Machine Learning, 4, 227-243 (1989)

An Empirical Comparison of Pruning Methods for Decision Tree Induction

JOHN MINGERS BSRCD@CU.WARWICK.AC.UK
School of Industrial and Business Studies, University of Warwick, Coventry CV4 7AL, England

Editor: Jaime Carbonell

Abstract. This paper compares five methods for pruning decision trees, developed from sets of examples. When used with uncertain rather than deterministic data, decision-tree induction involves three main stages—creating a complete tree able to classify all the training examples, pruning this tree to give statistical reliability, and processing the pruned tree to improve understandability. This paper concerns the second stage—pruning. It presents empirical comparisons of the five methods across several domains. The results show that three methods—critical value, error complexity and reduced error—perform well, while the other two may cause problems. They also show that there is no significant interaction between the creation and pruning methods.

Key Words: Decision trees, Knowledge acquisition, Uncertain data, Pruning

1. Introduction

Several approaches to inductive learning have been developed [Michalski, Carbonell & Mitchell, 1983, 1986; Bratko & Lavrac, 1987], and one of these involves the construction of decision trees. Based on initial work by Hunt, Marin & Stone [1966], Quinlan [1979, 1983b] developed the ID3 algorithm for deterministic problems such as chess endgames. At the same time, Breiman, Friedman, Olshen, & Stone [1984] were developing a similar approach to classification problems. This approach may be applied to knowledge acquisition for expert systems [Hart, 1985a, 1986; Michalski & Chilausky 1980], where an expert supplies examples and the resulting decision tree may be used in the formulation of rules. This is the context for the experiments described here.

Most recent work has focused on the use of such methods in domains where the data are not deterministic but uncertain [Quinlan, 1987; Niblett, 1987; Cestnik, et al., 1987; Kodratoff & Manago, 1987]. Uncertainty in the data may be due to noise in the measurements or to the presence of factors which cannot be measured. When used in this context, there are three phases to rule induction: first, creating an initial, large rule tree from the set of examples; second, pruning this tree to remove branches with little statistical validity; and third, processing the pruned tree to improve its understandability. Mingers [1989] compared several methods for tree creation in terms of the size and classification accuracy of the trees produced. The study concluded that there was little difference between the methods and that their use reduced the size of a tree rather than improving its accuracy. This paper deals with the pruning stage and reports an empirical comparison of the main

methods proposed so far. Section 2 outlines the basic Quinlan algorithm and five methods for pruning. Section 3 describes the data and experimental procedure, and Section 4 summarizes the results.

2. Inducing Decision Trees in Uncertain Domains

This section outlines problems encountered in dealing with uncertain data and presents the pruning methods that are to be evaluated.

2.1. The ID3 Algorithm and Uncertain Data

For a detailed description of the ID3 algorithm, see Quinlan [1979, 1983a, 1986]. Briefly, the approach begins with a set of examples (the training data) consisting of an attribute-value list. (Attributes are either numeric or symbolic.) Each example belongs to a particular known class. The aim is to develop a series of rules which will correctly classify further examples into one of these classes, when only the values of the attributes of the example are known. The algorithm examines each attribute in turn and calculates an information-theoretic measure of how well the attribute discriminates between the classes (a "goodness of split" measure). The best attribute is chosen, and the data are partitioned into subsets according to the values of that attribute. This process is recursively applied to each subset until all the examples are correctly classified. The result is a tree in which nodes represent attributes and branches represent possible attribute values or ranges of values. Terminal nodes (leaves) of the tree correspond to sets of examples, all of which are in the same class.

With deterministic data, an example in the training set can always be correctly classified from its known attributes. However, in many real problems there may be a degree of uncertainty present in the data. This uncertainty may arise from two different sources. The first is mis-measurement: for a variety of reasons, the value of an attribute or class may be incorrectly measured or may be missing. This may happen because of incorrect perception, measurement, recording, or transcription. I will refer to this source of uncertainty as noise. The second source of uncertainty is the occurrence of extraneous factors which are not recorded, but which affect the results. Thus the class of an example cannot be determined wholly from its recorded attributes. I will refer to this source as residual variation, and it is often of more significance than noise in real-world problems.

When ID3 classifies such data, the resulting tree tends to be very large. However, many of the branches will reflect chance occurrences in the particular data rather than representing underlying relationships. These are very unlikely to occur in further examples. Pruning methods identify the least reliable branches and remove them. In domains with high uncertainty, this often results in the removal of all but the first two or three levels of the tree. Pruning a tree will increase the number of classification errors made on the training data, but should decrease the error rate on independent test data.

2.2. Methods for Pruning Decision Trees

This study examines five of the principal methods for pruning decision trees. All the methods begin with a full tree developed from a set of training data.

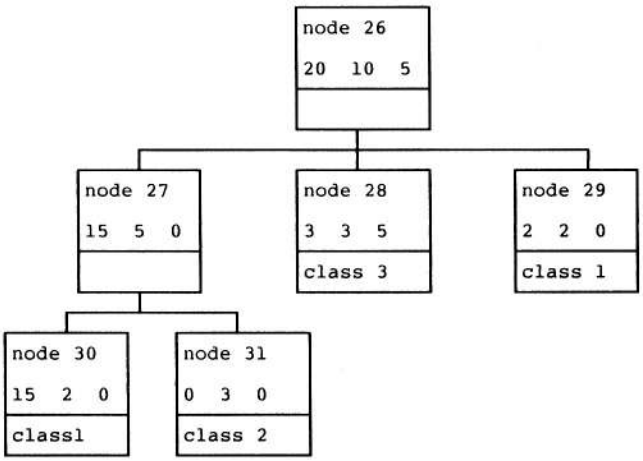

Figure 1. Example of a partially pruned sub-tree.

The methods are described in this section with the help of the sub-tree shown in Figure 1. This comes from a domain with three classes and 200 examples in the training set. Four leaves and two nodes from the full tree are shown. Some pruning has already occurred so that the leaves have examples from more than one class. At each node, three numbers show how many examples are in each of the three classes. At the leaves, the selected class is also shown.

2.2.1. Error-Complexity Pruning (Err-comp) Breiman et al. [1984] have developed a two-stage method, first generating a series of trees pruned by different amounts, and then selecting one of these by examining the number of classification errors each of them make with an independent data set. In pruning, the error-complexity method takes account of both the number of errors and the complexity (size) of the tree.

The method works as follows. Each node in the tree is the starting point for a sub-tree which will end with several leaves. Before pruning, the leaves will contain examples belonging to only one class, but, as pruning progresses, the remaining leaves will include examples from several different classes. When this happens, the examples at the leaf are examined and the leaf is allocated to the class which occurs most frequently. The error rate of a leaf is then the proportion of training examples which do not belong to that class. If the sub-tree is pruned, then the expected error rate is that of the starting node, which becomes a leaf. If the sub-tree is not pruned then the error rate is the average of the error rates at the leaves weighted by the number of examples at each leaf. With the training data, pruning will always lead to an increase in the error rate, and this increase is a measure of the worth of the sub-tree. Dividing this increase by the number of leaves in the sub-tree gives a measure of the reduction in error per leaf for that sub-tree. This is the error-complexity measure.

To illustrate, consider node 26 in Figure 1. In the notation of Breiman [1984]: t is a node, T is a sub-tree. The sub-tree rooted at node 26 has 4 leaves, $N_T = 4$. If this is pruned, node 26 becomes a leaf of class 1, and so 15 of the 35 examples are then wrongly classified.

Therefore, the error rate $r(t) = 15/35$. The proportion of data at t is $p(t) = 35/200$, so the error cost of node t is

$$R(t) = r(t)p(t) = \frac{15}{35} \times \frac{35}{200} = \frac{15}{200} .$$

If the node is not pruned, the error cost for the sub-tree is

$$R(T_t) = \Sigma R(i), \qquad \text{for } i = \text{sub-tree leaves}$$

$$= \frac{2}{17} \times \frac{17}{200} + 0 + \frac{6}{11} \times \frac{11}{200} + \frac{2}{4} \times \frac{4}{200} = \frac{10}{200} .$$

The complexity cost is the cost of one extra leaf in the tree, α. Then the total cost of the sub-tree is

$$R(T_t) + \alpha N_T$$

and of the node, if the sub-tree is pruned

$$R(t) + \alpha .$$

These are equal when

$$\alpha = \frac{R(t) - R(T_t)}{N_T - 1} = \frac{15/200 - 10/200}{4 - 1} = 5/600 . \tag{1}$$

α gives a measure of the value of the sub-tree, i.e., the reduction in error per leaf.[1]

The algorithm calculates α for each sub-tree (except the first) and selects the sub-tree with the smallest value of α for pruning. Repeating this process until there are no sub-trees left yields a series of increasingly pruned trees. The next step is to select one of these as the final tree. The criterion for selection of the final tree is the lowest mis-classification rate; however, this criterion cannot be based on the training set, since that would always give the unpruned tree as best. Instead, each of the pruned trees is used to classify an independent test data set. The number of mis-classifications made, as the size of the tree reduces, generally conforms to a standard pattern. A slow fall as the branches due to chance are removed is followed by a rapid rise once a large proportion of the tree has been pruned. A graph of the mis-classification rate against the number of leaves remaining in a tree is shown in Figure 2. The curve is very flat; thus, the selection of the particular tree which has the smallest mis-classification rate is somewhat arbitrary. Breiman's method, therefore, selects the smallest tree with a mis-classification rate within one standard error[2] of the minimum.

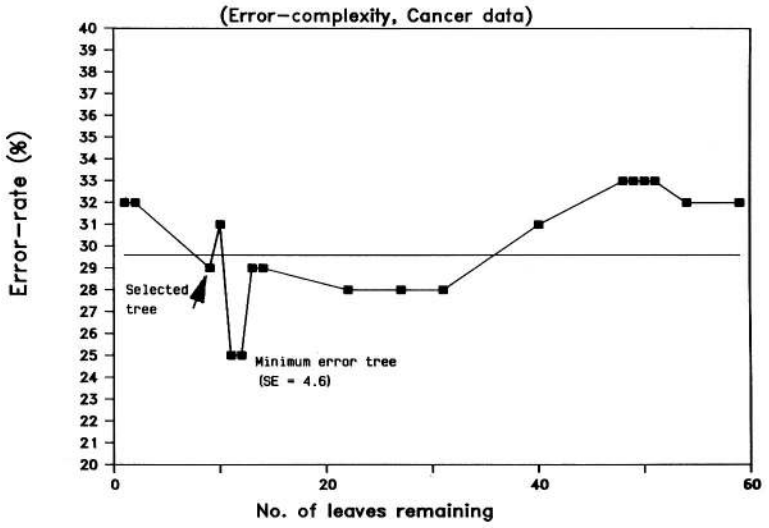

Figure 2. Reduction in error rate.

The standard error of the mis-classification rate (assuming a binomial distribution) is

$$SE = \sqrt{\frac{R \times (100 - R)}{N}} \tag{2}$$

where R = mis-classification rate of the pruned tree, and N = number of examples in the test data.

In Figure 2, the smallest mis-classification rate is 25% for a tree with 12 leaves left. The standard error of this tree is 4.6%. The smallest tree within one standard error has nine leaves and a mis-classification rate of 29%.

This pruning method is intuitively appealing because it takes into account both the classification error and the tree complexity with its measure of reduction in error per leaf. It also produces a selection of trees for the expert to study. It does, however, require a separate test set for selection purposes. Breiman, et al. [1984] show how their approach can be generalized to deal with unequal mis-classification costs and alternative prior probabilities. The first addresses the situation where mistakes in classification have different costs depending on the nature of the mis-classification. The second occurs when it is known that the distribution of classes is different in the general population.

2.2.2. Critical Value Pruning (Critical) Mingers' [1987a] method relies on estimating the importance or strength of a node from calculations done in the tree creation stage. As explained above, in creating the original tree, a goodness of split measure determines the attribute at a node. The value of the measure reflects how well the chosen attribute splits the data between classes at that node. This pruning method specifies a critical value and prunes those nodes which do not reach the critical value, unless a node further along the

branch does reach it. The larger the critical value selected, the greater the degree of pruning and the smaller the resulting tree. In practice, a series of pruned trees is generated using increasing critical values. A single tree can be chosen in the same way as for error-complexity pruning.

The particular critical value used depends on the measure used in creating the tree [Mingers, 1989]. For the measures used in this paper, critical values might range from 4 to 100 in steps of 5, although this will vary depending on the data set. Some creation measures rely on a probability rather than an absolute value, and, in this case, the critical value must also be a probability. Typically, values in the range 0.9500 − 0.9995 (95%–99.95%) are suitable for generating a set of pruned trees. A probability level of 95% implies that there is only a 5% chance of such a high value of the measure occurring at a node if there is actually no association between the particular attribute and the classes.

In the example, the value of the measure used (the G-Statistic) in creating node 26 is 15.38. We would only prune this node if the critical value being used were greater than this.

2.2.3. Minimum-Error Pruning (Min-err) Niblett and Bratko [1986] have developed a method to find the single tree which should, theoretically, give the minimum error rate when classifying independent sets of data.

Assume a set of data with k classes; assume also that we have observed n examples, of which the greatest number, n_c, are in class c. If we predict that all future examples will be in class c, what is the expected error rate—that is, proportion of wrong classifications?

Niblett and Bratko show that the expected error rate, E_k, is given by:

$$E_k = \frac{(n - n_c + k - 1)}{(n + k)} \tag{3}$$

Note that this assumes that all the classes are equally likely.

Using this measure, the method of pruning is as follows.

At each (non-terminal) node in the tree, calculate the expected error rate if that sub-tree is pruned so that the node becomes a leaf, using the numbers in each class at the node. Then calculate the expected error rate if the node is not pruned using the error rates for each branch, combined by weighting according to the proportion of observations along each branch. The procedure is recursive since the error rate for a branch cannot be calculated until you know if the branch itself is to be pruned. If pruning the node leads to a greater expected error rate, then keep the sub-tree; otherwise, prune it. The result should be a pruned tree that minimizes the expected error rate in classifying independent data.

For example: at node 27, pruning the sub-tree gives

$$E_k = \frac{20 - 15 + 3 - 1}{20 + 3} = 0.304$$

while not pruning it gives

$$E_k = \frac{17}{20}\left(\frac{17 - 15 + 3 - 1}{17 + 3}\right) + \frac{3}{20}\left(\frac{3 - 3 + 3 - 1}{3 + 3}\right) = 0.220\,.$$

The expected error from pruning is greater, so do not prune.

At node 26, if the sub-tree is pruned

$$E_k = \frac{35 - 20 + 3 - 1}{35 + 3} = 0.447 ;$$

if it is not pruned

$$E_k = \frac{20}{35} \times 0.220 + \frac{11}{35}\left[\frac{11 - 5 + 3 - 1}{11 + 3}\right] + \frac{4}{35}\left[\frac{4 - 2 + 3 - 1}{4 + 3}\right] = 0.370 .$$

The expected error from pruning is greater so do not prune. The result is that the expected error for the node 26 sub-tree is 37%.

Theoretically this is the ideal solution, since it minimizes the total expected error and does not require a separate testing set. However, there are several problems. First, the assumption of equally likely classes is seldom true in practice, although the effect of this is not clear. Second, the procedure produces only a single tree. This is a disadvantage in the context of expert systems, where it is helpful if several trees, pruned to different degrees, are available. Third, as the results will show, the number of classes strongly affects the degree of pruning, leading to unstable results.

2.2.4. Reduced-Error Pruning (Reduce) Quinlan [1987b] suggests a method which produces a series of pruned trees by using the test data directly, rather than using it only for selection of the best tree.

The method is as follows: Start with a complete tree and run the test data through it, noting the numbers occurring in each class at each node. For each non-leaf node, count the number of errors if the sub-tree is kept and the number if it becomes a leaf through pruning. The pruned node will often make fewer errors on the test data than the sub-tree makes. The difference between the numbers of errors (if positive) is a measure of the gain from pruning the sub-tree. From all the nodes, choose the one with the largest difference as the sub-tree to prune. Continue this process, including those nodes where the reduction is zero, until further pruning would increase the mis-classification rate. This produces the smallest version of the most accurate tree with respect to the test set.

There may be a number of sub-trees with the same (largest) difference. Quinlan [1987] does not specify which sub-tree to choose in this situation—for example, the largest or the smallest. Experiments show that the choice makes little difference in terms of classification accuracy; therefore, the largest is selected on the grounds that this reduces the number of iterations necessary to prune the tree completely.

This approach generates a set of trees, ending with the smallest minimum-error tree on the test data.

2.2.5. Pessimistic Error Pruning (Pessim) This is another method due to Quinlan [1986b], which aims to avoid the necessity of a separate test data set. As has been seen, the mis-classification rates produced by a tree on its training data are overly optimistic and, if used

for pruning, produce overly large trees. Quinlan suggests using the continuity correction for the binomial distribution to obtain a more realistic estimate of the mis-classification rate.

If $N(t)$ = number of training set examples at node t, and $e(t)$ = number of examples mis-classified at node t, then

$$r(t) = \frac{e(t)}{N(t)}$$

is an estimate of the mis-classification rate. The rate with the continuity correction is

$$r'(t) = \frac{e(t) + 1/2}{N(t)} . \tag{4}$$

For a sub-tree T_t the mis-classification rate will be

$$r(T_t) = \frac{\Sigma\, e(i)}{\Sigma\, N(i)}$$

where i covers the leaves of the sub-tree. Thus the corrected mis-classification rate will be

$$r'(T_t) = \frac{\Sigma\, (e(i) + 1/2)}{\Sigma\, N(i)} = \frac{\Sigma\, e(i) + N_T/2}{\Sigma\, N(i)} \tag{5}$$

where N_T is the number of leaves.

In (4) and (5), $N(t) = \Sigma N(i)$ as they refer to the same set of examples; therefore, the rates can be simplified to numbers of misclassifications:

$$n'(t) = e(t) + 1/2 \qquad \text{for a node}$$

$$n'(T_t) = \Sigma\, e(i) + N_T/2 \qquad \text{for a sub-tree.}$$

With the training data, the sub-tree will always make fewer errors than the corresponding node, but this is not so when the corrected figures are used, since they depend on the number of leaves, not just the number of errors. However, it is likely that even this corrected estimate of the number of mis-classifictions made by the sub-tree will be optimistic. So the algorithm only keeps the sub-tree if its corrected figure is more than one standard error (as defined earlier) better than the figure for the node.

The standard error for the number of mis-classifications is derived from that given earlier for the rate of mis-classifications:

$$SE(n'(T_t)) = \sqrt{\frac{n'(T_t) \times (N(t) - n'(T_t))}{N(t)}} \tag{6}$$

So Quinlan suggests pruning the sub-tree unless its corrected number of mis-classifications is lower than that for the node by at least one standard error.

The algorithm evaluates each node starting at the root of the tree. This means that it does not need to consider nodes that are in sub-trees which have already been pruned.

For the example, number of corrected mis-classifications at node 26, $n'(t) = 15 + 1/2 = 15.5$, and number of corrected mis-classifications for sub-tree, $n'(T_t) = (2 + 0 + 6 + 2) + 4/2 = 12.0$

$$SE = \sqrt{\frac{12 \times (35 - 12)}{35}} = 2.8 .$$

Since $12.0 + 2.8 = 14.8$, which is less than 15.5, the sub-tree should be kept and not pruned.

The statistical justification of this method is somewhat dubious. The continuity correction is not concerned with an overly optimistic mis-classification estimate, nor is it always added.[3] As used, the method is a crude, but nevertheless successful, heuristic, which compares the number of mis-classifications at a node with the number of leaves in the corresponding sub-tree. The continuity correction is incidental. The method does have advantages: it does not require a test data set, and it is very quick because it only has to make one pass and only looks at each node once. However, it does not produce a selection of trees.

3. Experimental Comparison of the Methods

The pruning methods described above were compared by applying them to five domains with widely differing characteristics. The measures used in the generation of the original tree were also varied. As explained earlier, several different measures of goodness of split are in use. Although earlier results [Mingers, 1989] suggest that they are equally effective in isolation, they may interact with the pruning method used. The four measures selected were the G-statistic (G-stat), the G-statistic with Marshall's correction (Marsh), the probability of G from the Chi-Square distribution (Prob), and Quinlan's gain-ratio measure (G-R).

3.1. Data Sets

The experiment used five data sets—four from natural domains and one constructed artificially.

B.A. Business Studies Degree Student Profiles (Babs) These data relate various attributes of each student on entry to the course to the final class of degree achieved. There are 186 observations with seven attributes—age (years), type of entry qualification (A-level,[4] BTEC Ordinary National Diploma, or some other), sex (male/female), number of O-levels, number of points at A-level (0–20), grade of maths O-level (A, B, C, Fail), and full-time employment before the course (yes/no). There are four possible classes of degree—first, upper second, lower second, or third. Three of the attributes are integer and four symbolic. There is no known noise, but many other factors affecting the results have not been (and probably could not be) measured; thus, the residual variation is high.

The Recurrence of Breast Cancer (Cancer) These data, containing 286 examples, derive from those used in Bratko and Kononenko [1986] and concern the recurrence of breast cancer. There are two classes (recur or not recur) and nine attributes, of which four are integer. These include age, tumor size, number of nodes, malignant (yes/no), age of menopause (< 60, $> = 60$, not occurred), breast (left, right), radiation treatment (yes/no), and area of breast (left, right, top, bottom, center). There are both missing data and residual variation.

Classifying Types of Iris (Iris) Kendall and Stewart [1976] use these data as a test of discriminant analysis. There are 150 examples of three different varieties of iris, with roughly equal numbers of each variety. The four integer attributes are measurements such as petal length and petal width, from which the examples can be classified. There is little noise or residual variation.

Recognizing LCD Display Digits (Digits) This is an artificial domain suggested by Breiman [1984]. A digit in a calculator display consists of seven lines, each of which may be on or off. Thus, there are ten classes (one for each digit) and seven binary-valued attributes (one for each line). Noise is introduced by assuming that a malfunction leads to a 10% chance of a line being incorrect. Such errors affect the attributes but not the class. Note that the chance of an example being completely correct is $0.9^7 = 0.48$. Three hundred cases were randomly generated.

Predicting Soccer Results (Footb) These data contain the results of 346 British league soccer matches. There are three classes (win, lose, draw) and five integer attributes measuring the past performance of the teams. There is little noise, but a high degree of residual variation.

These data sets vary widely in terms of degree of noisiness and residual variation, number of classes and attributes, mix of integer and discrete attributes, and the distribution of classes.

3.2. Criteria for Evaluation

There are two important criteria for evaluating a decision tree—size and accuracy.

Size. It is generally accepted that the fewer terms in a model the better (Occam's razor). This is particularly the case with statistical models, in which complexity will usually improve explanatory power on the training data, but worsen the predictive ability of the model on independent test data. Accordingly, one should attempt to minimize the size of the induced decision tree, as measured by the number of either nodes or of leaves. These two measures are related. For example, if the tree is strictly binary (i.e., every node has two branches), then number of leaves = (number of nodes + 1). If multi-valued attributes produce a tree that is not strictly binary, then the number of leaves will be greater than this. In this experiment, the number of leaves has been selected as the measure of size because it corresponds to the number of distinct rules contained within the decision tree.

Accuracy. This refers to the predictive ability of a decision-tree to classify an independent set of test data. It is measured by the error rate, i.e., the proportion of incorrect predictions that a tree makes on the test data. However, this is a crude measure since it does not reflect the accuracy of predictions for different classes within the data. Classes are not equally likely, and those with few examples are usually predicted badly.

3.3. Experimental Procedure

The experiments tested the five pruning methods with four different goodness-of-split measures on each of the five data sets.

Of each data set, 60% was randomly allocated to a training set, and the remainder was randomly split into two equal test sets. The first of these is needed by some of the pruning methods. This is either in selection of the final pruned tree (critical value and error-complexity) or in the actual pruning process (reduced-error method). The second test set is required for the accuracy measurements, since use of the same test set would bias the results in favor of these methods. As a single random split of the data may give unrepresentative results, the whole procedure is repeated nine times, giving nine different groups of training and testing data for each domain.

The results in Tables 1 and 2 show the number of leaves and the error rate (i.e., percentage of mis-classifications) for each combination of pruning method, goodness-of-split measure, and domain. They are averages across the nine sets. These results were then analyzed using Analysis of Variance (ANOVA)[5] to detect statistically significant differences between pruning methods or between measures and interactions between them.

4. Results

Table 1 shows the average number of leaves in the pruned trees for each combination, and Table 2 shows the average percentage mis-classification rate on the second test sets.

4.1. Size of Tree—Main Results

In Table 1, column six shows the average size of the tree in each domain before pruning. Comparing this with the size after pruning shows that, generally, the effects of pruning are very strong, reducing large trees to only a few leaves. Obtaining the correctly sized tree is very important for accuracy; the most common fault is not enough pruning. Wrongly sized trees account for many of the different error rates reported in Section 4.3.

Within the pruning methods, analysis of variance shows that there are significant differences between the pruning methods ($F = 589.6$) and between the measures ($F = 8.9$). There is also a significant interaction between pruning and domain ($F = 217.8$), but not between pruning and measure. Minimum-error pruning produces the largest trees, and error complexity and critical value produce the smallest. Particular interactions are analyzed in the next section.

Table 1. Average size of pruned tree (number of leaves).

Domain	Measure	Pruning Method					Size Unpruned
		Critical	Min-err	Err-comp	Pessim	Reduce	
Babs							
	G-stat	5.4	6.6	2.7	9.0	8.3	60.4
	Marsh	10.0	4.3	2.4	7.0	6.0	
	Prob	10.8	9.1	2.4	6.3	5.1	
	G-R	7.7	9.2	2.6	4.8	5.8	
Cancer							
	G-stat	2.3	44.1	3.3	13.4	7.3	52.8
	Marsh	4.3	43.9	2.7	7.8	7.3	
	Prob	4.2	43.2	2.6	12.2	8.1	
	G-R	4.1	40.4	3.0	13.7	11.9	
Iris							
	G-stat	3.3	4.7	3.2	3.9	3.0	6.9
	Marsh	3.9	4.8	3.2	3.9	3.6	
	Prob	6.7	9.3	6.3	6.8	6.7	
	G-R	3.3	4.7	3.2	3.8	3.0	
Digits							
	G-stat	14.6	13.7	12.7	12.8	18.3	56.6
	Marsh	14.8	12.9	13.3	13.3	17.6	
	Prob	17.1	20.4	14.6	17.3	19.9	
	G-R	11.8	13.7	10.9	12.3	15.2	
Footb							
	G-stat	2.8	58.4	2.2	38.0	8.6	85.7
	Marsh	6.6	58.6	2.1	34.6	10.2	
	Prob	4.0	58.7	3.0	36.6	8.9	
	G-R	5.1	51.9	2.8	33.4	9.8	
Total		142.8	512.6	99.2	290.9	184.6	

4.2. Size of Tree—Analysis

4.2.1. The Pruning Method/Domain Interaction Interaction between pruning method and domain is evident, with minimum-error and pessimistic pruning both producing larger trees with Footb and Cancer. The results for minimum-error pruning are particularly disturbing: not only are the trees often much larger than with the other methods, but there is no consistent pattern. Cancer and Footb are left almost entirely unpruned, while Babs, with a similar degree of noise and residual variation, is strongly pruned. The reason is that minimum-error pruning is very sensitive to the number of classes in the data. Cancer has two classes, Footb three, and Babs four, so the results suggest that the more classes the greater the degree of pruning. This was tested by specifying a greater number of classes in the Babs data than there are. Since the actual data had not changed, the results should be the same.

Table 2. Average error rate for pruned trees (% mis-classified)

		Pruning Method				
Domain	Measure	Critical	Min-err	Err-comp	Pessim	Reduce
Babs						
	G-stat	42.8	41.9	42.5	44.4	40.7
	Marsh	43.4	43.6	42.1	43.9	41.8
	Prob	44.2	43.7	42.1	40.1	40.2
	G-R	45.8	44.4	42.3	42.3	42.1
Cancer						
	G-stat	30.4	31.7	28.0	27.5	27.1
	Marsh	27.7	31.3	27.7	28.1	26.9
	Prob	29.0	32.6	28.5	28.1	27.2
	G-R	30.7	33.2	30.3	28.4	29.2
Iris						
	G-stat	7.2	7.2	7.2	7.2	7.2
	Marsh	8.9	7.8	7.6	8.8	7.6
	Prob	8.7	6.7	9.5	7.6	10.7
	G-R	7.5	6.8	7.5	7.5	7.5
Digits						
	G-stat	30.2	30.9	29.5	30.1	28.8
	Marsh	29.6	30.8	29.3	29.9	29.2
	Prob	31.2	29.8	30.6	29.9	29.9
	G-R	31.6	31.6	30.9	31.3	30.5
Footb						
	G-stat	49.5	54.4	47.3	55.5	48.6
	Marsh	49.1	52.9	46.9	53.7	47.5
	Prob	48.4	54.5	48.2	55.2	48.3
	G-R	48.7	57.3	48.7	56.2	50.9
Total		664.6	673.1	626.7	655.7	621.9

With Error-complexity and Critical this was indeed the case, but with Minimum-error the average number of leaves with four classes was 10.4 while with six classes it was 2.8.

Minimum-error is sensitive to the number of classes, k, because increasing k in equation (3) increases the calculated error rate. However, the increase is greater the smaller n (the number of examples) and n_c (the number of examples in the chosen class). When a node is split into a sub-tree, the result will be a set of branches with smaller numbers and, thus, relatively greater error rates. Increasing the number of classes magnifies this effect and biases minimum-error pruning against keeping the sub-tree. This results in smaller trees.[6] This behavior of minimum-error pruning is unfortunate in two ways. First, it leads to vastly different degrees of pruning with broadly similar data; second, it does not give the same results with data that is identical except for the number of classes.

Pessimistic pruning also gives results which vary significantly with the domain and generates large trees in the Cancer and Footb domains. The explanation of this, while not obvious, appears to be as follows. Pessimistic pruning compares the corrected number of mis-classifications at a node with the corrected number of mis-classifications for the sub-tree plus one standard error. The sub-tree usually makes no errors since the original tree can classify all the data (except for any contradictions) correctly. Therefore, the main variable on the right-hand side of (6) is the number of leaves N_T. Thus the comparison is really between the number of errors at a node and the number of leaves in the corresponding sub-tree. To produce a high degree of pruning, the major, early nodes in the tree must be pruned. For these nodes, the node error reflects the unpredictability of the data before analysis, and the number of leaves reflects the extent to which it can be successfully analyzed—i.e., the level of noise. Iris and Digit both have low levels of noise so that the node errors quickly reduce, leading to pruning. Footb and Cancer are much more uncertain; thus, the node errors do not reduce rapidly and major pruning does not occur so readily. Of the two, Footb is initially the least predictable and so has the highest node errors and is the most strongly affected.

4.2.2. Differences Between Measures The significant difference between the types of measure is chiefly attributable to Prob, which gives larger-than-average trees on Digit and Iris. The explanation is a problem with the algorithm for calculating probabilities. These two data sets have attributes that are so significant at the early levels (e.g., G-statistic values of 120 with 2 degrees of freedom) that the probability routine cannot calculate correct probabilities and so the best attribute is not chosen.

4.3. Accuracy—Main Results

The achievable accuracy differs markedly between domains, depending on their inherent uncertainty. Mingers [1989] compared the accuracy of pruned and unpruned trees and found that, for most domains, pruning improved accuracy by 20% to 25%. Here, it is the differences between pruning methods which are of interest. These are found to be statistically significant ($F = 30.4$). There is also a significant interaction between pruning and domain ($F = 16.6$), as well as some evidence of differences between the measures ($F = 8.4$).

Equally noteworthy are those differences and interactions which were not significant. The type of measure does not interact with the pruning method, so there is no difficulty in separating the two phases of creation and pruning. Also, the type of measure in independent of the domain, so certain measures do not work better in particular domains.

The totals in Table 2 reveal that, of the pruning methods considered, Minimum-error and Pessimistic pruning were the least accurate, while Reduced-error and Error-complexity were the most accurate. Minimum-error and Pessimistic are not worse in all domains, but are noticeably so with Footb and Cancer. This is because, as discussed in Section 4.2, they do not prune the trees sufficiently in these cases, so that accuracy on independent data is low. This accounts for the pruning/domain interaction.

The marginal difference between measures is due to Gain-ratio, which is marginally worse in all domains. This may, however, be a chance result, as it did not show up in the previous experiments using a larger number of measures [Mingers, 1989].

4.4. *Accuracy—Analysis*

As explained in Section 3.3, three of the pruning methods make use of a test data set, so the error rate on that data underestimates the true rate. A comparison of the performance on the two test data sets used in these experiments shows the extent of the underestimate. Table 3 shows the average error rate for each pruning method on test set one, which was used by three of the methods, and test set two which was not.

Table 3. Overall average error rate (%) on two test data sets.

	Reduce	Err-comp	Crit	Pessim	Min-err
Set 1	26.5	29.1	29.5	33.0	33.2
Set 2	31.1	31.3	32.2	32.8	33.7
Difference	+4.6	+2.2	+2.7	−0.2	+0.5

Reduced-error, which uses Set 1 in the pruning process, underestimates the true error rate by 17%, while Error-complexity and Critical, which only use Set 1 to select a pruned tree, underestimate it by about 7%. This clearly illustrates the importance of using completely independent test data when assessing accuracy.

Some of the results of the present study can be directly compared with Quinlan's [1987] figures, as the Digit domain and three pruning methods are common to both.

The first row of Table 4 shows Quinlan's results using an independent test set. The second row shows equivalent results from Table 2. The two sets of results are generally similar, although Quinlan's are slightly lower. This is because of the much larger data set, 1000 training and 500 testing examples, used by Quinlan. The third row shows the effect of using the same pruned trees on a new, larger test set of 1000 test examples. The error rates reduce slightly and are closer to Quinlan's.

Table 4. Comparison with Quinlan's results for digit data (% error rate).

	Err-comp	Pessim	Reduce
Quinlan	28.7	27.4	28.0
Mingers (G-Stat)	29.5	30.1	28.8
Mingers (G-Stat, 1000)	29.3	28.6	27.9

5. Conclusion

Pruning can improve the accuracy of induced decision trees by up to 25% in domains with noise and residual variation. These experiments compare five different pruning methods in terms of the size of the pruned tree and its accuracy, the two aspects being strongly related.

The main conclusions are as follows. First, there are significant differences between the methods. Minimum-error pruning is very sensitive to the number of classes in the data. It produces markedly different levels of pruning even on essentially the same set of data. As a result it is the least accurate method. It also fails to provide a set of trees for the expert to examine, which is considered a drawback. Pessimistic pruning is the most crude, but is certainly the quickest and does not need a separate test data set. It gave bad results on certain data sets and should be treated with caution. Critical value, error-complexity, and reduced-error methods all performed well, producing consistently low error rates over all the data sets. They all provide a set of trees, but all require a test data set. Critical value pruning requires the specification of initial and incremental values and can be slow if these are not chosen appropriately, but if used in conjunction with a probability measure, actual significance levels can be determined.

Second, there is no evidence of an interaction between the type of measure used in tree creation and the pruning method. There is also no interaction between the type of measure and the domain.

This paper has concentrated on a quantitative assessment of the pruned decision trees in terms of their size and accuracy over several contrasting sets of data. Future work will need to assess the quality of the rules produced. How understandable are they? Do they correspond to the expert's or users' ways of thinking and needs? Do they include rules for all the classes or only some of them? Is the form of knowledge representation too limited? The answer to these questions may be even more important than the quantitative criteria discussed in the present paper.

Acknowledgments

Thanks to Clare Morris, Andrew Martin, Jaime Carbonell, and anonymous referees for helpful comments on earlier drafts.

Notes

1. The formula for error-complexity given in Quinlan [1987] can be derived from the above.
2. The standard error is a common statistical measure of the reliability of a calculated result.
3. It is used in statistics when a continuous distribution, such as the normal, is used to approximate a discrete one, such as the binomial.
4. A-level and O-level are British national exams taken at ages 18 and 16 respectively. BTEC is the Business and Technician Education Council which validates national exams.
5. Strictly speaking, ANOVA assumes that all factors have equal variances, which is clearly not the case here. However, Cochran [1947] argues that ANOVA remains the best form of analysis in this situation.
6. Mathematically, the change in error rate with varying k can be found by differentiating:

$$\frac{dE_k}{dk} = \frac{n_c + 1}{(n + k)^2}$$

With n_c a constant proportion of n, it can be seen that the larger n the smaller the change in rate.

References

Bratko, I., and Konenko, I. (1986). Learning diagnostic rules from incomplete and noisy data. *Seminar on AI methods in statistics*. London Business School, England: Unicom Seminars Ltd.

Bratko, I., and Lavrac, N. (Eds.) (1987). *Pregress in machine learning*. England: Sigma Press.

Breiman, L., Freidman, J., Olshen, R., and Stone, C. (1984). *Classification and regression trees*. California: Wadsworth International.

Cestnik, G., Kononenko, I., and Bratko, I. (1987). ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In I. Bratko and N. Lavrac (Eds.), *Progress in machine learning*. England: Sigma Press.

Cochran, W. (1947). Some consequences when the assumptions for the Analysis of Variance are not satisfied. *Biometrica* 3, 22–38.

Hart, A. (1985a). The role of induction in knowledge elicitation. *Expert Systems*, 2, 24–28.

Hart, A. (1986). *Knowledge acquisition for expert systems*. London: Kogan Page.

Hunt, E., Marin, J., and Stone, P. (1966). *Experiments in induction*. New York: Academic Press.

Kendall, M., and Stewart, A. (1976). *The advanced theory of statistics* (Vol. 3). London: Griffen.

Kodratoff, Y., and Manago, M. (1987). Generalization and noise. *International Journal of Man-Machine Studies* 27, 181–204.

Konenko, I., Bratko, I., and Roskar, E. (1984). *Experiments in automatic learning of medical diagnostic rules*. (Technical report). Ljubljana, Yugoslavia: Jozef Stefan Institute.

Marshall, R. (1986). Partitioning methods for classification and decision making in medicine. *Statistics in Medicine*, 5, 517–526.

Michalski, R. S., and Chilausky, C. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4, 125–161.

Michalski, R. S., Carbonell, J., and Mitchell, T. (1983). *Machine learning: An artificial intelligence approach*. (Vol. 1). Los Altos: Morgan Kaurman.

Michalski, R. S., Carbonell, J., and Mitchell, T. (1983). *Machine learning: An artificial intelligence approach*. (Vol. 2). Los Altos: Morgan Kaufman.

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

Mingers, J. (1987b). Rule induction with statistical data—a comparison with multiple regression. *Journal of the Operational Research Society*, 38, 347–352.

Mingers, J. (1989). An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3, 319–342.

Niblett, T. Constructing decision trees in noisy domains. In I. Bratko and N. Lavrac (Eds.), *Progress in machine learning*. England: Sigma Press.

Quinlan, J. R. (1979). Discovering rules from large collections of examples: A case study. In D. Michie (Ed.), *Expert systems in the micro electronic age*. Edinburgh: Edinburgh University Press.

Quinlan, J. R. (1983). The effect of noise on concept learning. In R. S. Michalski, J. Carbonell, T. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. Los Altos: Morgan Kaufman.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess and games. In R. S. Michalski, J. Carbonell, T. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. Los Altos: Morgan Kaufman.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.

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