# Very Simple Classification Rules Perform Well on Most Commonly Used Datasets

Robert C. Holte (holte@csi.uottawa.ca)
Computer Science Department, University of Ottawa, Ottawa, Canada K1N 6N5

The classification rules induced by machine learning systems are judged by two criteria: their classification accuracy on an independent test set (henceforth "accuracy"), and their complexity. The relationship between these two criteria is, of course, of keen interest to the machine learning community.

There are in the literature some indications that very simple rules may achieve surprisingly high accuracy on many datasets. For example, Rendell occasionally remarks that many real world datasets have "few peaks (often just one)" and so are "easy to learn" (Rendell & Seshu, 1990, p.256). Similarly, Shavlik et al. (1991) report that, with certain qualifications, "the accuracy of the perceptron is hardly distinguishable from the more complicated learning algorithms" (p.134). Further evidence is provided by studies of pruning methods (e.g. Buntine & Niblett, 1992; Clark & Niblett, 1989; Mingers, 1989), where accuracy is rarely seen to decrease as pruning becomes more severe (for example, see Table 1)[1]. This is so even when rules are pruned to the extreme, as happened with the "Err-comp" pruning method in Mingers (1989). This method produced the most accurate decision trees, and in four of the five domains studied these trees had only 2 or 3 leaves (Mingers, 1989, pp. 238-239). Such small trees cannot test more than one or two attributes. The most compelling initial indication that very simple rules often perform well occurs in (Weiss et al., 1990). In 4 of the 5 datasets studied, classification rules involving 2 or fewer attributes outperformed more complex rules.

This paper reports the results of experiments measuring the performance of very simple rules on the datasets commonly used in machine learning research. The specific kind of rules examined in this paper, called "1−rules", are rules that classify an object on the basis of a single attribute (i.e. they are 1-level decision trees). Section 2 describes a system, called 1R, whose input is a set of training examples and whose output is a 1−rule. In an experimental comparison involving 16 commonly used datasets, 1R's rules are only a few percentage points less accurate, on most of the datasets, than the decision trees produced by C4 (Quinlan, 1986). Section 3 examines possible improvements to 1R's criterion for selecting rules. It defines an upper bound, called 1R*, on the accuracy that such improvements can produce. 1R* turns out to be very similar to the accuracy of C4's decision trees. This result has two implications. First, it indicates that simple modifications to 1R might produce a system competitive with C4, although more fundamental modifications are required in order to outperform C4. Second, this result suggests that it may be possible to use the performance of 1−rules to predict the performance of the more complex hypotheses produced by standard learning systems.

---

[1] conditions under which pruning leads to a decrease in accuracy have been investigated by Schaffer (1992; in press) and Fisher & Schlimmer (1988).

**TABLE 1.** Results of a typical experimental study (Buntine & Niblett, 1992). For each dataset the error rates of 4 systems are sorted in increasing order. Only 3 entries violate the rule that error rate increases as complexity (leaf count) increases: these are marked with an asterisk.

| Dataset | Error Rates | | | | Corresponding Leaf Counts | | | |
|---|---|---|---|---|---|---|---|---|
| BC | 27.2 | 28.5 | 28.7 | 29.7 | 6.0 | 9.3 | 10.2 | 25.4 |
| GL | 39.6 | 40.5 | 50.6 | 53.2 | 8.1 | 8.5 | 8.9 | 21.8 |
| HY | 0.95 | 1.01 | 1.27 | 7.44 | 4.8 | 5.0 | 5.8 | 34.0 |
| IR | 4.9 | 5.0 | 5.5 | 14.2 | 3.5 | 3.5 | 3.4* | 12.1 |
| LY | 24.0 | 24.3 | 24.4 | 32.3 | 7.5 | 7.7 | 8.2 | 15.5 |
| MU | 1.44 | 1.44 | 7.31 | 8.77 | 12.4 | 12.4 | 23.3 | 48.7 |
| VO | 4.5 | 4.6 | 11.8 | 15.6 | 5.1 | 5.2 | 12.4 | 22.9 |
| V1 | 12.8 | 13.0 | 15.1 | 15.6 | 8.9 | 9.4 | 13.0 | 22.9 |
| led | 32.9 | 33.2 | 33.8 | 38.2 | 13.0 | 13.1 | 13.3 | 19.4 |
| pole | 15.0 | 15.4 | 15.5 | 26.4 | 5.4 | 5.7 | 5.8 | 22.8 |
| tumor | 60.9 | 61.6 | 62.7 | 67.9 | 19.6 | 17.6* | 22.5 | 32.8 |
| xd6 | 22.06 | 22.14 | 22.17 | 31.86 | 14.8 | 14.9 | 14.8* | 20.1 |

Section 4 defines a practical prediction system based on 1−rule accuracy, compares its predictions to the accuracies of all learning systems reported in the literature, and discusses its uses. Section 5 considers the practical significance of these results, and sections 6 and 7 discuss the implications of the results for machine learning applications and research.

## 1. 1R — a program that learns 1−rules from examples

Program 1R is ordinary in most respects. It ranks attributes according to error rate (on the training set), as opposed to the entropy-based measures used in C4. It treats all numerically-valued attributes as continuous and uses a straightforward method to divide the range of values into several disjoint intervals. It handles missing values by treating "missing" as a legitimate value. Appendix A gives pseudocode for 1R.

In datasets with continuously-valued attributes there is a risk of overfitting. In dividing the continuous range of values into a finite number of intervals it is tempting to make each interval "pure", i.e. contain examples that are all of the same class. But just as overfitting may result from deepening a decision tree until all the leaves are pure, so too overfitting may result from subdividing an interval until all the subintervals are pure. To avoid this, 1R requires all intervals (except the rightmost) to contain more than a predefined number of examples in the same class. Based on the results in Holte et al. (1989), the threshold was set at 6 for all datasets except for the datasets with fewest examples (LA,SO) where the threshold was set at 3.

A similar difficulty sometimes arises with nominal attributes. For example, consider a dataset in which there is a nominal attribute that uniquely identifies each example, such as the name of a patient in a medical dataset. Using this attribute, one can build a 1−rule that classifies a given training set 100% correctly: needless to say, the rule will not perform well on an independent test set. Although this problem is uncommon, it did arise in two of the datasets in this study (GL,HO); the problematic attributes have been manually deleted from the datasets.

### 1.1 The Datasets Used for Experimental Comparison.

Sixteen datasets were used to compare 1R with C4, a state-of-the-art learning algorithm. Fourteen of the datasets were selected from the collection of datasets distributed by the machine learning group at the University of California at Irvine (see Appendix B). The selection includes many of the datasets most commonly used in machine learning research. In addition to these fourteen datasets, the study includes a 2-class version of GL (G2), and, following (Buntine & Niblett, 1992), a version of VO in which the "best" attribute has been deleted (V1).

Table 2 gives a brief description of the datasets: note that they exhibit a wide variety of characteristics. "Dataset" gives the two-letter name used to refer to the dataset. If there are more than 2 classes in a dataset, the number of classes is indicated in parentheses

**TABLE 2.** Datasets used in the experiments.
(blank entries represent 0's)

| Dataset | Size | Baseline Accuracy | Missing Values | Attributes ... number of distinct values | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | cont | 2 | 3 | 4 | 5 | 6 | >6 | TOTAL |
| BC | 286 | 70.3 | yes | | 3 | 2 | | 1 | 1 | 2 | 9 |
| CH | 3196 | 52.2 | no | | 35 | 1 | | | | | 36 |
| GL (6) | 214 | 35.5 | no | 9 | | | | | | | 9 |
| G2 | 163 | 53.4 | no | 9 | | | | | | | 9 |
| HD | 303 | 54.5 | yes | 5 | 3 | 3 | 2 | | | | 13 |
| HE | 155 | 79.4 | yes | 6 | 13 | | | | | | 19 |
| HO | 368 | 63.0 | yes | 7 | 2 | 5 | 5 | 2 | 1 | | 22 |
| HY | 3163 | 95.2 | yes | 7 | 18 | | | | | | 25 |
| IR (3) | 150 | 33.3 | no | 4 | | | | | | | 4 |
| LA | 57 | 64.9 | yes | 8 | 3 | 5 | | | | | 16 |
| LY (4) | 141 | 56.7 | no | 2 | 9 | 2 | 5 | | | | 18 |
| MU | 8124 | 51.8 | yes | | 5 | 1 | 5 | 1 | 2 | 7 | 22 |
| SE | 3163 | 90.7 | yes | 7 | 18 | | | | | | 25 |
| SO (4) | 47 | 36.2 | no | | 13 | 3 | 4 | | | 1 | 35 |
| VO | 435 | 61.4 | yes | | 16 | | | | | | 16 |
| V1 | 435 | 61.4 | yes | | 15 | | | | | | 15 |

after the name. "Size" gives the total number of examples in the dataset. "Baseline Accuracy" gives the percentage of examples in the most frequently occurring class in the dataset. "Missing Values" indicates whether there are any examples in the dataset for which the value of some attribute is unknown. The remaining columns indicate the number of attributes having a given number of values. To be counted, in Table 2, as continuous (column entitled "cont") an attribute must have more than 6 numerical values. The total number of attributes in a dataset is given in the rightmost column. The total is the sum of the other "Attributes" columns plus the number of attributes in the dataset for which all examples have the same value. For example, in the SO dataset there are 13 attributes having 2 values, 3 attributes having 3 values, 4 attributes having values 4 values, and 1 attribute having more than 6 (non-numeric) values. This accounts for 21 of the 35 attributes in the dataset: the other 14 attributes have the same value in every example.

## 1.2 Experiment #1. Comparison of 1R and C4.

The version of C4 used in these experiments is C4.5 as distributed in May 1990. The default settings of all parameters were used, except that windowing was turned off. The accuracies of C4 and 1R on a dataset are computed in the usual way, namely:

1. randomly split the dataset into two parts, a training set (2/3 of the dataset) and a test set.
2. Using the training set alone, generate a rule.
3. Measure the accuracy of the rule on the test set.
4. Repeat 1-3 25 times and average the results.

The results of this experiment are given in Table 3.

**TABLE 3.** Results of Experiment #1 — Classification Accuracy.
1R — average accuracy on the test set of the 1−rule produced by 1R.
C4 — average accuracy on the test set of the pruned tree produced by C4

| | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | BC | CH | GL | G2 | HD | HE | HO | HY |
| 1R | 68.7 | 67.6 | 53.8 | 72.9 | 73.4 | 76.3 | 81.0 | 97.2 |
| C4 | 72.0 | 99.2 | 63.2 | 74.3 | 73.6 | 81.2 | 83.6 | 99.1 |

| | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | IR | LA | LY | MU | SE | SO | VO | V1 |
| 1R | 93.5 | 71.5 | 70.7 | 98.4 | 95.0 | 81.0 | 95.2 | 86.8 |
| C4 | 93.8 | 77.2 | 77.5 | 100.0 | 97.7 | 97.5 | 95.6 | 89.4 |

### 1.3 Discussion of Experiment #1.

On average 1R's accuracy is 5.7 percentage points lower than C4's. However, this average is quite misleading: on 12 of the 16 datasets, the difference between 1R's accuracy and C4's is less than the average. This skewness is caused by the two datasets (CH,SO) on which 1R's accuracy is extremely poor compared to C4's. On the other 14 datasets, 1R's accuracy is an average of 3.1 percentage points lower than C4's. On half the datasets, 1R's accuracy is within 2.6 percentage points of C4's. To summarize these results in general terms, one would say that on most of the datasets studied 1R's accuracy is about 3 percentage points lower than C4's.

These results raise two related questions:
(1) why was C4's accuracy not much greater than 1R's on most of the datasets ?
(2) is there anything special about the CH and SO datasets that caused 1R to perform so poorly ?

Considering question (1), there is no evidence that C4 missed opportunities to exploit additional complexity in order to improve its accuracy: C4's pruned trees were the same accuracy as its unpruned ones (not shown). It is possible that C4 is overfitting, i.e., that slightly less complex decision trees might have been more accurate, but this possibility has been explored only partially. Experiments were run in which C4 was forced to build 1−rules. These 1−rules were never more accurate than the pruned trees C4 would normally have produced: C4 is therefore correct in not pruning to the extreme. In fact, a survey of the literature reveals that C4's performance on these datasets is better than most learning systems (see Appendix C for details and section 4 for a discussion of this survey).

If the answer to question (1) lies not in the C4 algorithm, it must lie in the datasets themselves. It may simply be a fact that on these particular datasets 1−rules are almost as accurate as more complex rules. For example, on 2 datasets (BC,HE), few learning systems have succeeded in finding rules of any kind whose accuracy exceeds the baseline accuracy by more than 2 percentage points (see Appendix C).[2] On a few datasets (IR, for example) C4 prunes its decision tree almost to a 1−rule, a clear indication that, on these datasets, additional complexity does not improve accuracy. Section 6 examines in detail the complexity of C4's rules.

Turning to question (2), there is a characteristic of the CH and SO datasets that is a potential source of difficulty for a 1−rule learner. In these datasets there is only one attribute having more values than there are classes. In CH there are 2 classes, and there is one attribute having 3 values, and 35 attributes having 2 values. In SO there are 4 classes, and there is one attribute having 7 values, 4 attributes having 4 values, and 30 attributes having fewer than 4 values. By contrast, in almost all the other datasets there are continuous attributes (which can be divided into as many intervals as necessary) or several attributes having more values than there are classes.

---

[2] Clark & Boswell (1991) offers some discussion of this phenomenon.

To see why this characteristic can cause 1–rules to have unusually low accuracies, consider an extreme example — the soybean dataset used in Michalski & Chilausky (1980). In this dataset there are 15 classes, and there is one attribute having 10 values,[3] one attribute having 7 values, and 33 other attributes having 5 or fewer values. Assuming the attribute with 10 values perfectly separates the examples in the 10 largest classes, a 1–rule based on this attribute would achieve 86% accuracy. This is 11 percentage points lower than the accuracy of the complex rules reported in Michalski & Chilausky (1980). If this attribute turns out to be a poor classifier, the next best accuracy possible by a 1–rule is 76%, which happens only if the 7-valued attribute perfectly separates the examples of the 7 largest classes. The accuracy of 1–rules based on 5-valued attributes is 66% or less on this dataset. Of course, more complex rules can separate the examples in all of the classes, and one would expect them to clearly outperform 1–rules on datasets such as this.

This characteristic is thus an indication that 1–rules might perform poorly. However, one must not conclude that 1–rules will always perform poorly on datasets having this characteristic: VO and V1 provide examples to the contrary. In fact, on half the datasets, the number of leaves in 1R's rules is within 1 of the number of classes, as the following table shows.

| | BC | CH | GL | G2 | HD | HE | HO | HY | IR | LA | LY | MU | SE | SO | VO | V1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # leaves | 7 | 2 | 4 | 4 | 4 | 3 | 3 | 5 | 3 | 4 | 3 | 9 | 5 | 4 | 3 | 3 |
| # classes | 2 | 2 | 6 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 4 | 2 | 2 | 4 | 2 | 2 |

The numbers in this table include the leaf for "missing" providing it is non-empty. This is the reason that there are 3 leaves for the VO dataset even though all the attributes have 2 values. In the LY dataset two of the four classes have very few examples, so relatively high accuracy can be achieved with fewer leaves than classes.

If the poor performance of 1R on CH and SO is to be explained as a consequence of the datasets having only one attribute with more values than there are classes, it is then necessary to address the question, "why did 1R perform well on several datasets also having this property ?". The answer to this question, like the answer to question (1), may be that it is simply a fact about these particular datasets that classes and the values of some attributes are almost in 1-1 correspondence.

[3] In the version of this dataset in the Irvine collection, this attribute has only 4 values.

## 3. An Upper Bound on Improvements to 1R's Selection Criterion

Given a dataset, 1R generates its output, a 1−rule, in 2 steps. First it constructs a relatively small set of candidate rules (one for each attribute) and then it selects one of these rules. This 2-step pattern is typical of many learning systems. For example, C4 consists of two similar steps: first it constructs a large decision tree, and then, in the pruning step, it selects one of the subtrees of the tree constructed in the first step.

In any such 2-step system it is straightforward to compute an upper bound on the accuracy that can be achieved by optimizing the selection step. This is done by simply bypassing the selection step altogether and measuring the accuracy (on the test set) of all the rules available for selection. The maximum of these accuracies is the accuracy that would be achieved by the optimal selection method. Of course, in practice one is constrained to use selection methods that do not have access to the final test set, so it may not be possible to achieve the optimal accuracy. Thus, the optimal accuracy is an upper bound on the accuracy that could be achieved by improving the selection step of the system being studied.

There are at least two important uses of an upper bound computed in this way. First, if the system's current performance is close to the upper bound on all available datasets, then it will be impossible to experimentally detect improvements to the selection step. For example, before doing a large-scale study of various pruning methods, such as (Mingers,1989), it would have been useful to compute the upper bound on accuracy achievable by any pruning method. Such a study may have indicated that there was little room for variation amongst all possible pruning methods on the datasets being considered.

The second important use of this upper bound is in comparing two systems. If the upper bound on accuracy of one system, S1, is less than the actual accuracy of another system, S2, then the only variations of S1 that can possibly outperform S2 are ones whose first step is different than S1's. This is the use made of the upper bound in this section: the following experiment was undertaken to determine if modifications to 1R's selection step could possibly result in 1R equalling or exceeding C4's performance.

### 3.1 Experiment #2.

An upper bound on the accuracy achievable by optimizing 1R's selection step is computed as follows.

1. randomly split the dataset into two parts, a training set and a test set.
2. Using the training set alone, generate a set of rules.
3. Measure the **highest** accuracy of all the generated rules on the test set.
4. repeat 1-3 25 times and average the results.

The same training/testing sets were used as in experiment #1. The results of this experiment are given in table 4. For ease of reference, the upper bound is given the name 1R*.

**TABLE 4.** Results of Experiment #2 — Classification Accuracy.
1R, C4 — as in Table 3.
1R* — the highest accuracy on the test set of all the rules constructed by 1R with greater than baseline accuracy of the training set. This is an upper bound on the accuracy achievable by optimizing 1R's selection step.

| | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | BC | CH | GL | G2 | HD | HE | HO | HY |
| 1R | 68.7 | 67.6 | 53.8 | 72.9 | 73.4 | 76.3 | 81.0 | 97.2 |
| 1R* | 72.5 | 69.2 | 56.4 | 77.0 | 78.0 | 85.1 | 81.2 | 97.2 |
| C4 | 72.0 | 99.2 | 63.2 | 74.3 | 73.6 | 81.2 | 83.6 | 99.1 |

| | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | IR | LA | LY | MU | SE | SO | VO | V1 |
| 1R | 93.5 | 71.5 | 70.7 | 98.4 | 95.0 | 81.0 | 95.2 | 86.8 |
| 1R* | 95.9 | 87.4 | 77.3 | 98.4 | 95.0 | 87.0 | 95.2 | 87.9 |
| C4 | 93.8 | 77.2 | 77.5 | 100.0 | 97.7 | 97.5 | 95.6 | 89.4 |

### 3.2 Discussion of Experiment #2.

1R's accuracy cannot exceed 1R* because the rule selected by 1R is in the set of rules whose accuracies are used to compute 1R*. On average, 1R's accuracy is 3.6 percentage points lower than 1R*. On 5 datasets the difference in accuracies is negligible and on a further 5 datasets the difference is not large (3.8 percentage points or less). Bearing in mind that 1R* is a rather optimistic upper bound, one may conclude that changes to 1R's selection criterion will produce only modest improvement in accuracy on most of the datasets in this study.

The difference between C4's accuracy and 1R* is not particularly large on most of the datasets in this study. On two-thirds (10) of the datasets the difference is 2.7 percentage points or less. On average, 1R* is 2.1 percentage points less than C4's accuracy, and only 0.28 less if the CH dataset is ignored. On half of the datasets, 1R* is higher than or negligibly lower than C4's accuracy. For these reasons, one may conclude that the most accurate 1-rule constructed by 1R is, on almost all the datasets studied, about the same accuracy as C4's decision tree.

This result has two main consequences. First, it shows that the accuracy of 1–rules can be used to predict the accuracy of C4's decision trees. Section 4 develops a fast predictor, based on 1–rule accuracy, and discusses several different uses of such a predictor. Secondly, this result shows that 1R is failing to select the most accurate of the 1-rules it is constructng. With an improved selection criterion 1R might be competitive, as a learning system, with C4 (except on datasets such as CH). On the other hand, it is certain that however the selection criterion is improved, 1R will never significantly outperform C4. If C4 is to be surpassed on most datasets by a 1−rule learning system, changes of a more fundamental nature are required.

## 4. Using 1−rules to predict the accuracy of complex rules.

An ideal predictor would be a system that made a single, rapid pass over the given dataset and produced an accuracy comparable to C4's on the dataset. A natural candidate is 1R itself, using the whole dataset for both training and testing. 1Rw is defined to be the accuracy computed in this way:

1. Run program 1R with the whole dataset as a training set to generate a rule (called the W-rule).
2. 1Rw is the accuracy of the W-rule on the whole dataset.

Table 5 shows the value of 1Rw for the datasets in this study.

A careful comparison of 1Rw with C4's accuracy involves two steps. The first step is to use a statistical test (a two-tailed t-test) to evaluate the difference in accuracy on each individual dataset. This test computes the probability that the observed difference between 1Rw and C4's accuracy is due to sampling: "confidence" is 1 minus this probability. Unless confidence is very high, one may conclude that there is no significant difference between 1Rw and C4's accuracy on the dataset. If confidence is very high, one proceeds with the second step of the comparison in which the magnitude of the differences is considered. This step is necessary because significance tests are not directly concerned with magnitudes: very small differences can be highly significant. For

**TABLE 5.** 1Rw measured on the datasets.
C4 — as in Table 4.
1Rw — highest accuracy of the 1−rules produced when the whole dataset is used by 1R for both training and testing.

| | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | BC | CH | GL | G2 | HD | HE | HO | HY |
| C4 | 72.0 | 99.2 | 63.2 | 74.3 | 73.6 | 81.2 | 83.6 | 99.1 |
| 1Rw | 72.7 | 68.3 | 62.2 | 78.5 | 76.6 | 84.5 | 81.5 | 98.0 |

| | Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | IR | LA | LY | MU | SE | SO | VO | V1 |
| C4 | 93.8 | 77.2 | 77.5 | 100.0 | 97.7 | 97.5 | 95.6 | 89.4 |
| 1Rw | 96.0 | 84.2 | 75.7 | 98.5 | 95.0 | 87.2 | 95.6 | 87.4 |

example, the difference between C4's accuracy and 1Rw on the MU dataset, although it is one of the smallest in magnitude, is much more significant than the difference on any other dataset.

The results of the t-tests are as follows (see appendix D for details). The differences between C4's accuracy and 1Rw on the BC, GL, and VO datasets are not significant. The difference on the LY dataset is significant with 95% confidence. The differences on all other datasets are significant with greater than 99% confidence, i.e., the probability of observing differences of these magnitudes, if C4's accuracy is in fact equal to 1Rw, is less than .01.

The difference between C4's accuracy and 1Rw, although statistically significant, is not particularly large on most of the datasets in this study. On three quarters of the datasets the absolute difference is 3.3 percentage points or less. On average, 1Rw is 1.9 percentage points less than C4's accuracy, and only 0.007 less if the CH dataset is ignored. For these reasons, one may conclude that 1Rw is a good predictor of C4's accuracy on almost all the datasets studied.

### 4.1 1Rw as a Predictor of Accuracy of Other Machine Learning Systems.

In order to evaluate 1Rw as a predictor of the accuracy of machine learning systems in general, the machine learning literature was scanned for results on the datasets used in this study[4]. Appendix C lists the results that were found in ths survey. The G2, HO, and SO datasets do not appear in Appendix C because there are no reported results concerning them. A detailed comparison of the results for each dataset is impossible, because the results were obtained under different experimental conditions. Nevertheless, a general assessment of 1Rw as a predictor of accuracy can be made by comparing it on each dataset to the median of the accuracies for that dataset reported in the literature. 1Rw is very highly correlated with the medians, having a correlation coefficient (r) of 99% (if CH is ignored, 77% if CH is included). By fitting a line to this median-vs-1Rw data, one obtains a simple means of predicting medians given 1Rw. If this is done, the predicted value differs from the actual value by only 1.3 percentage points on average (if CH is ignored).

[4] It is not always possible to be certain that a dataset described in the literature is identical to the dataset with the same name on which 1Rw was measured. The survey includes all results for which there is no evidence that the datasets differ.

## 4.2 Uses of 1Rw

Predictors of accuracy, such 1Rw, or of relative accuracy, such as Fisher's measure of "attribute dependence" (Fisher (1987); Fisher & Schlimmer (1988)) are informative measurements to make on a dataset: they can be used in a variety of ways.

The most obvious use of 1Rw is as a benchmark accuracy for learning systems, i.e., as a standard against which to compare new results. The current benchmark is baseline accuracy, the percentage of examples in a dataset in the most frequently occurring class. For most datasets baseline accuracy is relatively low, and therefore not a useful benchmark. 1Rw is only slightly more expensive to compute and it is often a very challenging benchmark.

Alternatively, one can measure 1Rw before applying a learning algorithm to a dataset, in order to obtain a quick estimate of the accuracy that learned rules will have. This estimate could be compared to the accuracy required in the given circumstances. An estimated accuracy that is lower than the required accuracy is an indication that learning might not produce a rule of the required accuracy. In this case, the dataset should be "improved" by collecting or creating additional attributes for each example (e.g. compare V1 and VO), or reducing the number of classes (e.g. compare GL and G2), or in some other way changing the representation. In constructive induction systems (Rendell & Seshu, 1990) 1Rw is a natural method for evaluating new attributes, or even whole new representations (Saxena, 1989).

# 5. The Practical Significance of the Experimental Results

The preceding experiments show that most of the examples in most of the datasets studied can be classified correctly by very simple rules. The practical significance of this observation hinges on whether or not the procedures and datasets that have been used in the experiments — which are the standard procedures and datasets in machine learning — faithfully reflect the conditions that arise in practice. Of particular concern are the datasets. One does not intuitively expect "real" classification problems to be solved by very simple rules. Consequently, one may doubt if the datasets used in this study are "representative" of the datasets that actually arise in practice.

It is true that many of the classification problems that arise in practice do not have simple solutions. Rendell & Seshu (1990) call such problems "hard". The best known hard classification problem is protein structure prediction, in which the secondary structure of a protein must be predicted given a description of the protein as an amino acid sequence. Another well-known hard problem is the classification of a chess position as won or lost, given a description of the position in terms of "low level" features. The machine learning techniques developed for "easy" classification problems are, by definition, of limited use for hard classification problems: the development of techniques appropriate to hard problems is a challenging, and relatively new branch of machine learning.

However, the fact that some real problems are hard does not imply that all real problems are hard. A dataset faithfully represents a real problem providing it satisfies two conditions. First, the dataset must have been drawn from a real-life domain, as opposed to having been constructed artificially. All the datasets in this study satisfy this requirement. Secondly, the particular examples in the dataset and the attributes used to describe them, must be typical of the examples and attributes that naturally arise in the domain. That is, the datasets must not have been specially "engineered" by the machine learning community to make them "easy". The CH dataset does not satisfy this condition: its attributes were engineered explicitly for ID3 by a chess expert working with a version of ID3 built specially for this purpose (Shapiro, 1987, pp.71-73). Indeed, the development of CH was a case study of one particular technique ("structured induction") for transforming a hard classification problem into an easy one.

Thus, the practical significance of the present study, and other studies based on these datasets, reduces to the question: are the examples and attributes in these datasets natural or have they been specially engineered by the machine community learning (as in CH) in order to make induction easy ? The evidence pertaining to this question varies from dataset to dataset.

For 6 datasets (HY,LA,MU,SE,VO,V1) the process by which the dataset was created from the raw data is sufficiently well documented[5] that it can confidently be asserted that these datasets faithfully represent real problems. The only instance of data adaptation that is mentioned is in connection with the congressional voting data (VO, V1). In the original form, there were 9 possible positions a congressman could take towards a given bill. In the dataset, some of these possibilities are combined so that there are only three possible values for each attribute. The grouping is a natural one, and not one specially contrived to improve the results of learning.

For 3 datasets (BC,HO,LY), the creation of the dataset involved some "cleaning" of the raw data. The nature of this "cleaning" is not described in detail, but there is no suggestion that that it involves anything other than the normal activities involved in rendering a heterogeneous collection of records into a uniform structure suitable for machine learning experiments.[6] Thus there is no reason to doubt that these datasets faithfully represent real classification problems.

The preceding datasets are adaptations, involving minimal changes, of data that had already been collected for a purpose other than machine learning. The SO dataset is different, in that it was created for the purpose of machine learning. The account given of the creation of this dataset (pp.134-136, Michalski & Chilausky, 1980)) mentions two criteria for selecting attributes: (1) each attribute must be measurable by a layman, and (2) the dataset must include the attributes used in the expert system that was developed for comparison with the induced classification rules. The account suggests that the development of the expert system involved iterative refinement. Although this account does not explicitly comment on the extent to which the attributes evolved during the expert system development, it not unreasonable to suppose that the attributes in this dataset have been engineered, or at least selected, to ensure that accurate classification is

---

[5] HY, SE: Quinlan et al. (1986). LA: Bergadano et al. (1992). MU, VO (V1): Schlimmer (1987).

[6] HO: McLeish & Cecile (1990). BC, LY: Cestnik et al. (1987). For LY, some grouping of values was done, as in VO.

possible with relatively simple rules.

On the creation of the remaining datasets (GL, G2, HD, HE, IR) there is no published information.

In summary, only 2 of the datasets in this study may reasonably be judged to have been specially engineered by the machine learning community to be "easy". Ironically, it is these two datasets on which 1R performs most poorly. 9 of the datasets are known to be, or are very likely to be, representative of problems that naturally arise in practice. Although these datasets do not represent the entire range of "real" problems (e.g. they do not represent "hard" problems), the number and diversity of the datasets indicates that they represent a class of problems that often arises. The next two sections examine the role of very simple classification rules in machine learning applications and research within this class of problems.

## 6. Accuracy versus Complexity in 1R and C4

The preceding sections have established that there are a significant number of realistic datasets on which 1−rules are only slightly less accurate (3.1 percentage points) than the complex rules created by C4 and other machine learning systems. In order to get insight into the tradeoff between accuracy and complexity, the complexity of C4's trees was measured in experiment #1. The results are given in the following table.

| | BC | CH | GL | G2 | HD | HE | HO | HY | IR | LA | LY | MU | SE | SO | VO | V1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mx | 4 | 13 | 12 | 9 | 7 | 7 | 4 | 7 | 4 | 3 | 5 | 6 | 9 | 3 | 5 | 7 |
| dc | 0.9 | 4.5 | 5.1 | 3.7 | 2.7 | 2.2 | 1.6 | 1.2 | 1.9 | 1.4 | 2.2 | 1.6 | 1.5 | 2.0 | 1.4 | 2.2 |
| %>2 | 2 | 59 | 87 | 80 | 58 | 26 | 18 | 6 | 24 | 5 | 36 | 8 | 13 | 18 | 6 | 30 |

The "mx" row gives the maximum depth of the pruned trees built by C4 on each dataset. Maximum depth corresponds to the number of attributes measured to classify an example in the worst case. On average, the maximum depth of C4's trees is 6.6, compared to 1 for 1−rules. Maximum depth is usually regarded as an underestimate of the true complexity of a decision tree because it does not take into account the complexity due to the tree's shape. For this reason, researchers normally define complexity as the number of leaves or nodes in a tree. By this measure, C4's trees are much more complex than 1−rules.

Maximum depth, or number of leaves, are measures of the "static complexity" of a decision tree. However, considerations such as the speed of classification, or cost of measuring the attributes used during classification (Tan & Schlimmer, 1990), are dynamic properties of a tree which are not accurately reflected by static complexity. The dynamic complexity of a rule can be defined as the average number of attributes measured in classifying an example. The dynamic complexity of C4's pruned trees is given in the "dc" row of the table.[7] On datasets where C4's trees involve continuous attributes (GL, G2, and IR, for example), dynamic complexity is artificially high because C4 transforms these into binary attributes instead of N-ary attributes (N > 2). C4's dynamic complexity, averaged over the 16 datasets, is 2.3, compared to 1 for 1−rules. Furthermore, there is considerable variance in the number of attributes measured by C4's decision trees: on some datasets C4's dynamic complexity is considerably greater than 2.3, and in most datasets there are some examples, sometimes many, for which C4's decision trees measure more than 2.3 attributes. To illustrate the latter kind of variation, the third row in the table ("%>2") indicates the percentage of examples in each dataset for which classification by C4's decision trees involves measuring 3 or more attributes.

Thus in 1R and C4 there is a perfect symmetry in the relationship between accuracy and dynamic complexity. 1R's rules are always very simple, usually a little less accurate, and occasionally much less accurate. C4's rules are more accurate, a little less simple on average, and occasionally much less simple.

These differences between 1R and C4 have practical implications. In practice, different applications have different demands in terms of accuracy and static and dynamic complexity. Depending on these demands, either 1R or C4 will be the more appropriate learning system for the application. For example, C4 is appropriate for applications that demand the highest possible accuracy, regardless of complexity[8]. And, 1R is appropriate for applications in which static complexity is of paramount importance: for example, applications in which the classification process is required to be comprehensible to the user. In applications where simplicity and accuracy are equally important, the symmetry between 1R and C4 means that the two systems are equally appropriate.

## 7. The "Simplicity First" Research Methodology

One goal of machine learning research is to improve both the simplicity and accuracy of the rules produced by machine learning systems. In pursuit of this goal, the research community has historically followed a research methodology whose main premise is that a learning system should search in very large hypothesis spaces containing, among other things, very complex hypotheses. According to this "traditional" methodology, progress in machine learning occurs as researchers invent better heuristics for navigating in these spaces towards simple, accurate hypotheses.

The results of preceding sections do not lend support to the premise of the traditional methodology. Complex hypotheses need not be considered for datasets in which most examples can be classified correctly on the basis of 1 or 2 attributes. An alternative, "simplicity first" methodology begins with the opposite premise: a learning system should search in a relatively small space containing only simple hypotheses. Because the space is small, navigating in it is not a major problem. In this methodology, progress occurs as

[7] The dynamic complexity of the BC dataset is less than 1 because C4 occasionally produces a decision tree that consists of nothing but a leaf (all examples are classified the same without testing a single attribute).

[8] for example, the space shuttle application described in Catlett (1991a)

researchers invent ways to expand the search space to include slightly more complex hypotheses that rectify specific deficiencies.

The experiment with 1R* nicely illustrates how a researcher proceeds according to the "simplicity first" methodology. That experiment analyzed the potential for improving 1R by optimizing its selection criterion. The results showed that modifications to 1R's selection criterion would produce at best modest increases in accuracy. To achieve greater increases it is necessary to change the set of rules that 1R "searches" during its "construction" step. For example, 1R's method for partitioning the values of continuous attributes into a set of intervals does not consider all possible partitions. A method of partitioning that considered different partitions might construct 1−rules that are more accurate than any of the 1−rules constructed by the current version of 1R.[9] More fundamental changes might extend 1R's search space to include slightly more complex rules, such as rules that measure 2 attributes or linear trees.

The two methodologies have as their aim the same goal: improving both the accuracy and the simplicity of the rules produced by machine learning systems. But they provide different starting points, emphases, and styles of research towards this goal. The main practical differences in the methodologies are the following.

(1) Systems designed using the "simplicity first" methodology are guaranteed to produce rules that are near-optimal with respect to simplicity. If the accuracy of the rule is unsatisfactory, then there does not exist a satisfactory simple rule, so to improve accuracy one must increase the complexity of the rules being considered. By contrast, systems designed using the traditional methodology may produce rules that are significantly sub-optimal with respect to both simplicity and accuracy. For example, on the VO dataset Buntine & Niblett (1992) report a learning system that produces a decision tree having 12 leaves and an accuracy of 88.2%. This rule is neither accurate nor simple.[10] If this accuracy is unsatisfactory, there may exist a simpler rule that is more accurate. Or there may not. In the traditional methodology one must simply guess where to search for more accurate rules if an unsatisfactory rule is initially produced.

(2) Analysis, such as formal learnability analysis, of simple hypothesis spaces and the associated simple learning algorithms is easier than the corresponding analysis for complex hypothesis spaces. Iba & Langley (1992) give an initial analysis of 1-rule learning behaviour. In this regard, the "simplicity first" methodology for studying and designing learning systems parallels the normal methodology in mathematics of proceeding from simple, easily understood problems through progressively more difficult ones, with the solutions to later problems building upon the results, or using the methods, of the earlier ones. Because the methodologies are parallel, the theory and practice of machine learning may progress together.

---

[9] A recursive partitioning algorithm similar to Catlett's (1991b) creates partitions that are different than 1R's, and no less accurate.

[10] 1R produces a rule having 3 leaves and an accuracy of 95.2%.

(3) Simple hypothesis spaces are so much smaller that algorithms can be used which would be impractical in a larger space. For example, in an acceptably short amount of time PVM (Weiss et al., 1990) can search thoroughly (although not exhaustively) through its relatively small hypothesis space. As a result, PVM is able to find rules of maximal accuracy, at least for their length.[11]

(4) Finally, many of the same issues arise when using a simple hypothesis space as when using a complex one. For example, Weiss et al. (1990) address the issues of accuracy-estimation and partitioning continuous attributes into intervals. Other issues that arise equally with simple and complex hypothesis spaces are: overfitting, tie-breaking (choosing between rules that score equally well on the training data), and the handling of small disjuncts and missing values. Such issues are more easily studied in the smaller, simpler context, and the knowledge derived in this way is, for the most part, transferable to the larger context.

As these differences illustrate, the "simplicity first" methodology is a promising alternative to the existing methodology.

## 8. CONCLUSION

This paper presented the results of an investigation into the classification accuracy of very simple rules ("1–rules", or 1-level decision trees): ones which classify examples on the basis of a single attribute. A program, called 1R, that learns 1–rules from examples was compared to C4 on 16 datasets commonly used in machine learning research.

The main result of comparing 1R and C4 is insight into the tradeoff between simplicity and accuracy. 1R's rules are only a little less accurate (3.1 percentage points) than C4's pruned decision trees on almost all of the datasets. C4's trees are considerably larger in size ("static complexity") than 1–rules, but not much larger in terms of the number of attributes measured to classify the average example ("dynamic complexity").

The fact that, on many datasets, 1–rules are almost as accurate as more complex rules has numerous implications for machine learning research and applications. The first implication is that 1R can be used to predict the accuracy of the rules produced by more sophisticated machine learning systems. In research, this prediction can be used as a benchmark accuracy, giving a reasonable estimate of how one learning system would compare with others. In applications, it can be used to determine if learning is likely to achieve the required accuracy.

A more important implication is that simple-rule learning systems are often a viable alternative to systems that learn more complex rules. If a complex rule is induced, its additional complexity must be justified by its being correspondingly more accurate than a simple rule. In research, this observation leads to an new research methodology that

[11] On two datasets Weiss et al. (1990) searched exhaustively through all rules up to a certain length (2 in one case, 3 in the other). If longer, more accurate rules exist, no one has yet found them.

differs from the traditional methodology in significant ways. In applications, the accuracy and complexity demands of each particular application dictate the choice between the two kinds of system.

The practical significance of this research was assessed by examining whether or not the datasets used in this study are representative of datasets that arise in practice. It was found that most of these datasets are typical of the data available in a commonly occurring class of "real" classification problems. Very simple rules can be expected to perform well on most datasets in this class.

## Acknowledgements

This work was supported in part by an operating grant from the Natural Sciences and Engineering Research Council of Canada. I thank Ross Quinlan for providing the source for C4.5 and David Aha for his work in creating and managing the UCI dataset collection. I am greatly indebted to Doug Fisher, Peter Clark, Bruce Porter, and the "explanation" group at the University of Texas at Austin (Ray Mooney, Rich Mallory, Ken Murray, James Lester, and Liane Acker) for the extensive comments they provided on drafts of this paper. Helpful feedback was also given by Wray Buntine and Larry Watanabe.

## References

Aha, D.W. & Kibler, D. (1989). Noise−Tolerant Instance−based Learning Algorithms. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp.794−799). Morgan Kaufmann.

Bergadano, F., Matwin, S., Michalski, R.S., & Zhang, J. (1992). Learning Two−tiered Descriptions of Flexible Concepts: The Poseidon System. *Machine Learning*, 8, 5−44.

Buntine, W. (1989). Learning Clasification Rules Using Bayes. In A. Segre (Ed.), *Proceedings of the 6th International Workshop on Machine Learning* (pp.94−98). Morgan Kaufmann.

Buntine, W., & Niblett, T. (1992). A Further Comparison of Splitting Rules for Decision−Tree Induction. *Machine Learning*, 8, 75−86.

Catlett, J. (1991a). Megainduction: a Test Flight. In L.A. Birnbaum & G.C. Collins (Eds.), *Proceedings of the Eighth International Conference on Machine Learning* (pp.596−599). Morgan Kaufmann.

Catlett, J. (1991b). On Changing Continuous Attributes into Ordered Discrete Attributes. In Y. Kodratoff (Ed.), *Machine Learning — EWSL−91* (pp.164−178). Springer−Verlag.

Cestnik, B., & Bratko, I. (1991). On Estimating Probabilities in Tree Pruning. In Y. Kodratoff (Ed.), *Machine Learning — EWSL−91* (pp.138−150). Springer−Verlag.

Cestnik, G., Konenenko, I., & Bratko, I. (1987). Assistant−86: A Knowledge−Elicitation Tool for Sophisticated Users. In I. Bratko & N. Lavrac (Eds.), *Progress in Machine Learning* (pp.31−45). Sigma Press.

Clark, P., & Boswell, R. (1991). Rule Induction with CN2: Some Recent Improvements. In Y. Kodratoff (Ed.), *Machine Learning — EWSL−91* (pp.151−163). Springer−Verlag.

Clark, P. & Niblett, T. (1989). The CN2 Induction Algorithm. *Machine Learning,* 3, 261−283.

Clark, P. & Niblett, T. (1987). Induction in Noisy Domains. In I. Bratko & N. Lavrac (Eds.), *Progress in Machine Learning* (pp.11−30). Sigma Press.

de la Maza, M. (1991). A Prototype Based Symbolic Concept Learning System. In L.A. Birnbaum & G.C. Collins (Eds.), *Proceedings of the Eighth International Conference on Machine Learning* (pp.41−45). Morgan Kaufmann.

Diaconis, P. & Efron, B. (1983). Computer−Intensive Methods in Statistics. *Scientific American*, 248.

Fisher, D.H. (1987). Knowledge Acquisition via incremental conceptual clustering. *Machine Learning*, 2, 139−172.

Fisher, D.H., & McKusick, K.B. (1989). An Empirical Comparison of ID3 and Back−Propagation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp.788−793). Morgan Kaufmann.

Fisher, D.H., & Schlimmer, J.C. (1988). Concept Simplification and Prediction Accuracy. In J. Laird (Ed.), *Proceedings of the Fifth International Conference on Machine Learning* (pp.22−28). Morgan Kaufmann.

Hirsh, H. (1990). Learning From Data with Bounded Inconsistency. In B.W. Porter & R.J. Mooney (Eds.), *Proceedings of the Seventh International Conference on Machine Learning* (pp.32−39). Morgan Kaufmann.

Holder Jr., L.B. (1991). Maintaining the Utility of Learned Knowledge Using Model-Based Adaptive Control, Ph.D. thesis, Computer Science Department, University of Illinois at Urbana−Champaign.

Holte, R.C., Acker, L., & Porter, B.W. (1989). Concept Learning and the Problem of Small Disjuncts. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp.813−818). Morgan Kaufmann.

Iba, W.F. & Langley, P. (1992). Induction of One-Level Decision Trees. In D. Sleeman & P. Edwards (Eds.) *Proceedings of the Ninth International Conference on Machine Learning* (pp.233−240). Morgan Kaufmann.

Iba, W., Wogulis, J., & Langley, P. (1988). Trading off Simplicity and Coverage in Incremental Concept Learning. In J. Laird (Ed.), *Proceedings of the Fifth International Conference on Machine Learning* (pp.73−79). Morgan Kaufmann.

Jensen, D. (1992). Induction with Randomization Testing: Decision-oriented Analysis of Large Data Sets. Ph.D. thesis, Washington University, St. Louis, Missouri.

Kibler, D., & Aha, D.W. (1988). Comparing Instance−averaging with Instance−Filtering Learning Algorithms. In D. Sleeman (Ed.), *EWSL88: Proceedings of the 3rd European Working Session on Learning* (pp.63−69). Pitman.

Lopez de Mantaras, R. (1991). A Distance−based Attribute Selection Measure for Decision Tree Induction. *Machine Learning,* 6, 81−92.

McLeish, M. & Cecile, M. (1990). Enhancing Medical Expert Systems With Knowledge Obtained From Statistical Data. *Annals of Mathematics and Artificial Intelligence,* 2, 261−276.

Michalski, R.S. (1990). Learning Flexible Concepts: Fundamental Ideas and a Method Based on Two−Tiered Representation. In Y. Kodratoff & R.S. Michalski (Eds.), *Machine Learning: An Artificial Intelligence Approach (volume 3).* Morgan Kaufmann.

Michalski, R.S., & Chilausky, R.L. (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(2), 125−161.

Michalski, R.S., Mozetic, I., Hong, J., & Lavrac, N. (1986). The Multi−Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp.1041−1045). Morgan Kaufmann.

Mingers, J. (1989). An Empirical Comparison of Pruning Methods for Decision Tree Induction. *Machine Learning*, 4(2), 227−243.

Quinlan, J.R. (1989). Unknown Attribute Values in Induction. In A. Segre (Ed.), *Proceedings of the 6th International Workshop on Machine Learning* (pp.164−168). Morgan Kaufmann.

Quinlan, J.R. (1987). Generating Production Rules from Decision Trees. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp.304−307). Morgan Kaufmann.

Quinlan, J.R. (1986). Induction of Decision Trees. *Machine Learning*, 1, 81−106.

Quinlan, J.R., Compton, P.J., Horn, K.A., & Lazurus, L. (1986). Inductive knowledge acquisition: a case study. *Proceedings of the Second Australian Conference on Applications of Expert Systems.* Sydney, Australia.

Rendell, L. & Seshu, R. (1990). Learning Hard Concepts Through Constructive Induction. *Computational Intelligence*, 6, 247−270.

Salzberg, S. (1991). A Nearest Hyperrectangle Learning Method. *Machine Learning*, 6, 251−276.

Saxena, S. (1989). Evaluating Alternative Instance Representations. In A. Segre (Ed.), *Proceedings of the Sixth International Conference on Machine Learning* (pp.465−468). Morgan Kaufmann.

Schaffer, C. (in press). Overfitting Avoidance as Bias. to appear in *Machine Learning*.

Schaffer, C. (1992). Sparse data and the effect of overfitting avoidance in decision tree induction. *Proceedings of AAAI-92, the Tenth National Conference on Artifical Intelligence*.

Schlimmer, J.S. (1987). Concept Acquisition Through Representational Adjustment (Technical Report 87−19). Ph.D. thesis, Department of Information and Computer Science, University of California, Irvine.

Schoenauer, M. & Sebag, M. (1990). Incremental Learning of Rules and Meta−Rules. in B.W. Porter & R.J. Mooney (Eds.), *Proceedings of the Seventh International Conference on Machine Learning* (pp.49−57). Morgan Kaufmann.

Shapiro, A.D. (1987). *Structured Induction of Expert Systems.* Addison−Wesley.

Shavlik, J., Mooney, R.J., & Towell, G. (1991). Symbolic and Neural Learning Algorithms: An Experimental Comparison. *Machine Learning,* 6, 111−143.

Tan, M., & Eshelman, L. (1988). Using weighted networks to represent classification knowledge in noisy domains. In J. Laird (Ed.), *Proceedings of the Fifth International Conference on Machine Learning* (pp.121−134). Morgan Kaufmann.

Tan, M., & Schlimmer, J. (1990). Two Case Studies in Cost-Sensitive Concept Acquisition. *Proceedings of AAAI-90, the Eighth National Conference on Artifical Intelligence* (pp.854−860). MIT Press.

Utgoff, P.E., & Bradley, C.E. (1990). An Incremental Method for Finding Multivariate Splits for Decision Trees. In B.W. Porter & R.J. Mooney (Eds.), *Proceedings of the Seventh International Conference on Machine Learning* (pp.58−65). Morgan Kaufmann.

Weiss, S.M., Galen, R.S., & Tadepalli, P.V. (1990). Maximizing the Predictive Value of Production Rules. *Artificial Intelligence,* 45, 47−71.

Weiss, S.M., & Kapouleas, I. (1989). An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp.781−787). Morgan Kaufmann.

Wirth, J., & Catlett, J. (1988). Experiments on the Costs and Benefits of Windowing in ID3. In J. Laird (Ed.), *Proceedings of the Fifth International Conference on Machine Learning* (pp.87−99). Morgan Kaufmann.

Yeung, D-Y. (1991). A Neural Network Approach to Constructive Induction. In L.A. Birnbaum & G.C. Collins (Eds.), *Proceedings of the Eighth International Conference on Machine Learning* (pp.228−232). Morgan Kaufmann.

**APPENDIX A. A brief description of the program 1R.**

1R and 1R* are implemented in one program: they are identical except for about 2 lines of code which, if executed, produces 1R*-output in addition to 1R-output (see step 5 below). The user sets a flag to select 1R or 1R*. The user also sets SMALL, the "small disjunct" threshold (Holte et al., 1989).

**Top-level pseudocode.**

1. In the training set count the number of examples in class C having value V for attribute A: store this information in a 3-d array, COUNT[C,V,A].

2. The default class is the one having the most examples in the training set. The accuracy of the default class is the number of training examples in the default class divided by the total number of training examples.

3. FOR EACH NUMERICAL ATTRIBUTE, A, create a nominal version of A by defining a finite number of intervals of values. These intervals become the "values" of the nominal version of A. For example, if A's numerical values are partitioned into three intervals, the nominal version of A will have three values "interval 1", "interval 2", and "interval 3". COUNT[C,V,A] reflects this transformation: COUNT[C,"interval I",A] is the sum of COUNT[C,V,A] for all V in interval I.

   definitions:
   Class C is optimal for attribute A, value V, if it maximizes COUNT[C,V,A].
   Class C is optimal for attribute A, interval I, if it maximizes
   COUNT[C,"interval I",A].

   Values are partitioned into intervals so that every interval satisfies the following constraints:
   (a) there is at least one class that is "optimal" for more than SMALL of the values in the interval. This constraint does not apply to the rightmost interval.
   (b) If V[I] is the smallest value for attribute A in the training set that is larger than the values in interval I then there is no class C that is optimal both for V[I] and for interval I.

4. FOR EACH ATTRIBUTE, A, (use the nominal version of numerical attributes):
   (a) Construct a hypothesis involving attribute A by selecting, for each value V of A (and also for "missing"), an optimal class for V. If several classes are optimal for a value, choose among them randomly.
   (b) add the constructed hypothesis to a set called HYPOTHESES. This set will ultimately contain one hypothesis for each attribute.

5. 1R: choose the rule from the set HYPOTHESES having the highest accuracy on the training set (if there are several "best" rules, choose among them at random).
   1R*: choose all the rules from HYPOTHESES having an accuracy on the training set greater than the accuracy of the default class.

**APPENDIX B. Source of the Datasets Used in this Study.**

All datasets are from the collection distributed by the University of California at Irvine (current contact person: Pat Murphy (pmurphy@ics.uci.edu)). Except as noted below, I used the datasets exactly as they are found in the April 1990 distribution.

Datasets BC and LY were originally collected at the University Medical Center, Institute of Oncology, Ljubljana, Slovenia, by M. Soklic and M. Zwitter, and converted into easy-to-use experimental material by Igor Kononenko, Faculty of Electrical Engineering, Ljubljana University.

BC: breast-cancer/breast-cancer.data
CH: chess-end-games/king-rook-vs-king-pawn/kr-vs-kp.data
GL: glass/glass.data. First attribute deleted. This dataset is sometimes described as having 7 classes, but there are no examples of class 4.
G2: GL with classes 1 and 3 combined and classes 4 through 7 deleted.
HD: heart-disease/cleve.mod. Last attribute deleted to give a 2-class problem.
HE: hepatitis/hepatitis.data
HO: undocumented/taylor/horse-colic.data + horse-colic.test
Attribute V24 is used as the class. Attributes V3, V25, V26, V27, V28 deleted.
HY: thyroid-disease/hypothyroid.data
IR: iris/iris.data
LA: labor-negotiations. The dataset in the April-1990 distribution was in an unusable format. I obtained a usable version — which I believe is now in the UCI collection — from the original source: Stan Matwin, University of Ottawa.
LY: lymphography/lymphography.data
MU: mushroom/agaricus-lepiota.data
SE: thyroid-disease/sick-euthyroid.data
SO: soybean/soybean-small.data
VO: voting-records/house-votes-84.data
V1: VO with the "physician-fee-freeze" attribute deleted.

**APPENDIX C. Survey of results for each dataset.**

The results included in this survey were produced under a very wide variety of experimental conditions and therefore it is impossible to compare them in any detailed manner. Most of the results are averages over a number of runs, where each run involves splitting the dataset into disjoint training and test sets and using the test set to estimate the accuracy of the rule produced given the training set. But the number of runs varies considerably, as does the ratio of the sizes of the training and test set, and different methods of "randomly" splitting have sometimes been used (e.g. cross-validation, stratified sampling, and unstratified sampling). Furthermore, it is virtually certain that some papers reporting results on a dataset have used slightly different versions of the dataset than others, it being common practice to make "small" changes to a dataset for the purposes of a particular experiment.

**Dataset BC**
62.0, "B" (Schoenauer & Sebag, 1990)
62.0, Assistant (no pruning) (Clark & Niblett (1987,1989))
65.0, Bayes (Clark & Niblett (1987,1989))
65.1, CN2 (ordered,laplace) (Clark & Boswell, 1991)
65.3, nearest neighbour (Weiss & Kapouleas, 1989)
65.6, Bayes(second order) (Weiss & Kapouleas, 1989)
65.6, quadratic discriminant (Weiss & Kapouleas, 1989)
66.0, AQTT15 (Michalski, 1990)
66.3, ID unpruned (Peter Clark, personal communication)
66.8, CN2 ordered (Peter Clark, personal communication)
66.8, G-R, Min-err (Mingers, 1989)
66.9, C4 unpruned (Peter Clark, personal communication)
67.0, Assistant (no pruning) (Michalski, 1990)
67.4, Prob, Min-err (Mingers, 1989)
68.0, AQ11/15 (Tan & Eshelman, 1988)
68.0, AQ15 (Salzberg, 1991)
68.0, AQTT15 (biggest disjuncts) (Michalski, 1990)
68.0, AQTT15 (unique>1) (Michalski, 1990)
68.0, Assistant (pruning) (Clark & Niblett (1987,1989))
68.3, G-stat, Min-err (Mingers, 1989)
68.7, ========================================================= 1R ====
68.7, Marsh, Min-err (Mingers, 1989)
69.0, CN2 (ordered,entropy) (Clark & Boswell, 1991)
69.2, Gain Ratio (Lopez de Mantaras, 1991)
69.3, G-R, Critical (Mingers, 1989)
69.3, ID3 (Tan & Eshelman, 1988)
69.6, G-stat, Critical (Mingers, 1989)
69.7, G-R, Err-comp (Mingers, 1989)
70.0, Assistant (no pruning) (Cestnik et al., 1987))
70.3, ============================== BASELINE ACCURACY ================
70.4, random (Buntine & Niblett, 1992)
70.6, Distance (Lopez de Mantaras, 1991)
(continued next page)

**Dataset BC** (continued)
70.8, G-R, reduce (Mingers, 1989)
71.0, CN2(99) (Clark & Niblett (1987,1989))
71.0, Prob, Critical (Mingers, 1989)
71.5, C4 pruned (Peter Clark, personal communication)
71.5, EACH without feature adjustment (Salzberg, 1991)
71.5, Info Gain (Buntine & Niblett, 1992)
71.5, Prob, Err-comp (Mingers, 1989)
71.5, neural net (Weiss & Kapouleas, 1989)
71.6, G-R, Pessim (Mingers, 1989)
71.6, linear discriminant (Weiss & Kapouleas, 1989)
71.8, Bayes (Weiss & Kapouleas, 1989)
71.9, Marsh, Pessim (Mingers, 1989)
71.9, Prob, Pessim (Mingers, 1989)
72.0, AQ15 (Michalski et al., 1986)
72.0, AQR (Clark & Niblett (1987,1989))
72.0, Assistant (pruning) (Michalski, 1990)
72.0, C4 (pruned) (this paper)
72.0, G-stat, Err-comp (Mingers, 1989)
72.1, C4 (Clark & Boswell, 1991)
72.3, GINI (Buntine & Niblett, 1992)
72.3, Marsh, Critical (Mingers, 1989)
72.3, Marsh, Err-comp (Mingers, 1989)
72.5, G-stat, Pessim (Mingers, 1989)
72.7, ============================================================ 1Rw ===
72.8, ============================================================ 1R* ===
72.8, Prob, Reduce (Mingers, 1989)
72.9, G-stat, Reduce (Mingers, 1989)
72.9, Marsh (Buntine & Niblett, 1992)
73.0, CN2 (unordered,laplace) (Clark & Boswell, 1991)
73.1, Marsh, Reduce (Mingers, 1989)
73.4, IWN(add-or) (Tan & Eshelman, 1988)
73.5, IWN(max-or) (Tan & Eshelman, 1988)
74.3, ID3 (pruned) (Buntine, 1989)
75.0, "C2" (Schoenauer & Sebag, 1990)
75.6, Bayes/N (Buntine, 1989)
76.1, Bayes (Buntine, 1989)
76.2, ID3 (averaged) (Buntine, 1989)
77.0, Assistant (pre-pruning) (Cestnik et al., 1987))
77.1, CART (Weiss & Kapouleas, 1989)
77.1, PVM (Weiss & Kapouleas, 1989)
77.6, EACH with feature adjustment (Salzberg, 1991)
78.0, "D3" (Schoenauer & Sebag, 1990)
78.0, Assistant (post-pruning) (Cestnik et al., 1987))
78.0, Bayes (Cestnik et al., 1987))

**Dataset CH**
67.6, ========================================================================== 1R ====
68.3, ========================================================================== 1Rw ===
69.2, ========================================================================== 1R* ===
85.4, Bayes (Buntine, 1989)
91.0, CN2 (Holte et al., 1989)
93.9, perceptron (Shavlik et al., 1991)
96.3, back propagation (Shavlik et al., 1991)
96.4, ID3 (pruned) (Buntine, 1989)
96.9, ID3 (unpruned) (Buntine, 1989)
97.0, ID3 (Shavlik et al., 1991)
99.2, C4 (pruned) (this paper)

**Dataset GL**
45.5, NTgrowth (de la Maza, 1991)
46.8, random (Buntine & Niblett, 1992)
48.0, Proto-TO (de la Maza, 1991)
49.4, Info Gain (Buntine & Niblett, 1992)
53.8, ========================================================================== 1R ====
56.3, ========================================================================== 1R* ===
59.5, Marsh (Buntine & Niblett, 1992)
60.0, GINI (Buntine & Niblett, 1992)
62.2, ========================================================================== 1Rw ===
63.2, C4 (pruned) (this paper)
65.5, C4 (de la Maza, 1991)

**Dataset HD**
60.5, perceptron (Shavlik et al., 1991)
70.5, growth (Aha & Kibler, 1989)
71.1, K-nearest neighbour growth (K=3) (Aha & Kibler, 1989)
71.2, ID3 (Shavlik et al., 1991)
71.3, disjunctive spanning (Aha & Kibler, 1989)
71.4, growth (Kibler & Aha, 1988)
73.4, =========================================================================== 1R ===
73.6, C4 (pruned) (this paper)
74.8, C4 (Kibler & Aha, 1988)
75.4, C4 (Aha & Kibler, 1989)
76.2, proximity (Aha & Kibler, 1989)
76.4, IRT (Jensen, 1992)
76.6, ========================================================================== 1Rw ===
77.0, NTgrowth (Kibler & Aha, 1988)
77.9, NTgrowth (Aha & Kibler, 1989)
78.0, ========================================================================== 1R* ===
78.7, NT disjunctive spanning (Aha & Kibler, 1989)
79.2, K-nearest neighbour (K=3) (Aha & Kibler, 1989)
79.4, NT K-nearest neighbour growth (K=3) (Aha & Kibler, 1989)
80.6, back propagation (Shavlik et al., 1991)

**Dataset HE**

38.7, NTgrowth (de la Maza, 1991)
71.3, CN2 (ordered,entropy) (Clark & Boswell, 1991)
76.3, ============================================================ 1R ====
77.6, CN2 (ordered,laplace) (Clark & Boswell, 1991)
77.8, ID unpruned (Peter Clark, personal communication)
78.6, Gain Ratio (Lopez de Mantaras, 1991)
79.3, C4 (Clark & Boswell, 1991)
79.3, Distance (Lopez de Mantaras, 1991)
79.4, ================================ BASELINE ACCURACY ================
79.8, C4 (de la Maza, 1991)
79.8, m=0.0 (Cestnik & Bratko, 1991)
79.8, m=0.01 (Cestnik & Bratko, 1991)
79.9, Proto-TO (de la Maza, 1991)
80.0, (cited in the UCI files) (Diaconis & Efron, 1983)
80.0, Assistant (no pruning) (Cestnik et al., 1987))
80.1, CN2 (unordered,laplace) (Clark & Boswell, 1991)
81.1, m=1 (Cestnik & Bratko, 1991)
81.2, C4 (pruned) (this paper)
81.5, m=0.5 (Cestnik & Bratko, 1991)
82.0, Assistant (post-pruning) (Cestnik et al., 1987))
82.1, laplace (Cestnik & Bratko, 1991)
82.1, m=2 (Cestnik & Bratko, 1991)
83.0, Assistant (pre-pruning) (Cestnik et al., 1987))
83.6, m=3 (Cestnik & Bratko, 1991)
83.8, m=128 (Cestnik & Bratko, 1991)
83.8, m=32 (Cestnik & Bratko, 1991)
83.8, m=64 (Cestnik & Bratko, 1991)
83.8, m=999 (Cestnik & Bratko, 1991)
84.0, Bayes (Cestnik et al., 1987))
84.0, m=8 (Cestnik & Bratko, 1991)
84.5, ============================================================ 1Rw ===
84.5, m=4 (Cestnik & Bratko, 1991)
85.5, m=12 (Cestnik & Bratko, 1991)
85.5, m=16 (Cestnik & Bratko, 1991)
85.8, ============================================================ 1R* ===

**Dataset HY**
88.4, quadratic discriminant (Weiss & Kapouleas, 1989)
92.4, Bayes(second order) (Weiss & Kapouleas, 1989)
92.6, random (Buntine & Niblett, 1992)
93.9, linear discriminant (Weiss & Kapouleas, 1989)
95.3, nearest neighbour (Weiss & Kapouleas, 1989)
96.1, Bayes (Weiss & Kapouleas, 1989)
97.1, growth (Kibler & Aha, 1988)
97.2, ============================================================ 1R ====
97.4, ============================================================ 1R* ===
97.7, NTgrowth (Kibler & Aha, 1988)
98.0, ============================================================ 1Rw ===
98.2, C4 (Kibler & Aha, 1988)
98.5, neural net (Weiss & Kapouleas, 1989)
98.7, Marsh (Buntine & Niblett, 1992)
99.0, GINI (Buntine & Niblett, 1992)
99.1, C4 (pruned) (this paper)
99.1, Info Gain (Buntine & Niblett, 1992)
99.1, PT2 (Utgoff & Bradley, 1990)
99.3, C4-rules (Quinlan, 1987)
99.3, PVM (Weiss & Kapouleas, 1989)
99.4, C4 (Quinlan, 1987)
99.4, CART (Weiss & Kapouleas, 1989)
99.7, C4 (Quinlan et al., 1986)

**Dataset IR**
84.0, Bayes(second order), (cross-validation) (Weiss & Kapouleas, 1989)
85.8, random (Buntine & Niblett, 1992)
89.3, Prob, Reduce (Mingers, 1989)
90.5, Prob, Err-comp (Mingers, 1989)
91.1, Marsh, Critical (Mingers, 1989)
91.2, Marsh, Pessim (Mingers, 1989)
91.3, Prob, Critical (Mingers, 1989)
92.2, Marsh, Min-err (Mingers, 1989)
92.3, NTgrowth (de la Maza, 1991)
92.4, Marsh, Err-comp (Mingers, 1989)
92.4, Marsh, Reduce (Mingers, 1989)
92.4, Prob, Pessim (Mingers, 1989)
92.4, growth (Kibler & Aha, 1988)
92.5, G-R, Critical (Mingers, 1989)
92.5, G-R, Err-comp (Mingers, 1989)
92.5, G-R, Pessim (Mingers, 1989)
92.5, G-R, reduce (Mingers, 1989)
92.6, EACH without feature adjustment (Salzberg, 1991)
92.8, G-stat, Critical (Mingers, 1989)
92.8, G-stat, Err-comp (Mingers, 1989)
92.8, G-stat, Min-err (Mingers, 1989)
(continued next page)

**Dataset IR** (continued)
92.8, G-stat, Pessim (Mingers, 1989)
92.8, G-stat, Reduce (Mingers, 1989)
93.0, CART (Salzberg, 1991)
93.2, G-R, Min-err (Mingers, 1989)
93.3, Bayes, (cross-validation) (Weiss & Kapouleas, 1989)
93.3, Prob, Min-err (Mingers, 1989)
93.5, ============================================================ 1R ====
93.8, C4 (pruned) (this paper)
94.0, ID3 (pruned) (Buntine, 1989)
94.2, C4 (de la Maza, 1991)
94.2, ID3 (Catlett, 1991a)
94.2, ID3 (new version) (Catlett, 1991a)
94.4, C4 (Kibler & Aha, 1988)
94.4, ID3 (averaged) (Buntine, 1989)
94.5, Marsh (Buntine & Niblett, 1992)
94.7, Dasarathy (Hirsh, 1990)
95.0, GINI (Buntine & Niblett, 1992)
95.1, Info Gain (Buntine & Niblett, 1992)
95.3, CART, (cross-validation) (Weiss & Kapouleas, 1989)
95.3, EACH with feature adjustment (Salzberg, 1991)
95.4, NTgrowth (Kibler & Aha, 1988)
95.5, Bayes (Buntine, 1989)
95.5, Bayes/N (Buntine, 1989)
95.9, ============================================================ 1R* ===
96.0, ============================================================ 1Rw ===
96.0, PVM, (cross-validation) (Weiss & Kapouleas, 1989)
96.0, Proto-TO (de la Maza, 1991)
96.0, nearest neighbour, (cross-validation) (Weiss & Kapouleas, 1989)
96.7, IVSM, (Hirsh, 1990)
96.7, neural net, (cross-validation) (Weiss & Kapouleas, 1989)
97.3, quadratic discriminant, (cross-validation) (Weiss & Kapouleas, 1989)
98.0, linear discriminant, (cross-validation) (Weiss & Kapouleas, 1989)

**Dataset LA**
71.5, ============================================================ 1R ====
77.0, 1-nearest neighbour (Bergadano et al., 1992)
77.3, C4 (pruned) (this paper)
80.0, 5-nearest neighbour (Bergadano et al., 1992)
80.0, AQ15 (strict or flexible matching) (Bergadano et al., 1992)
83.0, 3-nearest neighbour (Bergadano et al., 1992)
83.0, AQ15 ("top rule" truncation) (Bergadano et al., 1992)
83.0, AQ15 (TRUNC-SG,flexible matching) (Bergadano et al., 1992)
84.2, ============================================================ 1Rw ===
86.0, Assistant (with pruning) (Bergadano et al., 1992)
87.4, ============================================================ 1R* ===
90.0, AQ15 (TRUNC-SG,deductive matching) (Bergadano et al., 1992)

**Dataset LY**
56.1, m=999 (Cestnik & Bratko, 1991)
67.7, random (Buntine & Niblett, 1992)
69.1, m=128 (Cestnik & Bratko, 1991)
70.7, ============================================================ 1R ====
71.5, CN2 (ordered,entropy) (Clark & Boswell, 1991)
74.8, m=64 (Cestnik & Bratko, 1991)
75.0, m=16 (Cestnik & Bratko, 1991)
75.6, GINI (Buntine & Niblett, 1992)
75.7, ============================================================ 1Rw ===
75.7, Marsh (Buntine & Niblett, 1992)
75.9, m=12 (Cestnik & Bratko, 1991)
75.9, m=32 (Cestnik & Bratko, 1991)
76.0, AQR (Clark & Niblett (1987,1989))
76.0, Assistant (no pruning) (Cestnik et al., 1987))
76.0, Assistant (no pruning) (Michalski, 1990)
76.0, Assistant (post-pruning) (Cestnik et al., 1987))
76.0, Assistant (pre-pruning) (Cestnik et al., 1987))
76.0, Info Gain (Buntine & Niblett, 1992)
76.4, C4 (Clark & Boswell, 1991)
76.7, ID3 (pruned) (Buntine, 1989)
76.8, m=8 (Cestnik & Bratko, 1991)
77.0, Assistant (pruning) (Michalski, 1990)
77.1, laplace (Cestnik & Bratko, 1991)
77.1, m=4 (Cestnik & Bratko, 1991)
77.3, ============================================================ 1R* ===
77.3, m=0.01 (Cestnik & Bratko, 1991)
77.3, m=0.5 (Cestnik & Bratko, 1991)
77.3, m=1 (Cestnik & Bratko, 1991)
77.3, m=2 (Cestnik & Bratko, 1991)
77.3, m=3 (Cestnik & Bratko, 1991)
77.5, C4 (pruned) (this paper)
77.5, m=0.0 (Cestnik & Bratko, 1991)
78.0, Assistant (pruning) (Clark & Niblett (1987,1989))
78.4, ID3 (averaged) (Buntine, 1989)
79.0, Assistant (no pruning) (Clark & Niblett (1987,1989))
79.0, Bayes (Cestnik et al., 1987))
79.6, CN2 (ordered,laplace) (Clark & Boswell, 1991)
80.0, AQTT15 (unique>1) (Michalski, 1990)
81.0, AQTT15 (Michalski, 1990)
81.7, CN2 (unordered,laplace) (Clark & Boswell, 1991)
82.0, AQ15 (Michalski et al., 1986)
82.0, AQTT15 (biggest disjuncts) (Michalski, 1990)
82.0, Bayes/N (Buntine, 1989)
82.0, CN2(99) (Clark & Niblett (1987,1989))
83.0, Bayes (Clark & Niblett (1987,1989))
85.1, Bayes (Buntine, 1989)

**Dataset MU**
91.2, random (Buntine & Niblett, 1992)
92.7, Marsh (Buntine & Niblett, 1992)
95.0, HILLARY (Iba et al., 1988)
95.0, STAGGER (Schlimmer, 1987)
98.4, =========================================================== 1R ====
98.4, =========================================================== 1R* ===
98.5, =========================================================== 1Rw ===
98.6, GINI (Buntine & Niblett, 1992)
98.6, Info Gain (Buntine & Niblett, 1992)
99.1, neural net (Yeung, 1991)
99.9, ID3 (Wirth & Catlett, 1988)
100.0, C4 (pruned) (this paper)

**Dataset SE**
91.8, growth (Kibler & Aha, 1988)
95.0, =========================================================== 1R ====
95.0, =========================================================== 1R* ===
95.0, =========================================================== 1Rw ===
95.0, RAF (Quinlan, 1989)
95.2, RUU (Quinlan, 1989)
95.4, RSS (Quinlan, 1989)
95.9, NTgrowth (Kibler & Aha, 1988)
96.1, RPF (Quinlan, 1989)
96.2, RFF (Quinlan, 1989)
96.3, RIF (Quinlan, 1989)
96.8, RCF (Quinlan, 1989)
97.3, C4 (Kibler & Aha, 1988)
97.7, C4 (pruned) (this paper)
99.2, C4 (Quinlan et al., 1986)

**Dataset VO**
84.0, 3-nearest neighbour (Bergadano et al., 1992)
84.0, 5-nearest neighbour (Bergadano et al., 1992)
84.6, random (Buntine & Niblett, 1992)
85.0, AQ15 ("top rule" truncation) (Bergadano et al., 1992)
85.2, NT K-nearest neighbour growth (K=3) (Aha & Kibler, 1989)
86.0, 1-nearest neighbour (Bergadano et al., 1992)
86.0, AQ15 (strict or flexible matching) (Bergadano et al., 1992)
86.0, Assistant (with pruning) (Bergadano et al., 1992)
86.2, K-nearest neighbour (K=3) (Aha & Kibler, 1989)
86.4, K-nearest neighbour growth (K=3) (Aha & Kibler, 1989)
88.2, Marsh (Buntine & Niblett, 1992)
90.4, Proto-TO (de la Maza, 1991)
90.6, NTgrowth (de la Maza, 1991)
90.7, growth (Aha & Kibler, 1989)
90.8, IWN(add-or) (Tan & Eshelman, 1988)
(continued next page)

**Dataset VO** (continued)
91.7, proximity. (Aha & Kibler, 1989)
91.9, NTgrowth (Aha & Kibler, 1989)
92.0, AQ15 (TRUNC-SG,deductive matching) (Bergadano et al., 1992)
92.0, AQ15 (TRUNC-SG,flexible matching) (Bergadano et al., 1992)
92.4, disjunctive spanning (Aha & Kibler, 1989)
92.9, NT disjunctive spanning (Aha & Kibler, 1989)
93.6, CN2 (ordered,entropy) (Clark & Boswell, 1991)
93.9, IWN(max-or) (Tan & Eshelman, 1988)
94.0, ID3 (Fisher & McKusick, 1989)
94.3, IWN(add-or) (Tan & Eshelman, 1988)
94.5, C4 (Aha & Kibler, 1989)
94.8, CN2 (ordered,laplace) (Clark & Boswell, 1991)
94.8, CN2 (unordered,laplace) (Clark & Boswell, 1991)
95.0, IRT (Jensen, 1992)
95.0, STAGGER (Schlimmer, 1987)
95.2, ============================================================ 1R ====
95.2, ============================================================ 1R* ===
95.3, C4 (de la Maza, 1991)
95.3, neural net (Yeung, 1991)
95.4, Info Gain (Buntine & Niblett, 1992)
95.5, GINI (Buntine & Niblett, 1992)
95.6, ============================================================ 1Rw ===
95.6, C4 (Clark & Boswell, 1991)
95.6, C4 (pruned) (this paper)

**Dataset V1**
84.4, random (Buntine & Niblett, 1992)
84.9, Marsh (Buntine & Niblett, 1992)
86.8, ============================================================ 1R ====
87.0, Info Gain (Buntine & Niblett, 1992)
87.2, GINI (Buntine & Niblett, 1992)
87.4, ============================================================ 1Rw ===
87.9, ============================================================ 1R* ===
89.4, C4 (pruned) (this paper)

## APPENDIX D. Data from the 25 runs on each dataset.

| Dataset | 1R* mean | 1R* std.dev. | C4 mean | C4 std.dev. | 1Rw | t-values C4-1R* | t-values 1R*-1Rw | t-values C4-1Rw |
|---|---|---|---|---|---|---|---|---|
| BC | 72.46 | 4.23 | 71.96 | 4.36 | 72.7 | -0.82 | -0.27 | -0.83 |
| CH | 69.24 | 0.95 | 99.19 | 0.27 | 68.3 | 134.7 | 4.85 | 571 |
| GL | 56.44 | 5.06 | 63.16 | 5.71 | 62.2 | 4.98 | -5.53 | 0.86 |
| G2 | 77.02 | 3.88 | 74.26 | 6.61 | 78.5 | -1.93 | -1.90 | -3.17 |
| HD | 78.00 | 2.68 | 73.62 | 4.44 | 76.57 | -5.18 | 2.62 | -3.26 |
| HE | 85.14 | 6.23 | 81.23 | 5.12 | 84.5 | -2.69 | 0.51 | -3.13 |
| HO | 81.18 | 1.95 | 83.61 | 3.41 | 81.5 | 4.05 | -0.86 | 3.01 |
| HY | 97.20 | 0.67 | 99.13 | 0.27 | 98.0 | 13.44 | -5.88 | 20.59 |
| IR | 95.92 | 1.55 | 93.76 | 2.96 | 96.0 | -3.61 | -0.25 | -3.71 |
| LA | 87.37 | 4.48 | 77.25 | 5.89 | 84.2 | -7.11 | 3.47 | -5.78 |
| LY | 77.28 | 3.75 | 77.52 | 4.46 | 75.7 | 0.21 | 2.07 | 2.00 |
| MU | 98.44 | 0.20 | 100.0 | 0.0 | 98.5 | 37.47 | -1.44 | ∞ |
| SE | 95.00 | 0.54 | 97.69 | 0.40 | 95.0 | 30.3 | 0.04 | 32.8 |
| SO | 87.00 | 6.62 | 97.51 | 3.94 | 87.2 | 6.55 | -0.15 | 12.81 |
| VO | 95.18 | 1.52 | 95.57 | 1.31 | 95.6 | 1.59 | -1.34 | -0.12 |
| V1 | 87.93 | 2.22 | 89.36 | 2.45 | 87.4 | 3.93 | 1.17 | 3.92 |