Working Paper Series
ISSN 1170-487X

Selecting Multiway Splits in Decision Trees

by: Eibe Frank, Ian H. Witten

Working Paper 96/31
December 1996

© 1996 Eibe Frank, Ian H. Witten
Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Selecting Multiway Splits in Decision Trees

Eibe Frank **Ian H. Witten**

Department of Computer Science
University of Waikato
Hamilton, New Zealand

Abstract: Decision trees in which numeric attributes are split several ways are more comprehensible than the usual binary trees because attributes rarely appear more than once in any path from root to leaf. There are efficient algorithms for finding the optimal multiway split for a numeric attribute, given the number of intervals in which it is to be divided. The problem we tackle is how to choose this number in order to obtain small, accurate trees.

We view each multiway decision as a model and a decision tree as a recursive structure of such models. Standard methods of choosing between competing models include resampling techniques (such as cross-validation, holdout, or bootstrap) for estimating the classification error; and minimum description length techniques. However, the recursive situation differs from the usual one, and may call for new model selection methods.

This paper introduces a new criterion for model selection: a resampling estimate of the information gain. Empirical results are presented for building multiway decision trees using this new criterion, and compared with criteria adopted by previous authors. The new method generates multiway trees that are both smaller and more accurate than those produced previously, and their performance is comparable with standard binary decision trees.

Keywords: Inductive learning, classification, decision-tree learning, recursive model selection, cross-validation.

Email: {eibe, ihw}@cs.waikato.ac.nz

Phone: 0064 7 856 2889 6026

Fax: 0064 7 838 4155

Selecting Multiway Splits in Decision Trees

Abstract: Decision trees in which numeric attributes are split several ways are more comprehensible than the usual binary trees because attributes rarely appear more than once in any path from root to leaf. There are efficient algorithms for finding the optimal multiway split for a numeric attribute, given the number of intervals in which it is to be divided. The problem we tackle is how to choose this number in order to obtain small, accurate trees.

We view each multiway decision as a model and a decision tree as a recursive structure of such models. Standard methods of choosing between competing models include resampling techniques (such as cross-validation, holdout, or bootstrap) for estimating the classification error; and minimum description length techniques. However, the recursive situation differs from the usual one, and may call for new model selection methods.

This paper introduces a new criterion for model selection: a resampling estimate of the information gain. Empirical results are presented for building multiway decision trees using this new criterion, and compared with criteria adopted by previous authors. The new method generates multiway trees that are both smaller and more accurate than those produced previously, and their performance is comparable with standard binary decision trees.

1 Introduction

This paper studies methods for generating concise decision trees with multiway splits for numeric attributes—or, in general, any attribute whose values form a totally ordered set (we also accommodate nominal attributes in the normal manner). Multiway trees offer the advantage over binary trees that an attribute rarely appears more than once in any path from root to leaf. This makes them easier to comprehend. Moreover, in our experience users of machine learning are often interested in ranking the attributes according to how much they contribute to the classification of particular instances. Such a ranking can be read off from a multiway tree simply by tracking the classification of an instance, starting at the root, and collecting the attributes being tested.

Several algorithms for finding the optimal multiway split on a numeric attribute with respect to a given splitting criterion have appeared recently (Maass, 1994, Fulton *et al.*, 1995, Elomaa & Rousu, 1996). However, how to determine the

number of split points at each node in order to get a small and accurate tree is still an open question.

Two previous papers introduce methods of creating multiway trees. Fulton *et al.* (1995) devise a dynamic programming algorithm to generate a split into a given number of intervals which is optimal with respect to an additive impurity measure.[1] They proceed to investigate two ways of using this algorithm for tree-building. First, with the classification error as the impurity measure, they use Quinlan's (1993) information gain ratio, which penalizes larger numbers of splits, to select the actual number of intervals. The gain ratio cannot be used directly as the impurity measure because it is not additive. However, in their second method, Fulton *et al.* devise an *ad hoc* modification of the information gain that does preserve additivity (although it is not fully described in the paper). Neither method is particularly satisfactory, and they conclude that the problem remains unsolved, proposing in the future to take a different tack by investigating an extension of Breiman's (1984) "twoing" rule for multiway splits.

Fayyad and Irani (1993) create multiway trees by devising a way of generating a multiway split on a numeric attribute that incorporates the decision of how many intervals to create. It operates by building a binary decision tree for the class based on this attribute alone, using the information gain as splitting criterion and a minimum description length heuristic to decide whether or not to expand a node. The multiway split consists of the intervals corresponding to this tree's leaves. Then, this procedure is used recursively to build a decision tree for the original problem, the appropriate attribute at each node being determined by comparing the information gain of the multiway splits for each attribute. Note that although the multiway split procedure determines the number of intervals for the split, it does not necessarily find the optimal multiway split for that number of intervals.

Both these procedures for generating multiway trees create full trees, and although the resulting trees could no doubt be pruned, the pruning problem is not addressed. In contrast, we take a more general view. The task of choosing a multiway split is a model selection problem—each split is a model that divides the domain into disjoint subsets. One possibility is the null model, which does not divide the domain at all. This model selection approach automatically incorporates pre-pruning of the tree.

---

[1] To find the best n-way split for the first i sorted examples, dynamic programming takes the best $(n-1)$-way split for the first j examples and supposes the remaining $j-i$ examples to be covered by an nth interval. If the impurity measure is additive, the impurity of the resulting split is just the impurity of the original split plus the impurity of the nth interval, which is a key property for the application of the dynamic programming method.

In machine learning, several theoretically well-founded methods have been developed for selecting between competing classification models. These divide broadly into resampling procedures for estimating classification error, such as cross-validation, holdout, and bootstrap methods, and estimates based on the training data, such as minimum description length methods and *ad hoc* approximations like Quinlan's gain ratio. Kearns *et al.* (1995) show that resampling procedures are to be preferred if nothing is known *a priori* about the learning problem at hand. Model selection in our context, however, differs in that the models are selected recursively for increasingly smaller subsets of the domain—except for those directly above the leaves, the internal nodes are not classifiers in the ordinary sense. Consequently, instead of using a resampling estimate of the classification error, we introduce a resampling estimate of the information gain as the criterion for recursive model selection.

It is well-known that information gain is more appropriate than classification error as a criterion for tree-building, and the main claim of this paper is that if it is estimated correctly—that is, using resampling techniques—it can simultaneously solve the problems of determining the number of intervals for multiway splits, and of terminating growth by pre-pruning. To support this claim, we present empirical results for building decision trees by selecting multiway splits using this new criterion, its error-based counterpart, the minimum description length principle, and the gain ratio criterion.

The paper is organized as follows. Section 2 identifies differences between global model selection, as employed when choosing between different classifiers, and recursive model selection. Section 3 explains the criteria we investigated in our experiments for selecting multiway splits. Section 4 presents empirical results obtained by testing these criteria on fifteen UCI data sets. In the last section we summarize our conclusions and present ideas for future work.

2 Model Selection and Recursive Model Selection

The aim of model selection is to estimate the true performance of a model in order that models can be compared according to a given performance criterion. When models are used for classification, performance is generally measured by the classification error. It is well known that for practical learning problems the classification error on the training examples is an unreliable indicator of performance on new examples, because it is biased towards unnecessarily complex models. For this reason, classifiers produced by different learning algorithms are generally compared using estimates like cross-validation that measure performance on data different from that used for training. These methods are called "resampling"

methods. An alternative to resampling is to estimate the performance of a model by its classification error on the training data, but to incorporate a correction that penalizes complex models. The minimum description length principle, which can be justified using Bayes' theorem, provides one such method. Another correction is Quinlan's (1993) well-known gain ratio criterion.

In this paper, we extend the idea of model selection to apply recursively at each node during construction of a decision tree, rather than applying it to the finished model. When building a conventional decision tree, the model selection problem arises whenever attributes with different numbers of branches are compared. When constructing a multiway tree for numeric attributes, the problem is more acute because as well as comparing attributes with each other, one must decide for each individual attribute how many ways to branch.

Most conventional decision tree methods use some kind of *ad hoc* correction to rectify the tendency to select attributes that branch many ways. The problem has been studied in a general setting—though in the context of nominal attributes only—by Kononenko (1996), who compared several criteria for choosing between attributes in a decision tree. He concludes that a minimum description length criterion is the one with the most appropriate bias.

Brodley (1995a) introduced the idea of using a resampling estimate, namely cross-validation, for determining the best model to use at each node of the tree. She was concerned with selecting the best of three model types—decision tree, linear classifier, or k-nearest neighbor—for each node, and was the first to view the construction of a decision tree as a recursive model selection problem, a fruitful general perspective which our work capitalizes on. In her experiments cross-validation led to less accurate decision trees than a set of hand-crafted rules, which were specifically designed to select between the above three model types. However, in the cross-validation experiments she chose to estimate the classification error as the basis for model selection, which in our view is a mistake. A better choice, and one that we investigate in this paper, is the information gain.

It is well known that building decision trees using information gain as the splitting criterion results in more accurate trees than trying to minimize the classification error at each node directly (Pazzani *et al.*, 1994). This is consistent with results for entropy-based and error-based discretization of numeric attributes: Kohavi and Sahami (1996) found that discretization using an entropy-based scheme leads to better results than discretization using an error-based scheme.

Kohavi and Sahami (1996) give a compelling intuitive explanation of their findings in the context of discretization, which it is worth reviewing because the use of an entropy-based rather than an error-based measure is central to this paper. Consider

the situation depicted in Figure 1. Splitting at point **a** will ultimately produce a better decision tree because a secondary split will be made on the other attribute. An error-based method for selecting the split-point will find point **b**—it will not choose **a** since that is not necessary to minimize the classification error. On the other hand, an entropy-based method, being sensitive to changes in the class distribution, would detect both **a** and **b** as potential split-points. In general, an error-based criterion may not find any clearly defined split-point and base its decision on minor random fluctuations, leading it to choose an unimportant attribute on which to divide the training set and thereby reducing the likelihood of identifying important attribute interactions further down the tree because of the limited amount of data. In contrast, entropy-based methods have a better chance of discovering important attribute interactions since these induce major changes in the class distribution.

In summary, the problem of recursive model selection has been considered by

- Brodley (1995), who used cross-validation based on the classification error and concluded that it gave worse results than a set of hand-crafted rules;
- Kononenko (1996), who compared several criteria and concluded that a minimum description length criterion had the most appropriate bias;
- Fulton *et al.* (1995), who used the gain ratio criterion as well as a heuristic of their own devising and concluded that other methods should be pursued;
- Fayyad & Irani (1993), who used a form of the MDL principle.

Of these, Brodley alone specifically identified the problem as a general model selection one; she used a variety of different model types. Kononenko considered decision trees with nominal attributes only. The last two papers examined the multiway split problem for numeric attributes. Although Fayyad and Irani used the minimum description length principle, they used it to terminate the expansion of a binary tree—which is not equivalent to a minimum description length evaluation of the overall multiway split. None of the four considered the use of a resampling estimate of the information gain, which we have argued is the most promising choice based on *a priori* arguments and evidence from similar situations.

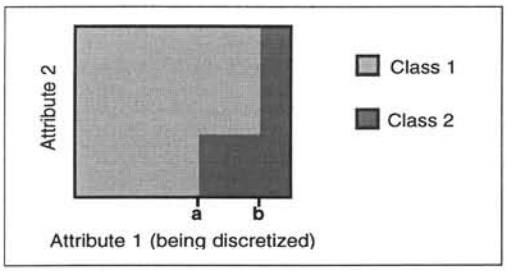


Figure 1: Example for the failure of error-based discretization

3 Criteria for Selecting Multiway Splits

Four possible alternative criteria for generating and selecting multiway splits arise out of the above discussion:

- generate splits using the classification error and evaluate them with a resampling technique (cross-validation, holdout or bootstrap) to estimate the true classification error;
- generate splits using the information gain and evaluate them with a resampling technique to estimate the true information gain;
- generate splits using the information gain and evaluate them using minimum description length;
- generate splits using the information gain and evaluate them with the gain ratio criterion.

We examine each in turn: an experimental comparison appears in the next section.

Resampling using classification error

A general way of combating the problem of overfitting is to use a resampling estimate of the true performance of a model, measured in terms of the classification error. Kohavi (1995) recommends ten-fold cross-validation for model selection, and this is what we employed (we also used the bootstrap method, and it gave results so similar that they are not reported here). Suppose we are measuring the effect of splitting on a numeric attribute for a given data set. For each value of k, from 1 up to the number of potential splitpoints $k_{\max}$ for that attribute, we consider a k-way split and calculate an estimate of the resulting classification error using ten-fold cross-validation. As discussed below, this can be done in time linear in $k_{\max}$. This estimate is employed to select the optimum value of k, and also to compare the effect of splitting on this attribute against other attributes. The attribute and k-value which score best are chosen for this node of the decision tree. (We follow the same procedure for nominal attributes, for which, of course, k is predetermined.)

The classification error for a k-way split of a numeric attribute is calculated as the average over ten folds of $L_{ErrorRate}(S',S)$, where S' and S are the training and test sets for one fold of the cross-validation, and the error for this fold is

$$L_{ErrorRate}(S',S) = \frac{1}{|S|}\sum_{i=1}^{k}\left[|S_i| - \left|\{s \in S_i \mid Class(S_i) = majorityClass(S_i')\}\right|\right] \qquad (1)$$

Here, $|S_i|$ is the number of members of the test set S that fall into the ith subinterval of the discretization, from which we subtract the number of these that belong to the majority class dictated by the training set on that subinterval, S'_i.

Resampling using information gain

Essentially the same procedure is used to estimate the information gain of a proposed split. However, a problem arises because when a multiway split is generated on a training set, it is quite common for an interval to occur that contains no examples of a particular class, giving it zero frequency in that interval. If the test set contains an example of the class that falls into the interval, it is not possible to code it with respect to the zero frequency assigned for the training set. This problem is well known in text compression (Witten and Bell, 1991), and a simple solution is to use the Laplace correction for the frequency counts. This solution also helps with sparsely populated regions of the domain, which are common in our situation because the cross-validation procedure is applied recursively to increasingly smaller regions. The Laplace correction helps to prevent the generation of unnecessary splits in these regions.

The estimated information gain for a k-way split of a numeric attribute is calculated in the same way as the classification error, but with $L_{ErrorRate}(S',S)$ replaced by

$$L_{InfoGain}(S',S) = -\frac{1}{|S|}\sum_{c=1}^{C}|S_c|\log\left(\frac{|S'_c|+1}{|S'|+C}\right) - \frac{1}{|S|}\sum_{i=1}^{k}\sum_{c=1}^{C}|S_{c,i}|\log\left(\frac{|S'_{c,i}|+1}{|S'_i|+C}\right) \qquad (2)$$

C is the number of classes, and $|S_c|$ is the number of elements of class c in the test data. The "+1" and "+C" in the numerator and denominator respectively are the Laplace correction. In the second expression, we sum over the C classes and then the k subintervals. $|S_{c,i}|$ is the number of elements of class c in the ith partition for the test data, while $|S'_{c,i}|$ is the corresponding figure for the training data.

An argument, illustrated in Figure 1, was presented above for the use of information gain rather than classification error for generating splits. This argument assumes that the split will be followed by further splits. However, in the situation depicted in Figure 2a, all successors are leaf nodes (marked black). In this case the multiway split is used directly to classify instances, and should therefore be generated using the classification error rather than the entropy (see Brodley, 1995b, for a discussion of this problem in the context of decision trees with binary splits on numeric attributes). The situation in Figure 2b, where not all the successors are leaf nodes, is not so clearcut.

One solution to this problem is to identify multiway splits corresponding to nodes which have only leaves as successors, and rebuild them (recursively) using the error-based criterion. However, we found that this modification did not significantly change the accuracy of the resulting tree; moreover, it does not address situations like the one of Figure 2b. In the end we adopted the admittedly *ad hoc* solution of (recursively) rebuilding multiway splits using the error-based criterion if they send more than 50% of training instances directly to leaf nodes. The merit of this procedure will be shown later when we present our experimental results.

Minimum description length

The resampling estimates discussed above are computationally demanding. It is not enough to generate one multiway split for a node. With n-fold cross-validation, n extra splits have to be generated. For fixed n this does not affect the asymptotic complexity; nevertheless it affects the time taken to build decision trees in practice.

Minimum description length techniques allow one to measure performance on the training set, and then apply a correction based on the complexity of the split to estimate the true information gain or classification error. We use Kononenko's (1995) formulation of the description length for the selection of nominal attributes, and introduce a modification to handle numeric attributes that penalizes splits with many subintervals. This modification is similar in spirit to Quinlan's (1996) penalty for binary splits on numeric attributes.

The MDL evaluation of a k-way split on a numeric attribute is

$$L_{MDL}(S) = \log(m_{att}) + \log\binom{m_{att}-1}{k-1} + \sum_{i=1}^{k}\left[\log\binom{|S_i|+C-1}{C-1} + \log\binom{|S_i|}{|S_{1,i}|,\ldots,|S_{1,C}|}\right], \quad (3)$$

where m_{att} is the number of distinct values for that attribute in the dataset. Note that there is no division into training and test sets. The third term is Kononenko's for a nominal attribute; the first two are the extra penalty for a k-way split.

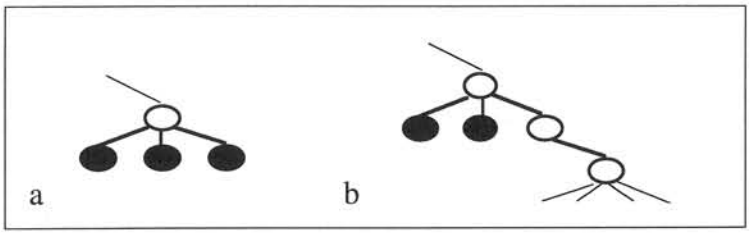


Figure 2: Two situations where multiway splits are used for classification

Gain ratio

Quinlan's (1993) gain ratio heuristic was applied by Fulton *et al.* (1995) for selecting multiway splits on numeric attributes. It differs from the other criteria in that it always opts to expand a node, because the gain ratio for the no-split option is zero. For this reason it cannot be used for pre-pruning. We employ it to provide a basis of comparison with Fulton *et al.*'s work.

The gain ratio evaluation of a k-way split on a numeric attribute is

$$L_{GainRatio}(S) = \frac{L_{InfoGain}(S)}{-\frac{1}{|S|}\sum_{i=1}^{k}|S_i|\log\left(\frac{|S_i|}{|S|}\right)} \tag{4}$$

where $L_{InfoGain}(S)$ is as in equation (2), called with both arguments the same.

Missing values

Some of our test datasets have substantial numbers of missing values. To handle these we adopt the procedure employed by the two-level decision tree algorithm T2 (Auer *et al.*, 1995): a missing value is considered as just another value of the attribute. As a result, trees contain missing-value branches for each internal node. In contrast to T2, we allow nodes corresponding to a missing-value branch to be expanded further. This does not pose any problem for the evaluation of the criteria defined above. The MDL formula treats the missing value for a nominal attribute just like any other value. In the case of a numeric attribute, the coding cost for the missing value is added to the coding cost induced by the other branches.

4 Empirical Results

We ran a set of experiments to compare the four criteria described above for recursive model selection. We compare them by analyzing the accuracy and size of the trees that they produce when run on standard datasets. We used fifteen datasets from the UCI repository, summarized in Table 1.

Methods compared

The methods we employ for generating and selecting the multiway splits are:

I-CV Information gain used as the splitting criterion; split selected using ten-fold cross-validation estimate of the information gain.

E-CV Error rate used as the splitting criterion; split selected using ten-fold cross-validation estimate of the classification error.
MDL Information gain used as the splitting criterion; split selected using MDL.
GR Information gain used as the splitting criterion; split selected using Quinlan's gain ratio.

In each case, the indicated splitting criterion—either classification error or information gain—was used to find the best k-way split on a numeric attribute, for each possible value of k. We used a dynamic programming algorithm to find the optimal k-way split (Fulton *et al.*, 1995). Its complexity is quadratic in the number of instances. It computes the optimal k-way split for all k up to k_{max}, the number of potential splitpoints, in time linear in k_{max}. Since it works for any additive impurity measure it can be used for both classification error and information gain. We also performed experiments using a greedy algorithm with runtime linear both in the number of instances and in k_{max} (Nevill-Manning *et al.*, 1995), and found that the accuracy of the resulting decision trees did not differ significantly. However, since the purpose of this paper is to compare different criteria for selecting multiway splits, we only present results for the optimal algorithm.

The indicated selection method was employed to choose one of the possible ks for each numerical attribute, and finally to choose between the different (numerical or nominal) attributes. Splitting proceeded recursively until the indicated method

Dataset	Instances	Missing values (%)	Nominal attributes	Numeric attributes	Classes
More than two classes					
AD audiology	226	2.0	69	1	24
PT primary-tumor	339	3.9	17	0	22
SY soybean	307	6.6	35	0	19
AU autos	205	1.1	10	15	7
ZO zoo	101	0.0	16	1	7
AN anneal	898	65.0	32	6	6
GL glass	214	0.0	0	9	6
LY lymphography	148	0.0	15	3	4
IR iris	150	0.0	0	4	3
Two classes					
CR credit-rating	690	0.6	9	6	2
G2 glass*	163	0.0	0	9	2
GE german	1000	0.0	13	7	2
HO horse-colic	368	23.8	15	7	2
LA labor-negotiations	57	35.7	8	8	2
PI pima-indians	768	0.0	0	8	2

**With classes 1 and 3 combined and classes 4 to 7 deleted.*

Table 1: Datasets used for the experiments

preferred a model that did not split at all—except in the case of GR, in which a full tree is built.

The first two methods, I-CV and E-CV, were chosen in order to compare entropy-based with error-based recursive model selection directly. The third, MDL, was chosen so that the entropy-based resampling method could be compared directly to one that achieves the same aim by penalizing the complexity of a model—as Fayyad and Irani's (1993) does, though in a less direct way. The fourth, GR, was chosen to provide a comparison with Fulton *et al.*'s (1995) work. They employed the gain ratio for evaluation but used classification error instead of information gain as splitting criterion in one experiment, and a modified, additive, version of the gain ratio in another. We chose the information gain because we found in subsidiary experiments that it gives at least as good results as the classification error; we could not use their version of the gain ratio because it is not described in their paper. In any event, we felt that GR captured the essence of their approach in a more natural way.

We noted earlier that multiway splits used for classifying instances directly should be selected by an error-based criterion. For this we use:

I/E-CV First, build a decision tree using I-CV. Then, for nodes which transfer 50% of training instances directly to leaf nodes, replace the multiway branch by one generated using E-CV. Recursively apply this procedure to a parent node after it has been applied to all children.

Finally, for comparison, we present results using Quinlan's (1993) C4.5, with default parameter settings.

Experimental results

The results produced by these six different methods on the datasets are summarized in Tables 2 and 3. Table 2 shows the classification errors, averaged over ten ten-fold cross-validation runs, along with the standard deviations. Table 3 shows the corresponding tree sizes in terms of number of nodes, including leaves. The same folds were used to evaluate each method.

First, we compare entropy-based with error-based multi-split selection, I-CV *vs* E-CV. I-CV produces more accurate trees when the dataset contains more than two well-represented classes. For four of these datasets (AD, ZO, AN, GL) it performs significantly better and for one significantly worse (LY);[2] inspection of LY reveals that only two of the four classes are well-represented (one class is represented by

[2]Throughout, we speak of results being "significantly different" if the difference is statistically significant at the 95%-level according to a paired *t*-test.

only one instance, another by four). For datasets with just two classes the situation is less clear: E-CV performs significantly better on four (CR, G2, GE, HO) and significantly worse on one (LA). However, for the first four the absolute difference in accuracy is always small—less than 4%—whereas on LA, which contains a high percentage of missing values, it exceeds 12%. For these datasets the anticipated advantage of I-CV, that it is more likely to discover important attribute interactions, seems to be offset by the fact that E-CV chooses multiway splits with smaller number of split-points and is therefore less likely to overfit the data.

Comparing the tree sizes produced by the same two methods, one might expect E-CV to build smaller trees since the set of potential split points for the classification error as splitting criterion is smaller than for the information gain. This is in fact the case for all datasets with two classes. With more than two well-represented classes the effect is less pronounced, perhaps because I-CV discovers important attribute interactions that remain undetected by E-CV.

We now turn to a comparison between entropy-based cross-validation with the minimum description length formula for multi-split selection, I-CV *vs* MDL. MDL's accuracy is significantly lower for five datasets with more than two well-represented classes (AD, PT, SY, AU, GL), and significantly higher for two (ZO, IR). In the two-class case its accuracy is significantly higher for three datasets (CR, HO, PI)–for four if LY is included–and significantly lower for two (G2, LA). For all datasets except LA MDL produces smaller trees. Trees generated by MDL for datasets with more than two well-represented classes seem to underfit the data—too

Dataset	I-CV	E-CV	MDL	GR	I/E-CV	C4.5
≥ three classes						
AD	22.4±0.8	27.1±1.8	27.4±1.1	23.7±1.6	23.1±2.1	22.8±1.0
PT	56.9±1.5	58.8±2.7	61.1±1.6	63.8±1.4	55.1±1.0	58.6±1.8
SY	22.4±2.4	23.4±1.5	24.6±1.0	9.4±0.9	13.6±0.7	16.2±1.5
AU	23.0±1.6	24.8±3.1	38.9±2.9	22.7±2.6	25.5±3.0	24.5±2.2
ZO	5.1±1.0	14.3±1.6	3.8±1.0	4.6±1.3	12.2±1.5	8.2±1.3
AN	0.6±0.2	4.1±0.7	0.8±0.3	0.9±0.2	1.2±0.2	7.5±0.7
GL	31.2±1.7	36.2±3.1	35.0±1.4	35.1±1.4	32.8±2.3	31.9±1.3
LY	25.4±2.0	23.4±1.4	22.6±1.0	23.1±2.4	23.8±1.1	24.9±1.8
IR	5.8±0.6	5.5±1.1	5.2±0.7	5.1±0.8	5.5±1.1	4.9±0.7
Two classes						
CR	16.4±0.9	14.8±0.7	14.1±0.4	19.6±1.1	14.9±0.9	15.6±0.9
G2	21.9±2.3	19.6±2.3	28.4±1.6	23.8±3.9	20.3±1.7	23.9±2.6
GE	30.8±1.2	27.5±0.8	30.3±0.8	40.0±0.6	28.3±0.9	28.6±0.9
HO	20.9±1.5	17.0±1.0	19.6±1.1	20.6±1.8	15.5±1.0	15.2±1.1
LA	5.8±1.7	17.9±2.8	12.5±2.5	14.1±3.4	12.8±3.1	15.1±1.7
PI	30.0±1.5	29.0±1.2	26.8±1.0	35.8±1.0	28.0±1.2	28.4±1.3

Table 2: Average accuracy for the test datasets (mean ± standard deviation)

Dataset	I-CV	E-CV	MDL	GR	I/E-CV	C4.5
≥ three classes						
AD	75.4±1.2	62.1±2.2	62.5±1.9	138.0±1.3	72.4±2.3	51.8±0.7
PT	100.3±3.7	48.4±4.1	75.6±4.6	491.9±3.0	62.9±3.4	79.0±1.8
SY	80.1±2.4	107.6±2.4	71.3±1.3	141.4±1.0	88.4±1.2	80.7±1.6
AU	104.1±9.8	69.4±2.5	59.1±2.7	90.6±2.3	70.5±1.3	62.7±2.3
ZO	40.7±3.0	51.0±16.3	29.6±0.6	29.4±0.5	51.3±16.3	15.5±0.2
AN	50.1±1.0	72.7±2.6	46.0±2.1	49.2±1.2	47.2±1.5	65.1±2.3
GL	52.1±1.6	46.4±2.3	15.0±0.3	96.6±2.2	48.7±2.4	51.8±2.2
LY	48.0±1.7	20.5±1.4	43.9±2.7	93.2±2.2	21.3±1.9	27.9±0.9
IR	9.7±0.7	6.5±0.7	7.3±0.2	23.4±0.4	6.5±0.7	9.0±0.4
Two classes						
CR	211.8±10.5	44.5±3.1	113.2±12.2	190.3±2.9	78.4±4.8	56.3±5.5
G2	32.3±1.3	19.6±1.5	9.7±0.6	49.6±1.5	22.7±2.0	28.3±1.4
GE	660.9±11.0	79.5±7.7	439.6±14.5	442.0±1.3	222.1±9.2	156.0±6.4
HO	118.9±2.4	41.0±3.7	105.1±2.8	112.4±1.9	44.0±4.7	17.6±1.7
LA	15.4±0.3	9.2±0.5	19.4±0.2	14.7±0.5	11.2±0.9	7.4±0.4
PI	175.0±3.7	58.9±11.9	12.8±0.8	322.9±3.6	102.5±8.8	128.1±4.5

Table 3: Average tree size for the test datasets (mean ± standard deviation)

high a penalty is paid for encoding the number of instances of each class in each interval. This is drastically demonstrated by the AU dataset where the average error for the MDL method is much higher and on average, trees are only one-third the size of those generated by I-CV. The disadvantage of MDL, compared to I-CV, is its lack of stability: for some datasets it produces results with much lower accuracy.

Comparing the gain ratio criterion with entropy-based selection using cross-validation, GR *vs* I-CV, the latter produces significantly higher accuracy on eight datasets (AD, PT, AN, GL, CR, GE, PI) and significantly lower accuracy on three (SY, LY, IR). GR does not always produce larger trees as one would expect (because no prepruning is applied)—indeed, it produces considerably smaller trees on four datasets. However, on CR and GE this results in much lower accuracy. It produces larger trees on eight datasets.

Entropy-based selection combined with error-based selection near the leaves, I/E-CV—compared to I-CV alone—decreases the tree size in all but two cases (SY, ZO) while showing similarily high accuracy. Its accuracy is significantly better than I-CV's on eight datasets (PT, SY, LY, CR, G2, GE, HO, PI), and significantly worse on five (AU, ZO, AN, GL, LA).

Multiway trees *vs* binary trees

Table 2 also includes results for C4.5's pruned trees (Quinlan, 1993), which are decision trees with binary splits on numeric attributes. The accuracy of the

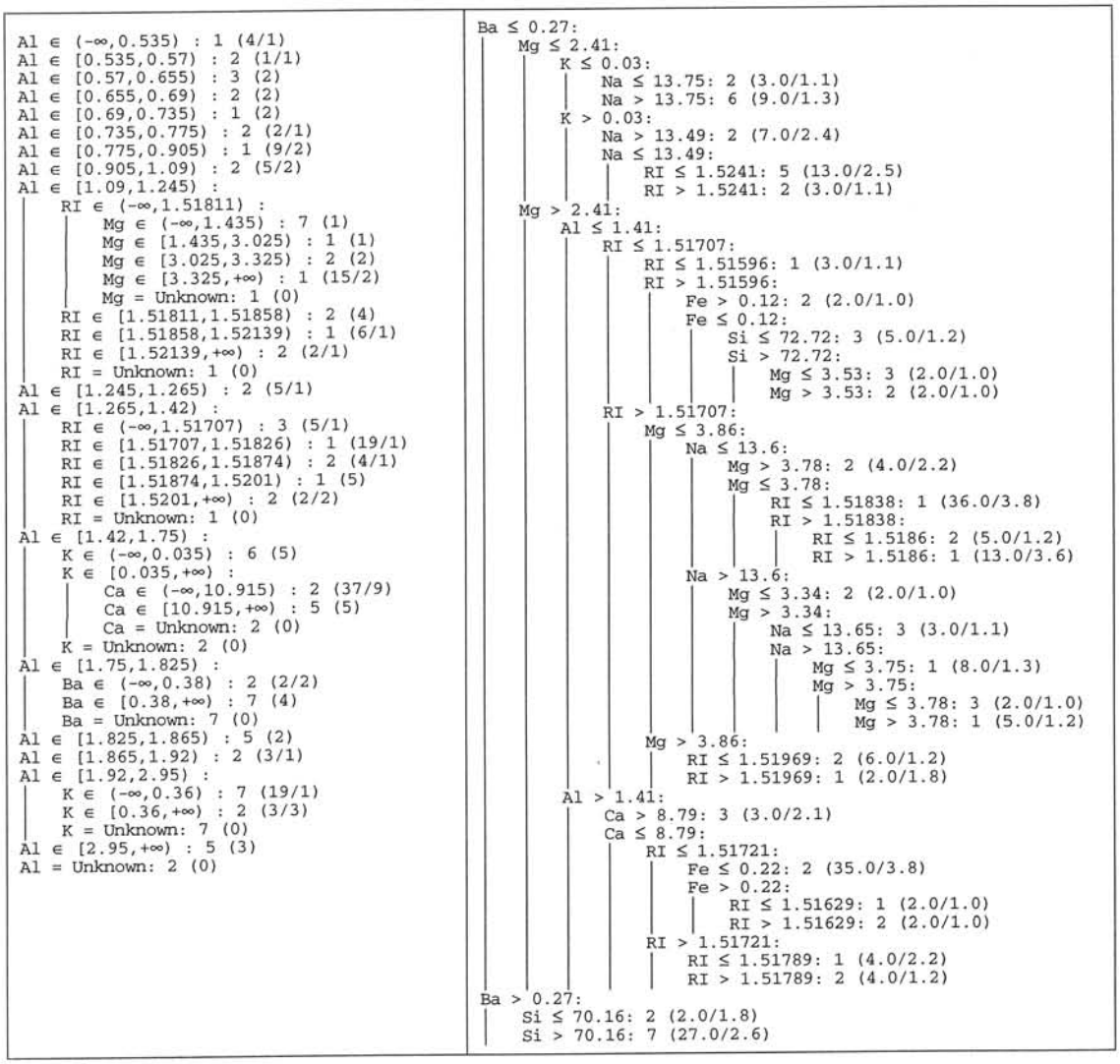
combined method is comparable to C4.5's accuracy. On five datasets it performs significantly better (PT, SY, AN, G2, LA), on one dataset significantly worse (ZO). For AN and LA the better results are probably due to the different handling of missing values. On seven datasets the trees are smaller than C4.5's pruned trees; on the remaining eight they are larger.

Two factors, both involving the different treatment of unknown values, contribute to the fact that multiway trees are larger than C4.5's trees for half the datasets. First, each internal node of the multiway tree has an "unknown value" branch even if the training data does not contain any missing values, in case missing values occur in test examples. Trees generated for the GL dataset are shown in Figure 3. The multiway tree contains eight leaves corresponding to branches on an unknown value, inflating tree size by more than 15%. Second, if a dataset contains missing values, the multi-split methods frequently expand an "unknown value" node into a subtree. Trees for the LA dataset consist mainly of a subtree of this type, and they give higher accuracy than do C4.5 trees. The same argument accounts for the high accuracy on the AN dataset.

Figure 3 also illustrates why multiway trees are easier to comprehend and analyze. In C4.5's tree on the right, which has 57 nodes, attributes frequently occur at several different levels in the same path—one attribute (Mg) occurs at as many as five different levels. These repetitive occurrences make it hard to see how an attribute influences the outcome, and nearly impossible to assess the relevance of a specific attribute to the decision process. The I/E-CV tree on the left, with 47 nodes, is more perspicuous.

5 Conclusion

This paper has presented a method, I-CV, for generating small and accurate decision trees with multiway splits. It operates by selecting splits based on their "true" information gain, estimated using a resampling procedure. This technique automatically incorporates pre-pruning. Experiments show that previously-used criteria, employed in the same manner, produce lower average accuracy on datasets with more than two classes, though their performance is comparable in two-class situations. When the new method is augmented with a post-processing step that uses classification error to rebuild the tree near the leaves, performance compares favorably with C4.5's pruned trees in terms of both accuracy and tree size. The new multiway trees have the advantage of greater comprehensibility because attributes rarely occur more than once in a path from root to leaf.



```
Al ∈ (-∞,0.535) : 1 (4/1)
Al ∈ [0.535,0.57) : 2 (1/1)
Al ∈ [0.57,0.655) : 3 (2)
Al ∈ [0.655,0.69) : 2 (2)
Al ∈ [0.69,0.735) : 1 (2)
Al ∈ [0.735,0.775) : 2 (2/1)
Al ∈ [0.775,0.905) : 1 (9/2)
Al ∈ [0.905,1.09) : 2 (5/2)
Al ∈ [1.09,1.245) :
|   RI ∈ (-∞,1.51811) :
|   |   Mg ∈ (-∞,1.435) : 7 (1)
|   |   Mg ∈ [1.435,3.025) : 1 (1)
|   |   Mg ∈ [3.025,3.325) : 2 (2)
|   |   Mg ∈ [3.325,+∞) : 1 (15/2)
|   |   Mg = Unknown: 1 (0)
|   RI ∈ [1.51811,1.51858) : 2 (4)
|   RI ∈ [1.51858,1.52139) : 1 (6/1)
|   RI ∈ [1.52139,+∞) : 2 (2/1)
|   RI = Unknown: 1 (0)
Al ∈ [1.245,1.265) : 2 (5/1)
Al ∈ [1.265,1.42) :
|   RI ∈ (-∞,1.51707) : 3 (5/1)
|   RI ∈ [1.51707,1.51826) : 1 (19/1)
|   RI ∈ [1.51826,1.51874) : 2 (4/1)
|   RI ∈ [1.51874,1.5201) : 1 (5)
|   RI ∈ [1.5201,+∞) : 2 (2/2)
|   RI = Unknown: 1 (0)
Al ∈ [1.42,1.75) :
|   K ∈ (-∞,0.035) : 6 (5)
|   K ∈ [0.035,+∞) :
|   |   Ca ∈ (-∞,10.915) : 2 (37/9)
|   |   Ca ∈ [10.915,+∞) : 5 (5)
|   |   Ca = Unknown: 2 (0)
|   K = Unknown: 2 (0)
Al ∈ [1.75,1.825) :
|   Ba ∈ (-∞,0.38) : 2 (2/2)
|   Ba ∈ [0.38,+∞) : 7 (4)
|   Ba = Unknown: 7 (0)
Al ∈ [1.825,1.865) : 5 (2)
Al ∈ [1.865,1.92) : 2 (3/1)
Al ∈ [1.92,2.95) :
|   K ∈ (-∞,0.36) : 7 (19/1)
|   K ∈ [0.36,+∞) : 2 (3/3)
|   K = Unknown: 7 (0)
Al ∈ [2.95,+∞) : 5 (3)
Al = Unknown: 2 (0)
```

```
Ba ≤ 0.27:
|   Mg ≤ 2.41:
|   |   K ≤ 0.03:
|   |   |   Na ≤ 13.75: 2 (3.0/1.1)
|   |   |   Na > 13.75: 6 (9.0/1.3)
|   |   K > 0.03:
|   |   |   Na > 13.49: 2 (7.0/2.4)
|   |   |   Na ≤ 13.49:
|   |   |   |   RI ≤ 1.5241: 5 (13.0/2.5)
|   |   |   |   RI > 1.5241: 2 (3.0/1.1)
|   Mg > 2.41:
|   |   Al ≤ 1.41:
|   |   |   RI ≤ 1.51707:
|   |   |   |   RI ≤ 1.51596: 1 (3.0/1.1)
|   |   |   |   RI > 1.51596:
|   |   |   |   |   Fe > 0.12: 2 (2.0/1.0)
|   |   |   |   |   Fe ≤ 0.12:
|   |   |   |   |   |   Si ≤ 72.72: 3 (5.0/1.2)
|   |   |   |   |   |   Si > 72.72:
|   |   |   |   |   |   |   Mg ≤ 3.53: 3 (2.0/1.0)
|   |   |   |   |   |   |   Mg > 3.53: 2 (2.0/1.0)
|   |   |   RI > 1.51707:
|   |   |   |   Mg ≤ 3.86:
|   |   |   |   |   Na ≤ 13.6:
|   |   |   |   |   |   Mg > 3.78: 2 (4.0/2.2)
|   |   |   |   |   |   Mg ≤ 3.78:
|   |   |   |   |   |   |   RI ≤ 1.51838: 1 (36.0/3.8)
|   |   |   |   |   |   |   RI > 1.51838:
|   |   |   |   |   |   |   |   RI ≤ 1.5186: 2 (5.0/1.2)
|   |   |   |   |   |   |   |   RI > 1.5186: 1 (13.0/3.6)
|   |   |   |   |   Na > 13.6:
|   |   |   |   |   |   Mg ≤ 3.34: 2 (2.0/1.0)
|   |   |   |   |   |   Mg > 3.34:
|   |   |   |   |   |   |   Na ≤ 13.65: 3 (3.0/1.1)
|   |   |   |   |   |   |   Na > 13.65:
|   |   |   |   |   |   |   |   Mg ≤ 3.75: 1 (8.0/1.3)
|   |   |   |   |   |   |   |   Mg > 3.75:
|   |   |   |   |   |   |   |   |   Mg ≤ 3.78: 3 (2.0/1.0)
|   |   |   |   |   |   |   |   |   Mg > 3.78: 1 (5.0/1.2)
|   |   |   |   Mg > 3.86:
|   |   |   |   |   RI ≤ 1.51969: 2 (6.0/1.2)
|   |   |   |   |   RI > 1.51969: 1 (2.0/1.8)
|   |   Al > 1.41:
|   |   |   Ca > 8.79: 3 (3.0/2.1)
|   |   |   Ca ≤ 8.79:
|   |   |   |   RI ≤ 1.51721:
|   |   |   |   |   Fe ≤ 0.22: 2 (35.0/3.8)
|   |   |   |   |   Fe > 0.22:
|   |   |   |   |   |   RI ≤ 1.51629: 1 (2.0/1.0)
|   |   |   |   |   |   RI > 1.51629: 2 (2.0/1.0)
|   |   |   |   RI > 1.51721:
|   |   |   |   |   RI ≤ 1.51789: 1 (4.0/2.2)
|   |   |   |   |   RI > 1.51789: 2 (4.0/1.2)
Ba > 0.27:
|   Si ≤ 70.16: 2 (2.0/1.8)
|   Si > 70.16: 7 (27.0/2.6)
```

Figure 3: Multiway and binary trees for the GL dataset: I/E-CV *vs* C4.5

Despite its good performance, the method for combining selection based on information gain and classification error is unsatisfactory, and further work on integrating the two is likely to produce better results. Another open issue is how well our pre-pruning performs in direct comparison to a post-pruning procedure.

Acknowledgments

All datasets used are from the repository maintained by the Department of Information and Computer Science at the University of California at Irvine.

Datasets LY and PT were collected at the University Medical Center, Institute of Oncology, Ljubljana, Slovenia, by M. Soklic and M. Zwitter.

References

Auer, P., Holte, R., Maass, W. (1995): "Theory and Applications of Agnostic PAC-Learning with Small Decision Trees," *Proc. 12th Int. Conf. on Machine Learning*.

Brodley, C. (1995a): "Recursive Automatic Bias Selection for Classifier Construction," *Machine Learning*, Vol. 20, pp. 63–94.

Brodley, C. (1995b): "Automatic Selection of Split Criterion During Tree Growing Based on Node Location," *Proc. 12th Int. Conf. on Machine Learning*.

Elomaa, T. & Rousu, J. (1996): "Finding Optimal Multi-Splits for Numerical Attributes in Decision Tree Learning," NeuroCOLT Technical Report, NC-TR-96-041.

Fayyad, U. & Irani, K. (1993): "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. 13th Int. Joint Conf. on Artificial Intelligence*, pp. 1022–1027.

Fulton, T., Kasif, S. & Salzberg, S. (1995): "Efficient Algorithms for Finding Multiway Splits for Decision Trees," *Proc. 12th Int. Conf. on Machine Learning*, pp. 244–251.

Kearns, M., Mansour, Y., Ng, A. & Ron, D. (1995): "An Experimental and Theoretical Comparison of Model Selection Methods," *Proc. 8th Annual Conf. on Computational Learning Theory*, Santa Cruz, CA, pp. 21–30.

Kohavi R.(1995): "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, pp. 1137–1143.

Kohavi R., Sahami M. (1996): "Error-Based and Entropy-Based Discretization of Continuous Features," *Proc. 2nd Int. Conference on Knowledge Discovery & Data Mining*, pp.114–119.

Kononenko, I. (1995): "On Biases in Estimating Multi-Valued Attributes," *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, pp. 1034–1040.

Nevill-Manning, C., Holmes, G., Witten, I. (1995): "The Development of Holte's 1R Classifier," *Proc. Conf. on Artificial Neural Networks and Expert Systems*, Dunedin, New Zealand, pp. 239–242.

Quinlan, R. (1993): *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Quinlan, R. (1996): "Improved Use of Continuous Attributes in C4.5," *Journal of Artificial Intelligence Research*, Vol. 4, pp. 77-90

Witten, I.H. and Bell, T.C. (1991): "The zero-frequency problem," *IEEE Trans Information Theory*, Vol. 37, No. 4, pp. 1085–1094.