

IS ERROR-BASED PRUNING REDEEMABLE?

LAWRENCE O. HALL*, KEVIN W. BOWYER†, ROBERT E. BANFIELD‡,
STEVEN ESCHRICH§ and RICHARD COLLINS

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

*†Computer Science & Engineering
384 Fitzpatrick Hall, Notre Dame, IN 46556*

**hall@csee.usf.edu*

†kwb@cse.nd.edu

‡rbanfiel@csee.usf.edu

§eschrich@csee.usf.edu

Received 1 February 2003

Accepted 1 April 2003

Error based pruning can be used to prune a decision tree and it does not require the use of validation data. It is implemented in the widely used C4.5 decision tree software. It uses a parameter, the certainty factor, that affects the size of the pruned tree. Several researchers have compared error based pruning with other approaches, and have shown results that suggest that error based pruning results in larger trees that give no increase in accuracy. They further suggest that as more data is added to the training set, the tree size after applying error based pruning continues to grow even though there is no increase in accuracy. It appears that these results were obtained with the default certainty factor value. Here, we show that varying the certainty factor allows significantly smaller trees to be obtained with minimal or no accuracy loss. Also, the growth of tree size with added data can be halted with an appropriate choice of certainty factor. Methods of determining the certainty factor are discussed for both small and large data sets. Experimental results support the conclusion that error based pruning can be used to produce appropriately sized trees with good accuracy when compared with reduced error pruning.

Keywords: Decision tree; pruning; error based pruning; reduced error pruning.

1. Introduction

In general, a decision tree can be grown so as to have zero error on the training set. Also, in general, “over-fitting” occurs and the tree needs to be pruned in order to generalize well on the test set. There are various approaches to pruning decision trees, including error-based pruning, reduced-error pruning, minimum description length pruning, and others.^{1,2,3,4,5} Several studies have examined cases in which decision tree pruning methods did not prune hard enough.^{6,7,8,9,3} That is, pruning left structure in the tree which did not contribute to its generalization ability and was, therefore, unnecessary.

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) under-prunes on all datasets that they tested — “... EBP performs well on average and shows a certain stability on different domains, but its bias toward under-pruning presents some drawbacks...”⁶

More recently, Oates and Jensen have studied decision tree pruning for large data sets.^{7,8,9} 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”.⁹ 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.⁷ Later, Frank³ argues that the modification leads to significantly higher error on 12 of 27 data sets though it does significantly decrease the tree size for all 27. The characterization of EBP as a method that inherently under-prunes has continued in a new analysis of reduced error pruning by Elomaa, *et al.*²

Current evaluations of error based pruning in the literature^{6,7,8,9,10,3} appear to have only worked with the default certainty factor. The default certainty factor as determined by Quinlan on a particular set of data sets is 25.¹ Experiments with parameter setting in C4.5 release 8 have been done by others.¹¹ However, the experimentation was done with the certainty factor and two other parameters. So, a search for a set of the best three parameters was undertaken. It did result in smaller trees when compared to C4.5 release 7 and they were maximizing accuracy in tuning the parameters. One cannot evaluate the effect of changing only the certainty factor from this experiment, but it does show that parameter tuning could be done automatically, and that it could result in smaller trees and increased accuracy.

The results presented in this paper are obtained using USFC4.5, which is our modification of C4.5 release 8. We show that when the certainty factor parameter for error-based pruning is appropriately set, the pathological behavior noted by Oates and Jensen disappears. We also provide a summary comparison with the results obtained by Esposito, showing that when the certainty factor is appropriately set, EBP does not under prune. We briefly discuss a methodology that can be utilized to appropriately set the certainty factor for small data sets. For large data sets, we show that a validation set can be used to set the certainty factor and we discuss how EBP relates to reduced error pruning (REP) in this case.

2. Error-Based Pruning

Error-based pruning considers the E errors among the N training examples at a leaf of the tree to give an estimate of the error probability for that node. The assumption is that these are E events in N independent trials which is, of course, not perfectly true. We want to know what the observed result tells us about the probability of an error over the entire population of examples that will end up at the leaf. Using the binomial theorem, confidence limits can be calculated for the probability of error for a given confidence level. The confidence level is the certainty factor parameter, CF , of C4.5. The upper limit of the probability is found. Given this value the predicted number of errors for each leaf of a test node being considered for pruning can be calculated by multiplying the number of examples at the leaf by the upper limit of the probability confidence limit. The predicted number of errors if a node was a leaf can be calculated from the observed number of errors after its leaves are collapsed. The leaves are pruned if the number of predicted errors after pruning is less than the sum of predicted errors across the leaves. The smaller the CF becomes the more certain we are that the confidence interval contains the true probability of error. That is, the confidence interval is wider, and the upper limit on the probability that a particular example is in error is higher, making an example more likely to be incorrect and hence more pruning will be done. With a $CF=100$ we have no confidence that the true error is in the interval and would simply take the observed error rate at the leaf.¹

In the implementation of EBP in USFC4.5, as in C4.5 release 8, a certainty factor of 100 still results in a fixed addition of 0.5 errors to the observed errors at the leaf. We have introduced a flag to prevent this.

EBP also performs subtree raising. In the case of subtree raising, an internal node can be replaced by the subtree of one of its children rather than a leaf.

In USFC4.5 the decision tree can be pruned on a validation set using EBP. In this case, the error estimates at each leaf come from the examples in the validation data that end up at that leaf. It is possible that a leaf will have no examples, in which case it is ignored when deciding to prune the subtree. If the entire subtree has no validation examples, then it will be pruned.

EBP on a validation set differs from REP pruning as defined by Elomaa² (which we believe is the most common implementation) in the utilization of the certainty factor to increase the error rate at a leaf, and in subtree raising. If the certainty factor is 100 and the automatic 0.5 error addition is turned off, then the only difference between the two decision tree pruning methods is subtree raising. USFC4.5 also allows subtree raising to be turned off in which case bottom up REP pruning will be applied to the tree with a certainty factor of 100 and no automatic error addition.

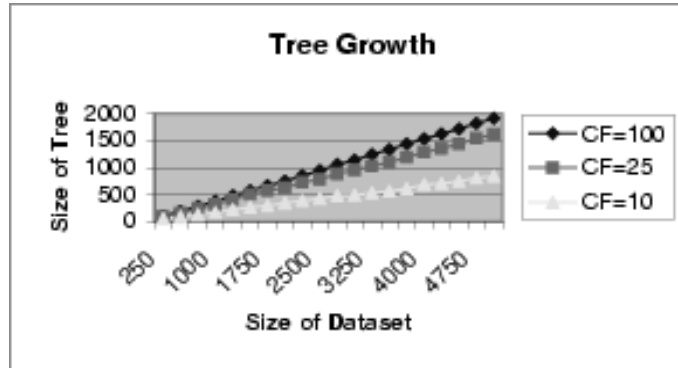
EBP will estimate that more errors are made than actually observed on the validation set for certainty factors lower than 100. Hence, if we ignore subtree raising, then EBP applied to a validation set will tend to prune more than REP for all $CF < 100$ and will certainly prune more of the tree for smaller CF s.

3. Experimental Results

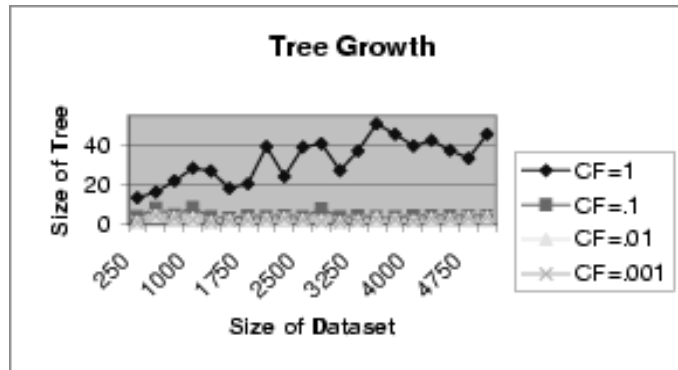
Results of experiments shown in this section answer the question, “Can EBP produce trees that do not grow with increasing training data unless their predictive accuracy is increasing?” We defer to the next section the question of how to automatically set the certainty factor to obtain pruned trees which are as small and general as possible while retaining predictive accuracy.

3.1. “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”.⁷ The appro-



(a)



(b)

Fig. 1. Tree growth with increasing training set size for two-class, “structure-less” data. For clarity, the data for lower certainty factors is plotted separately in part (b). Note that for low certainty factor, there is no growth in tree size.

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 though accuracy cannot increase.

Figure 1 shows results obtained with the same sort of structure-less data set as used by Oates and Jensen. The training set size is varied from 250 to 5000 examples, in increments of 250. Data is plotted as the CF is varied across values of 100 (minimal pruning), 25 (the default), and 10 in Figure 1a and 1, 0.1, 0.01 and 0.001 in Figure 1b. The curve for the default CF value mimics the results presented by Oates and Jensen.⁹ However, the family of curves clearly shows that the behavior depends on the CF value. If the CF 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.

3.2. Comparisons on 19 data sets used by Oates and Jensen

In previous work,⁹ 19 data sets taken from the UC Irvine repository¹² were used to examine how decision tree size changed as the training set size was increased. These experiments looked at increasing the training set size in increments of 5%. They compared five different pruning methods including EBP and REP. The other three were a minimum description length (MDL) pruning approach, cost-complexity pruning with the 1-SE rule (CCP1SE) and cost complexity pruning without the 1SE rule.⁴ The latter two pruning approaches come from the CART decision tree learning approach. Two runs of ten-fold cross validation were done at each level of training set size. Those runs were averaged to produce an accuracy versus size curve.

It was decided that accuracy ceased to grow when the mean of three adjacent accuracy estimates was no more than 1% less than the accuracy of the tree based on all available training data. The means were searched from the smallest three data set sizes to the largest three. They then recorded the percent kept for each of the pruning approaches to reach acceptable accuracy. Then they looked at the points on the tree size curve to the right of the training set size where accuracy ceased to grow, and used linear regression on those points to predict tree size from training set size. They report the significance of this regression fit (p and R^2). They also report the difference in size between the tree at which accuracy stops increasing and the final tree built on all the data as well as the difference in classification accuracy.

We have repeated these experiments with C4.5 release 8. The data sets used are highlighted in bold in Table 2. Our numbers vary somewhat with the default certainty factor, but are similar to those in previous work.⁹ In looking at the data from the experiments, we decided that it is more fair to look at the window of three means whose average is highest. In our experiments, we noted that it is possible to

Table 1. Summary of the effects of random data reduction for all of the pruning methods. The first five lines are repeated from Oates and Jensen 1997. EBP* is reporting %kept in the same way they did.

Pruning Method	% Kept < 100	$p < 0.1$	Mean R^2	Mean Δ size	Mean Δ accuracy
EBP	16	16	0.90	38.29	-0.14
REP	17	17	0.75	39.32	-0.32
MDL	18	17	0.88	44.03	-0.37
CCP1SE	19	10	0.62	30.11	-0.06
CCPOSE	17	11	0.58	47.40	-0.06
EBP* (CF = 0.001)	15	9	0.36	11.50	0.59
EBP (CF = 0.001)	10	6	0.28	9.54	0.69

decide that growth had stopped under the criteria above, but find that a particular window would have distinctly higher accuracy, sometimes higher than the tree built on all data. In Table 1 we repeat the results from⁹ and add two new lines for error based pruning with a certainty factor of 0.001. This low certainty factor will cause strong pruning. Utilizing our definition for percent kept, it is clear that EBP is actually doing the best among all approaches in not increasing tree size without increasing accuracy. We also show EBP* in the table for which we report percent kept as reported in the original experiment. EBP remained the best at preventing growth without accuracy increase.

Clearly, even with strong pruning, EBP does not stop tree growth without accuracy increase in all cases under the definitions used in the prior study.⁹ Let us examine more closely two of the data sets for which the most tree growth occurs without an accuracy increase. The accuracy and size at different certainty factor values for the trees incrementally built from the Cleveland data set are shown in Figure 2. Interestingly, the best accuracy comes from a subset of the data. However, we can see that the tree growth is quite small in numbers of nodes, even though it is about 50% (from an average of 3.7 nodes to 5.6 nodes with CF = 0.001). The hypothyroid data set is another example for which we plot accuracy and size at different certainty factor values as data is incrementally given to the tree building/pruning program. The graph is shown in Figure 3. In this case, it does appear that accuracy again peaks early. However, the overall trend is upward with more examples. While the tree grows 23%, from an average of 7 nodes to an average of 8.6, the number of nodes added is again small. Also, the accuracy does increase, though only by 0.15%, with CF = 0.001. In the case of both the Cleveland and the hypothyroid data set, at least with heavy pruning, the tree growth is actually quite mild.

3.3. *Effects of changing the certainty factor*

Performance over a number of real datasets may give a more useful view of practical performance when changing the certainty factor. Therefore, experiments were

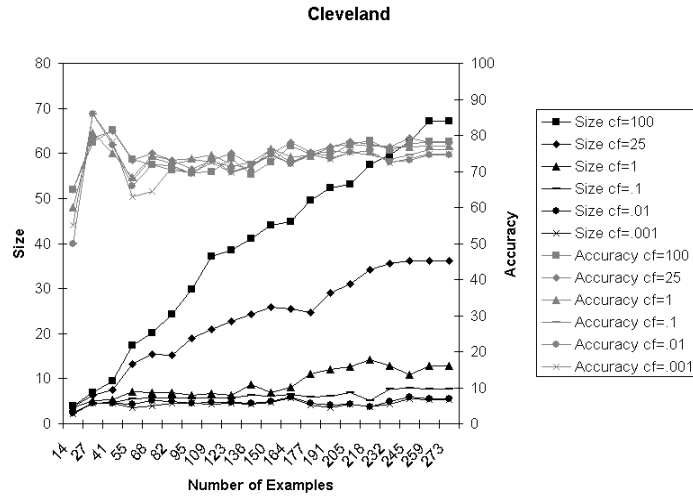


Fig. 2. Tree growth vs. accuracy with error-based pruning for the Cleveland data set.

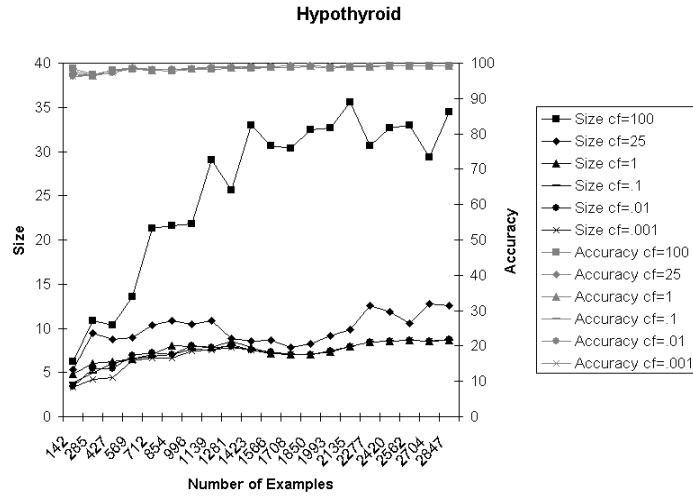


Fig. 3. Tree growth vs. accuracy with error-based pruning for the Hypothyroid data set.

also performed using the first thirty-two data sets described in Table 2. Most of these data sets come from the UCI Machine Learning repository.¹² One, the “Jones train1” 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”).¹³

Table 2. Description of real world data sets used.

Dataset Name	Data Instances	Continuous Features	Discrete Features	Classes	Majority Class Proportion
Jones train1	209539	315	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%
Ionosphere	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%
Breast Cancer	286	0	9	2	70.00%
Lymphography	148	2	17	2	54.70%

A ten-fold cross-validation experiment was done with C4.5 for each of the first thirty-two data sets in Table 2. 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 nodes¹⁴ and accuracy on the test set. The average accuracy on the test set and the average size were 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.

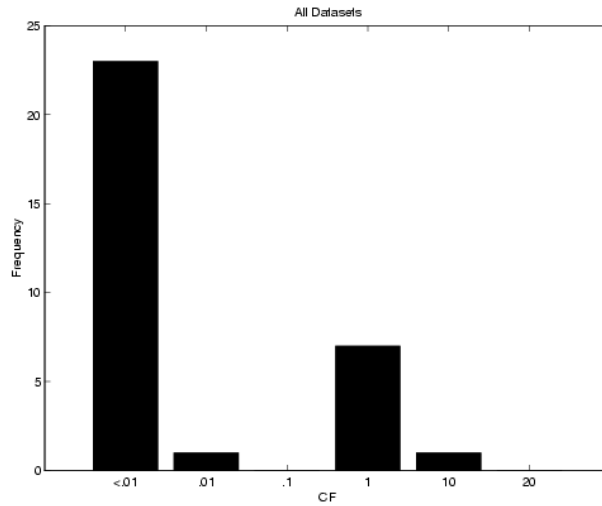


Fig. 4. Certainty factor at which increase in error is statistically significant.

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 any statistically significant decrease in accuracy. Figure 4 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 certainty factor of twenty-five, for which there is no statistically significant decrease in accuracy. In twenty-three of the thirty-two datasets, there was no statistically significant change in accuracy with the certainty factor reduced to 0.01.

For each CF value, we tested the null hypothesis that the error level of the tree at the given CF is less than or equal to the error level of the tree at the default level. A one-sided paired-t test was used to compare the variations across the tenfold cross validation with the significance level set as $\alpha = 0.05$.

Rather than looking at the certainty factor relative to the default value, we can also ask what value would produce the lowest error on the test set. The histogram for this result appears in Figure 5. There are actually seven of the thirty-two datasets in Table 2 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.

Next, we focus on the tree size, rather than tree accuracy. As the certainty factor decreases from 25 to 0.01 across all thirty-two datasets, tree size decreases

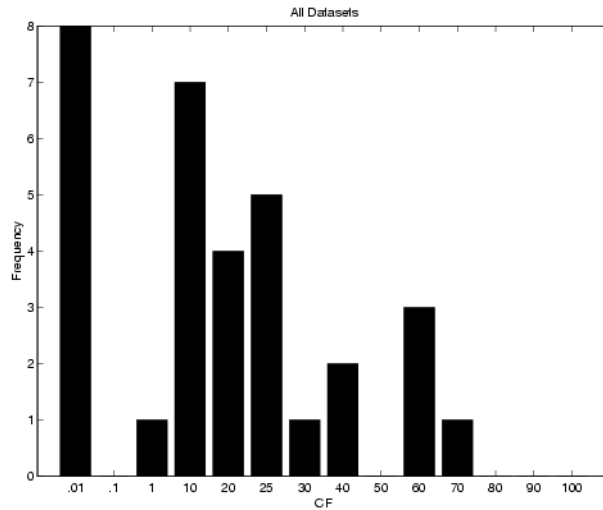


Fig. 5. Certainty factor value that yields smallest error on the test set.

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.

We also consider the change in accuracy going from $CF = 25$ to $CF = 0.01$. The maximum decrease in error rate is 8.3%, which occurred for the Jones train1 Dataset, and the maximum increase in error rate was 10.35%, which occurs for the Tic Tac Toe dataset. For each data set, we test the null hypothesis that accuracy with $CF = 0.01$ is less than or equal to the size with the default $CF (25)$. A one-sided paired t-test is used to compare the variations across the tenfold cross validation with the significance level set as $\alpha = 0.05$. The error increase is significant in nine of the thirty-two datasets.

Given the results presented in this subsection, it is clear that changing the certainty factor for EBP has a strong effect on both the size of the tree and the accuracy of the tree. It seems clear that the default choice of 25 is reasonable for a number of the data sets tested here, however a smaller certainty factor can often be used. The smaller value will result in a smaller tree which is often at least as accurate as the tree obtained with default pruning.

3.4. Comparison of results on under-pruning

Esposito *et al.* performed an empirical study of decision-tree pruning methods and reported that EBP under-prunes on all datasets that they tested.⁶ When assessing EBP, Esposito *et al.* define under-pruning as pruning a tree to a size that is larger than the “optimally pruned trained tree” (OPTT) and has a higher error on the test set. The OPTT is derived as follows:

- Grow a tree on 70% of the dataset using C4.5.
- Use reduced error pruning (REP) to prune the tree using the remaining 30% of the data as the prune set.
- Evaluate the tree using the same 30% used to prune the tree (with REP) as the test set.

Esposito *et al.* claim that this tree is the best subtree of the trained tree with respect to the test set. For this reason, all other trees created by pruning the trained tree are compared to the OPTT.

We experimented with 14 of the 15 data sets used in Esposito.⁶ We did not use LED-200 due to uncertainty in how to re-create the test set that they used. Our tree size for comparison was the average over a ten-fold cross validation with a certainty factor of 0.01. For 12 of the 14 data sets the tree size was smaller than or equal to the optimally pruned trained tree. Only for the Iris data set was our tree size larger, with potential significance. Our results indicate an average size of 5.4 nodes in the tree versus the 4 nodes per tree reported by Esposito. This is a three-class problem and we believe it is necessary, with binary splits on continuous values, to have at least five nodes in the tree so that there is a leaf for each of the three classes. Hence, we do not believe their reported result is correct. Table 3 contains the comparative results.

We conclude that the certainty factor of EBP can be lowered so that it does not under prune. It is true that the smallest tree is not always the most accurate tree. However, we are simply addressing the concern that this pruning method cannot be used to produce appropriately small trees.

Table 3. Average size and error rate for tree built on 90% of the data and tested on the remaining 10% (10 replications per mean value).

	CF = 25		CF = .01		Reported OPTT ⁶	
	Mean Size	Mean Error	Mean Size	Mean Error	Mean Size	Mean Error
Iris	8.4	6.00	5.4	6.67	4.00	4.42
Glass	11.0	2.88	11.0	2.88	15.08	28.31
Led-1000	49.6	26.70	23.8	27.80	22.04	25.31
Hypo	24.6	0.73	8.6	0.73	9.36	0.35
P-gene	21.8	25.00	5.0	22.00	10.00	16.50
Hepatitis	17.0	18.66	2.8	21.35	4.36	16.34
Cleveland	39.2	20.00	6.6	21.33	16.36	20.84
Hungary	17.0	20.32	3.4	19.30	9.64	17.27
Switzerland	1.0	6.66	1.0	6.66	1.16	5.73
Long Beach	18.6	23.00	1.2	26.00	4.92	23.13
Heart	37.3	25.16	4.5	27.01	45.96	17.71
Blocks	84.2	3.33	27.0	3.43	30.44	2.35
Pima	39.6	25.40	4.6	26.06	22.68	27.71
Australian	33.7	12.89	3.0	14.47	24.76	12.04

4. Choosing the Certainty Factor

We consider two cases for choosing the certainty factor for pruning, where the training set size either is or is not large enough to reasonably allow the use of a validation set. The solution we will present for small training sets requires more computational time per example and hence may itself help draw the dividing line between small and “large enough to use a validation set” training data sets.

4.1. *Choosing the certainty factor: Large data*

In the case that the training set is large enough that a validation set can be subtracted from it without an expectation that the accuracy of the tree built from the remaining data will be greatly lowered, one can simply use a validation set to decide how much to prune. The first thing to recognize is that a tree pruned at CF_2 can be obtained by pruning the tree pruned at CF_1 when $CF_1 > CF_2$. So, the search for the appropriate certainty factor would consist of choosing an initial certainty factor, CF_i , for pruning and then evaluating the resultant tree on the validation set to determine its accuracy. Next, choose a new certainty factor δ lower than the last and prune the pruned tree. Evaluate the resultant tree on the validation set. Continue creating new pruned trees until the stopping criterion is met. The most reasonable stopping criterion is a decrease in classification accuracy on the validation set.

The above approach requires a choice of initial certainty factor and the amount to change the certainty factor after each tree is built. To maximally search the space $\delta = 1$ is appropriate until a value of 1 is reached for the certainty factor. At that point, from our experiments the following values appear sensible, 0.1, 0.01, 0.001, and 0.0001. A larger value of delta may often be reasonable. However, it does not take very long to prune an already built tree. Certainly it requires much less time than building the tree.

4.2. *Choosing the certainty factor: Small data*

Kohavi^{15,16} discussed the use of wrappers for choosing the certainty factor for C4.5 decision tree pruning. The wrapper approach makes use of best-first search and cross validation to set parameters utilizing just training data. Frank³ also had success in choosing a parameter for an enhanced REP pruning algorithm using cross validation. In the case of small data sets where we do not want to set aside data for validation purposes, wrappers can be applied to determine the appropriate certainty factor. The approach would work as follows if we assume the use of five-fold cross validation as suggested by Kohavi.

Choose a certainty factor modification operator, call it δ . Choose an initial certainty factor, CF_i . Use fivefold cross validation to get an accuracy estimate from the training data. The accuracy will be the average accuracy over the five experiments. Generate successor certainty factors by $CF_s = CF_i \pm \delta$. For each

certainty factor value, do a fivefold cross validation to find an average accuracy and size. Put the average accuracy, size and the certainty factor value onto a list ordered by accuracy and size. Choose the first element of the list and create the successor states via successor certainty factors. Evaluate them and appropriately place them on the list. Continue this until a stopping criterion is met. The stopping criterion can be a decrease in accuracy upon the evaluation of n successor states.

This approach requires building five trees to evaluate each certainty factor value. However, if the training set is small this will not be too big a penalty. There are choices of how much to change the certainty factor and where to start and when to stop. Kohavi tried using multiple modification operators, for example one that changes the certainty factor by 1 together with one that changes the certainty factor by 5. So, the highest accuracy tree's certainty factor would be used to generate four new certainty factors for evaluation. Under this scheme, we would include values less than 1 for searching. So, we would use a list of certainty factor values $\{0.1, 0.01, 0.001, 0.0001\} \cup [1, 100]$. Now, given that the best certainty factor was 1 we would also evaluate trees created with a certainty factor of 2 and 0.1 if $\delta = 1$.

4.3. EBP evaluated on validation data

To illustrate how error based pruning can be applied to a large data set, we used the data from the domain of protein secondary structure prediction. We used the same training set of 209,529 examples and a test set (17,731) amino acids from a set of protein chains that were considered non homologous to the training set¹³ plus a separate validation set of 74,813 examples.

EBP was applied to the training data with seven different levels of pruning. The pruned tree was evaluated on the validation data. The tree that had the best performance on the validation data was the one for which the certainty factor was equal to 0.001. That particular tree also had the best performance, at 60.4% accuracy on the test data. The results are as one would expect with complete results shown in Table 4.

Table 4. Tree size and accuracy for a tree built on Jones train1 with evaluation after pruning on a validation set for each certainty factor and evaluation on the test data.

EBP Certainty Factor	Tree Size	Error on Validation Data	Error on Test Data
25	46,103	46.2%	47.9%
10	40,825	45.1%	46.5%
5	36,137	43.8%	45.2%
1	25,121	41.1%	42.3%
0.1	14,719	39.3%	40.7%
0.01	8,111	38.8%	40.0%
0.001	5,743	38.5%	39.6%

4.4. *EBP and REP pruning on a validation set*

The REP and EBP algorithms were compared on the Jones train1 data set. The results are shown in Table 5. For comparison purposes, we applied REP to the test set to get the optimally pruned trained tree⁶ which was 81.7% accurate on the test data and consisted of 24,365 nodes. It is significantly more accurate than any other tree we obtained.

REP applied to the validation data resulted in a tree of 28,077 nodes. EBP applied to the validation set with no certainty factor (no additional errors estimated at the leaves) and just subtree raising resulted in a much smaller tree of 15,105 nodes that also made 26 fewer errors. With a $CF = 50$, a tree of 9,755 nodes was obtained which was the most accurate on the test data at 61.2%. This experiment shows that EBP will prune more than REP when both are applied to a validation set, as expected. In this particular case, the smaller tree is actually more accurate on unseen test data. To decide how to set the certainty factor on this large data set, we would recommend having a pruning data set and a validation data set for evaluating the pruned trees.

Table 5. Tree size and accuracy for trees built on Jones train1 and pruned on a validation set with EBP and REP. Error is on the unseen test data.

Pruning Algorithm	Pruning Data set	Size	Test Set Error
REP	Test	24,365	18.3%
REP	Validation	27,255	39.9%
EBP (no CF)	Validation	15,105	39.8%
EBP (CF = 100)	Validation	14,213	39.6%
EBP (CF = 75)	Validation	13,277	39.3%
EBP (CF = 50)	Validation	9,755	38.8%

5. 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.^{6,10} 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.⁹

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 parameter tuning.

We have addressed how to choose the CF for small data sets. The wrapper approach¹⁵ to searching for the best certainty factor value through cross validation will not be overly expensive for small data set sizes. For larger data sets, where there is enough data to create a separate validation set, we have shown that a validation set can be effectively used to set the certainty factor. A learned tree can be pruned, successively, to different levels and tested with the most accurate tree retained, as was done with the Jones train1 data. For large labeled data sets, there is no major drawback in using a validation set. In absence of any information the current default value is reasonable. However, for most data sets smaller trees can be obtained with no decrease in accuracy by utilizing a smaller certainty factor.

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. As we have shown, EBP 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 technique.

Acknowledgments

This work was supported in part by the United States Department of Energy through the Sandia National Laboratories ASCI VIEWS Data Discovery Program, contract number DE-AC04-76DO00789 and and the National Science Foundation under grant EIA-0130768. Thanks to Divya Bhadoria who ran some of the experiments and Philip Kegelmeyer for his comments on the content of the paper.

References

- [1] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo California, 1993.
- [2] T. Elomaa and M. Kaariainen, An analysis of reduced error pruning. *Journal of Artificial Intelligence Research*, 15:163–187, 2001.
- [3] E. Frank, *Pruning Decision Trees and Lists*. PhD thesis, University of Waikato, Department of Computer Science, Hamilton, New Zealand, 2000.
- [4] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.
- [5] M. Kearns and Y. Mansour, Bottom-up decision tree pruning algorithm with near-

- optimal generalization. In *Proceedings of the 15th International Conference on Machine Learning*, pp. 269–277. Morgan Kaufmann, 1998.
- [6] F. Esposito, D. Malerba and G. Semeraro, A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, 1997.
 - [7] T. Oates and D. Jensen, Toward a theoretical understanding of why and when decision tree pruning algorithms fail. In *AAAI'99*, 1999.
 - [8] T. Oates and D. Jensen, Large datasets lead to overly complex models: an explanation and a solution. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 294–298, 1998.
 - [9] T. Oates and D. Jensen, The effects of training set size on decision tree complexity. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 254–262, 1997.
 - [10] L. A. Breslow and D. W. Aha, Simplifying decision trees: A survey. *Knowledge Engineering Review*, 12:1–40, 1997.
 - [11] L. A. Breslow and D. W. Aha, Comparing tree-simplification procedures. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pp. 67–74, 1997.
 - [12] C. J. Merz and P. M. Murphy, UCI repository of machine learning databases. Technical report, Univ. of CA., Irvine, Dept. of CIS, Irvine, CA, <http://www.ics.uci.edu/mlearn/MLRepository.html>, 1999.
 - [13] D. T. Jones, Protein secondary structure prediction based on decision-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 1999.
 - [14] R. W. Collins, Is default pruning enough? A study in C4.5 error-based pruning. Master's thesis, University of South Florida, CSE Dept., April 2002.
 - [15] R. Kohavi, *Wrappers for performance enhancement and oblivious decision graphs*. PhD thesis, Stanford University, Computer Science department, 1995. STAN-CS-TR-95-1560.
 - [16] R. Kohavi and G. H. John, Automatic parameter selection by minimizing estimated error. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 304–312, 1995.