be grown so as to have zero error on the training set. If there is any noise in the data set or it does not completely cover the decision space, then over fitting occurs and the tree needs to be pruned in order to generalize well to the test set. There are various approaches to pruning decision trees, including error-based pruning, reduced-error pruning, minimum description length pruning, and others [1,10]. One well-known element of machine learning folklore is that decision tree pruning methods generally do not prune hard enough. In particular, error-based pruning, which is a simple method that does not require a validation set, has been criticized on this count. For example, Esposito et al. performed an empirical study of decision-tree pruning methods and reported that error-based pruning (EBP) underprunes on all datasets that they tested – "… EBP performs well on average and shows a certain stability on different domains, but its bias toward underpruning presents some drawbacks …" [1].

More recently, Oates and Jensen have studied decision tree pruning for large data sets [2,3,4]. They also conclude that pruning methods generally do not work as desired, and summarize the problem as follows – "Despite the use of pruning algorithms to control tree growth, increasing the amount of data used to build a decision tree, even when there is no structure in the data, often yields a larger tree that is no more accurate than a tree built with fewer data" [4]. As one illustration of the problem, Oates and Jensen present a graph of results for tree size versus training set size using a synthetic training set with examples from two classes that have random labels. Their data show that tree size grows approximately linearly with training set size, regardless of whether error-based, reduced-error, or minimum-description-length pruning is used. They also present a modification to reduced-error pruning that at least partially addresses the problem [4].

Error-based pruning (EBP) uses the error that is made at a node of the tree on the training data in an estimate of the test set error at that node. EBP assumes

that the error rate follows a binomial distribution, and the *certainty factor* (CF) parameter then controls the pruning. The CF is used to estimate the upper limit of the probability that an error occurs over the population at a leaf. This is done by using the CF as a confidence limit for a binomial distribution. This entails the questionable assumption that the errors at a node with N examples are events in a sequence of trials. Predictions of how many errors would really be made at a leaf can then be made. The higher the CF, the more likely the current error rate is accepted and no pruning will be done (after all, the decision tree decided this was a good split with the given data). A lower CF means more errors than occurred in the train data will be predicted and hence there is more chance for pruning because we overestimate the error rate at the leaf. Thus a certainty factor of 100 indicates no pruning, and smaller values of the certainty factor indicate greater pruning because progressively more errors are predicted to occur at a leaf for the same number of training examples [5]. This is the pruning method used in, for example, C4.5 release 8 [5].

The default setting of the certainty factor in C.4.5 release 8 is 25. We show that when the certainty factor parameter for error-based pruning is appropriately set, the pathological behavior noted by Oates and Jensen disappears. Thus error-based pruning can in fact work appropriately for large or small datasets.

We do not explicitly address how to set the CF in this paper. Instead, an examination of whether any setting of the CF exists which will halt the growth in size of trees when there is no increase in accuracy. Current evaluations of error based pruning in the literature [1,2,4] appear to have only worked with the default certainly factor. The evaluation of error based pruning as inadequate appears commonly accepted (see [8] for example)

2. Experimental Results

2.1 "Structure-less" Data

One of the more striking results shown by Oates and Jensen involves the creation of decision trees with C4.5 for a family of "structure-less" training sets of difference sizes. This data consists of elements with "30 binary attributes and a binary class label, all with values assigned randomly from a uniform distribution" [4]. The appropriate result for this type of data would be a single-node tree that assigns elements the label of the most frequently occurring class. However, the results obtained with C4.5 using the default value for the certainty factor show that tree size grows linearly with the size of the training set. In other words, when

given a larger training set the tree becomes larger, even when accuracy cannot increase.
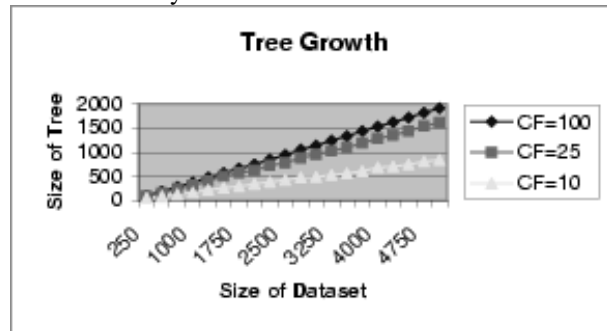
a)
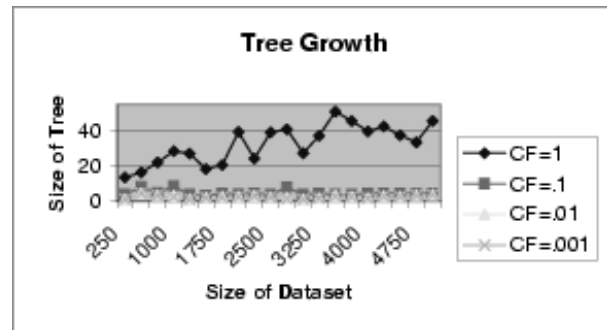
b)

Figure 1. Tree growth with error-based pruning and two class examples given a random class label. In a) we plot no pruning vs. default and cf = 10 and b) shows no growth for low cf.

Figure 1 shows results obtained with the same sort of structure-less data set used by Oates and Jensen. The training set size is varied from 250 to 5000, in increments of 250. Data is plotted as the C4.5 certainty parameter is varied across values of 100 (no pruning), 25 (the default), 10 in Figure 1a and 1, 0.1, 0.01 and 0.001 in Figure 1b. The curve for the default certainty factor value mimics the results presented by Oates and Jensen [4]. However, the family of curves clearly shows that the behavior depends on the value of the certainty factor. If the certainty factor is set as low as 0.01, then the average tree size varies between one (a "stump") and four over all training set sizes. That is, the tree size is minimal and constant, just as desired.

2.2 Structured Data

The structure-less data set is of course an extreme example. Performance over a number of real datasets may give a more useful view of practical performance. Therefore, experiments were also performed using the thirty-two data sets described in Table 1. Most of these data sets come from the UCI Machine Learning

repository [6]. One, the "Jones Protein Prediction" dataset, comes from the problem of predicting the secondary structure of proteins at each amino acid position. This particular dataset was used in constructing the classifier that won the Fourth Critical Assessment of Techniques for Protein Structure Prediction contest ("CASP-4") [7].

A ten-fold cross-validation experiment was done with C4.5 for each of the thirty-two data sets in Table 1. Each data set was divided into ten randomly selected one-tenths, and ten times C4.5 was trained on 90% of the data and tested on the other 10% of the data. The results recorded for each tree are the size of the tree, measured in number of nodes [9] and accuracy on the test set. The accuracy on the test set, and the average size and accuracy was computed across the ten test sets. This was done for each of fifteen different values of the certainty factor: 100, 90, 80, 70, 60, 50, 40, 30, 25, 20, 10, 1, 0.1, 0.01.
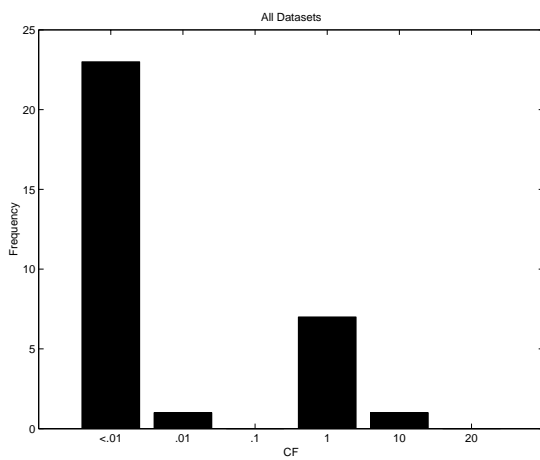
the error level of the tree at the default level (25) is tested. A one-sided paired t-test is used to compare the variations across the tenfold cross validation with the significance level set as $\alpha = 0.05$.

Rather than looking at certainty factor relative to the default value, we can also ask what value for the certainty factor would produce the lowest error on the test set. The histogram for this result appears in Figure 3. There are actually seven of the thirty-two datasets in Table 1 for which the certainty factor that results in the lowest error is greater than the default value, with the highest such setting being seventy. However, there are also eight datasets at which the best value of the certainty factor is 0.01 or lower. It should not be surprising that the accuracy of a decision tree technique is dependent on a parameter that controls the pruning strength. But it may be somewhat surprising that the ideal value of the parameter can vary so widely across different datasets, and that such a low certainty value can be appropriate so frequently.
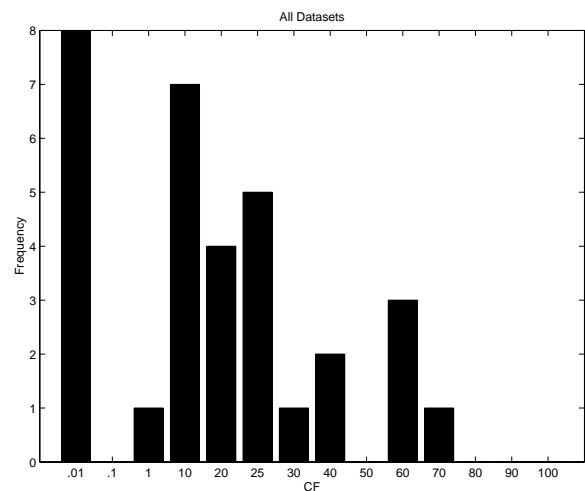

Figure 2 – Certainty Factor At Which Increase In Error Is Statistically Significant.

Figure 3 – Certainty Factor Value That Yields Smallest Error on the Test Set.

In all of the thirty-two data sets tested, the certainty factor can be set smaller than the default value, and so the size of the tree decreased, without a statistically significant increase in the error rate. Figure 2 shows a histogram of the smallest reasonable value of the certainty factor for the thirty-two data sets. Here, "smallest reasonable value" refers to the smallest value, less than the default of twenty-five, for which the error rate on the test set is not statistically significantly increased. In twenty-three of the thirty-two datasets, there was no statistically significant change in the error rate even with the certainty factor reduced to 0.01.

For each CF value, the null hypothesis that the error level of the tree at the given CF is less than or equal to

Next, we focus on the tree size, rather than tree accuracy. As the certainty factor varies from 25 to 0.01 across all thirty-two datasets, tree size decreases an average of 56.7% while error rate increases an average of 0.7%. The maximum decrease in size is 99.1%, which occurs for the German dataset, and the minimum decrease in tree size is zero, which occurred for the Glass and Mushroom datasets. The maximum decrease in error rate is 6.3%, which occurred for the Jones Protein Prediction Dataset, and the maximum increase in error rate was 10.35%, which occurs for the Tic Tac Toe dataset. The error increase is significant in only nine of the thirty-two datasets.

For each data set with CF= 0.01, the null hypothesis that the error level of the tree is less than or equal to

the error level of the tree at the default level (25) is tested. A one-sided paired t-test is used to compare the variations across the tenfold cross validation with the significance level set as $\alpha = 0.05$.

For datasets that represent "simple enough" pattern recognition problems, using a smaller certainty factor value does in fact cause tree size to become essentially constant as the training set size grows. Two examples of this are shown in Figure 4. A "simple enough" problem is one for which the classifier can achieve the greatest test accuracy possible using less than all of the available training data.
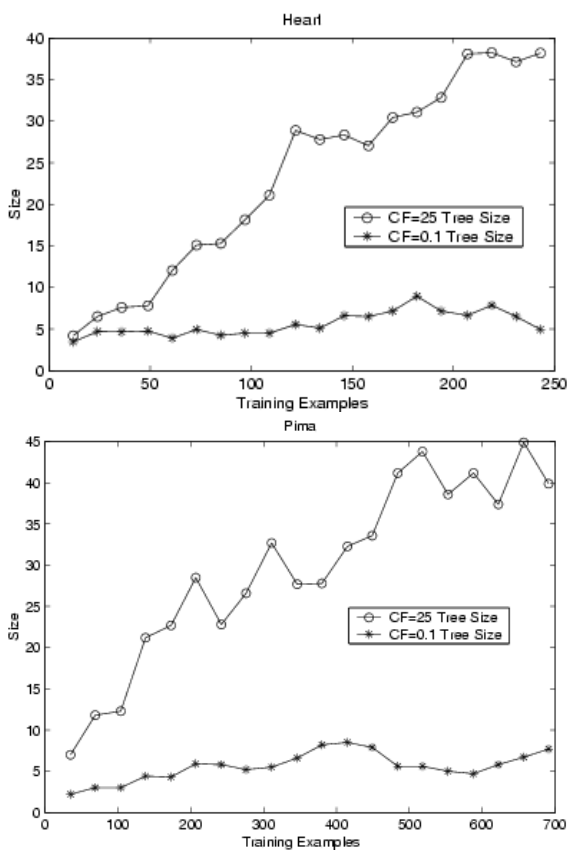

Figure 4 –Low Certainty Factor Can Give Constant Tree Size With Added Training Data.

3. Summary and Discussion

Error-based pruning is a simple method of pruning decision trees. It uses the training set error at a node and does not require a validation set. The degree of pruning is controlled by the certainty factor parameter. One objection to error-based pruning is that it has the general effect of under-pruning [1]. A related but more specific objection is that, for large datasets, error-based pruning results in trees that continue to increase in size

as the amount of training data increases, even when the resulting trees give no increased accuracy [4].

Our results show that these objections are valid only if one restricts attention to the default value for the certainty factor. When the certainty factor value is appropriately tuned for the data set, error-based pruning can give trees that are essentially constant in size regardless of the amount of training data. This generally requires values of the certainty factor much smaller than the default value in C4.5.

One could object to having to tune a parameter value for effective pruning, on the basis that, other things being equal, a parameter-free method is better. However, essentially all pruning methods are controlled by a parameter of some sort. For example, any method that requires a split of the available labeled data into a training set and a validation set effectively requires a parameter that is the split ratio and is vulnerable to an unfortunate group of examples in the validation set even with a good choice of split ratio. Thus an argument for one pruning method being better than another would have to be based on relative ease of tuning.

We have not addressed how to choose the CF. One possibility is to use a validation set. A learned tree can be pruned, successively, to different levels and tested with the most accurate tree retained. For large labeled data sets, there is no major drawback in using a validation set. For small data sets the current default CF appears adequate.

Error-based pruning has perhaps been too readily dismissed. For small datasets, it has the advantage that it does not require a split into train and validation data. For large datasets, as we have shown, it is able to produce trees that are essentially constant in size in the face of increasingly larger training sets. There is not yet a clear demonstration of a true problem with error-based pruning that is successfully addressed by some more sophisticated pruning technique.

4. References:

[1] F. Esposito, D. Malerba and G. Semeraro, "A Comparative Analysis of Methods For Pruning Decision Trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 476-491, 1997.

[2] T. Oates and D. Jensen, "Toward a theoretical understanding of why and when decision tree pruning algorithms fail'', *AAAI 99*, 1999.

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

[4] T. Oates and D. Jensen, "The effects of training set size on decision tree complexity," *Proceedings of the Fourteenth International Conference on Machine Learning*, 254-262, 1997.

[5] J.R Quinlan, C4.5: Programs for Machine Learning. San Mateo California: Morgan Kaufman, 1993.

[6] UCI Machine Learning Data Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html

[7] D.T. Jones, "Protein secondary structure prediction based on decision-specific scoring matrices'', *Journal of Molecular Biology* 292, 1999, 195-202.

[8] T. Elomaa and M. Kääriäinen, An Analysis of Reduced Error Pruning, Journal of Artifcial Intelligence Research 15 (2001) 163-187

[9] R.W. Collins, "Is Default Pruning Enough? A Study in C4.5 Error-Based Pruning," Master's Thesis, University of South Florida, April 2002.

[10] J. Mingers, "An Empirical Comparison of Pruning Methods for Decision Tree Induction'', Machine Learning, 3(4), pp. 227-243, 1989

Table 1. Description of real world data sets used.

Dataset Name	Data Instances	Continuous Features	Discrete Features	Classes	Majority Class Proportion
Protein Structure Prediction	209539	340	0	3	44.48%
Adult	32652	6	8	2	75.92%
Hyperthyroid	2800	7	22	4	92.14%
Australian	690	6	8	2	55.50%
Page Blocks	5473	10	0	5	89.77%
Breast Cancer Wisconsin	699	1	9	2	65.52%
Census Income	48845	6	8	2	54.12%
Cleveland	303	13	0	2	70.00%
German	1000	7	13	2	35.51%
Glass	214	10	0	7	55.56%
Heart	270	5	8	2	79.35%
Hepatitis	155	19	0	2	63.95%
Hungarian	294	13	0	2	64.10%
Ionoshpere	351	34	0	2	33.33%
Iris	150	4	0	3	33.33%
Kr vs Kp	3196	0	36	2	52.22%
Labor Negotiations	40	8	8	2	65.00%
LED	1000	0	7	10	10.90%
Letter	20000	16	0	26	4.07%
Long Beach	200	13	0	2	74.50%
Mushroom	8124	0	22	2	51.80%
PenDigits	10992	19	0	10	10.41%
Phoneme	5404	5	0	2	70.65%
Pima	768	8	0	2	65.10%
Promoter Gene	106	0	57	2	50.00%
Segmentation	2310	19	0	7	14.29%
Shuttle	43500	9	0	7	78.41%
Sick Euthyroid	3163	7	18	2	90.74%
Swiss	123	13	0	2	93.50%
Tic Tac Toe	958	0	9	2	65.34%
Congress Voting Record	435	0	16	2	61.38%
Congress Voting Record – Best Feature Removed	435	0	15	2	61.38%