The Attribute Selection Problem in Decision Tree Generation

Usama M. Fayyad
AI Group M/S 525-3660
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Fayyad@aig.jpl.nasa.gov

Keki B. Irani
AI Laboratory
E.E.C.S. Department
The University of Michigan
Ann Arbor, MI 48109
Irani@engin.umich.edu

Abstract

We address the problem of selecting an attribute and some of its values for branching during the top-down generation of decision trees. We study the class of impurity measures, members of which are typically used in the literature for selecting attributes during decision tree generation (e.g. entropy in ID3, GID3*, and CART; Gini Index in CART). We argue that this class of measures is not particularly suitable for use in classification learning. We define a new class of measures, called C-SEP, that we argue is better suited for the purposes of class separation. A new measure from C-SEP is formulated and some of its desirable properties are shown. Finally, we demonstrate empirically that the new algorithm, O-BTree, that uses this measure indeed produces better decision trees than algorithms that use impurity measures.

Introduction

Empirical learning from examples is receiving considerable attention in terms of research and applications. Programs that learn from pre-classified examples aim at circumventing the knowledge acquisition bottleneck in the development of expert systems. The problem is due to the fact that human experts find it difficult to express their (intuitive) knowledge of a domain in terms of concise, correct situation-action rules. Empirical learning algorithms attempt to discover relations between situations expressed in terms of a set of attributes and actions encoded in terms of a fixed set of classes. By examining large sets of pre-classified data, it is hoped that a learning program may discover the proper conditions under which each action (class) is appropriate.

Learning algorithms typically use heuristics to guide their search through the large space of possible relations between combinations of attribute values and classes. A powerful and popular such heuristic uses the notion of selecting attributes that locally minimize the information entropy of the classes in a data set. This heuristic is used in the ID3 algorithm [13] and its extensions, e.g. GID3 [2], GID3* [6], and C4 [14], in CART [1], in CN2 [3] and others; see [11] and [6] for a general discussion of the attribute selection problem.

We focus our attention on algorithms that learn classifiers in the form of decision trees. Decision tree based approaches to concept learning are typically preferred because they constitute an efficient means for processing large training data sets. In addition, the final classifier produced is symbolic and therefore not difficult for domain experts to interpret (as opposed to a neural network or pattern recognition based approach). Furthermore, we have been involved in using top-down decision tree generators, especially the GID3* algorithm, in real-world industrial applications in semiconductor manufacturing [10] and in other domains such as astronomical data processing at Caltech [7].

In brief, a top-down, non-backtracking decision tree algorithm is given a data set of classified examples expressed in terms of a set of attributes. The attributes may be nominal (discrete, categorical) or continuous-valued (numerical). The algorithm first discretizes the continuous-valued attributes by partitioning the range of each into at least two intervals. For each discrete (or discretized) attribute, the algorithm first formulates a logical test involving that attribute. The test partitions the data into several subsets. For example, in ID3 [13] and C4 [14], the value of the attribute is tested, and a branch is created for each value of the attribute. In GID3* [5, 6], only a subset of the values may be branched on, while the remaining values are grouped together in one default branch. A selection criterion is then applied to select the attribute that induces the "best" partition on the data. Once selected, a branch for each outcome of the test involving that attribute is created. This creates at least two child nodes to the parent node, and the algorithm is applied recursively to each child node. The algorithm refrains from further partitioning of a given node when all examples in it belong to one class, or when no more tests for partitioning it can be formulated. Thus a leaf node predicts a class (sometimes probabilistically).

We claim that the single most important aspect that determines the behavior of a top-down non-backtracking decision tree generation algorithm is the attribute (test) selection criterion used. The most widely used attribute selection criteria appear in the form of average impurity measures. This is a family of measures designed to capture aspects of partitions of examples relevant to good classification. Earlier comparisons of selection measures compared measures within the class of impurity measures (see below) and concluded that the choice of selection measure from within that class makes little difference [1]. Later in this paper we show that a new class of measures is needed.

Impurity Measures

Let S be a set of training examples with each example $e \in S$ belonging to one of the classes in $C = \{C_1, C_2, \ldots, C_k\}$. The class vector of S is a k-vector $\langle c_1, c_2, \ldots, c_k \rangle$, where each c_i is the number of examples in S that have class $C_i \in C$: $c_i = |\{e \in S | class(e) = C_i\}|$. Note that the class vector is a vector from the space $\mathcal{N}^k$, where $\mathcal{N}$ denotes the set of natural numbers. The *class probability vector* of S is the corresponding vector in $[0, 1]^k$:

$$\langle p_1, p_2, \ldots, p_k \rangle = \langle \frac{c_1}{|S|}, \frac{c_2}{|S|}, \ldots, \frac{c_k}{|S|} \rangle = \frac{1}{N} \langle c_1, c_2, \ldots, c_k \rangle.$$

A set of examples is said to be *pure* if all its examples belong to one class. Hence, if the class probability vector of a set of examples has a component with value 1, the set is pure. An extreme case of *impurity* occurs when all components of the class vector are equal. To quantify the notion of impurity, a family of functions known as impurity measures [1] is defined. We use ϕ to denote a function that assigns a merit value to a class probability vector.

Definition 1: Let S be a set of training examples having a class probability vector PC. A function $\phi : [0, 1]^k \rightarrow \Re$ such that $\phi(PC) \geq 0$ is an *impurity measure* if it satisfies the following conditions:

1. $\phi(PC)$ is minimum if $\exists i$ s.t. component $PC_i = 1$.
2. $\phi(PC)$ is maximum if $\forall i, 1 \leq i \leq k,\ PC_i = \frac{1}{k}$.
3. $\phi(PC)$ is symmetric with respect to components of PC.
4. $\phi(PC)$ is smooth (differentiable everywhere) in its range.

Conditions 1 and 2 of the definition are intended to fix the well-understood extreme cases. Condition 3 insures that the measure is not biased towards any of the classes. The fourth condition sometimes appears as a requirement that ϕ be convex downwards with respect to any of the components of PC. Usually this makes the analysis easier as well as providing desirable computational properties.

However, we need to evaluate the impurity of a partition induced by an attribute on a set of examples. Let $PC(S)$ be the class probability vector of S and let A be a discrete (or discretized) attribute defined over the set S. Assuming that A partitions S into the sets $S_1, \ldots, S_r$, the *impurity of the partition* is defined as the weighted average impurity of its component blocks:

$$\Phi(S, A) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} \phi\left(PC(S_i)\right).$$

Finally, the merit assigned to attribute A due to its partition of S is proportional to the reduction in impurity after the partition. Hence,

$$\Delta\Phi(S, A) = \phi(PC(S)) - \Phi(S, A).$$

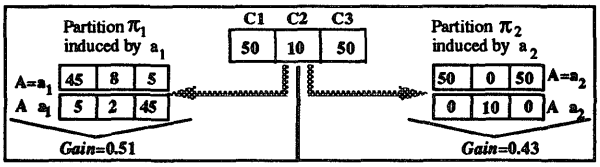

Figure 1: Two Possible Binary Partitions (3 classes).

It has been widely accepted that functions within this family are interchangeable for use in selecting attributes to branch on, and that they result in similar trees [1, 12]. This is not surprising since they all agree on the minima, maxima and smoothness. For example, if we assign

$$\phi(PC(S)) = \text{Ent}(S) = \sum_{i=1}^{k} -PC_i \log(PC_i)$$

then $\Phi(S, A)$ would be the entropy of the partition $E(A, S)$ and $\Delta\Phi(S, A)$ would correspond to the information gain $Gain(A, S)$ as defined in [1, 13]. Another impurity measure, used in [1] and claimed to result in trees similar to those resulting from information entropy minimization algorithms, is the *Gini index.* To obtain the Gini index we set ϕ to be

$$\phi(PC(S)) = \sum_{i \neq j} PC_i \cdot PC_j.$$

On the Suitability of Information Entropy

In this paper, we shall treat entropy as representative of the class of impurity measures. We shall examine the behaviour of entropy over the space of possible partitions and identify certain regions where entropy prefers partitions that do not conform with our intuitive expectation of how a "good" selection criterion should behave. Consider a set S of 110 examples of three classes { C1, C2, C3 } whose class vector is shown in Figure 1. Assume that the attribute-value pairs $\langle A, a_1 \rangle$ and $\langle A, a_2 \rangle$ induce the two binary partitions on S, π_1 and π_2, shown in the figure. Note that partition π_2 separates the classes C1 and C2 from the class C3. However, the information gain measure prefers partition π_1 ($gain = .51$) over π_2 ($gain = .43$).

Note that if partition π_2 is accepted, the subtree under this node has a lower bound of three leaves. On the other hand, the subtree under the node created for partition π_1 has a lower bound of six leaves. This does not necessarily imply that partition π_2 will generate a tree with a smaller number of leaves. However, unlike information entropy, intuitive evaluation of the two partitions clearly prefers π_2 to π_1 (assuming no further lookahead is allowed). If an attribute-value pair manages to isolate some classes from the rest, then it is clearly relevant to classification. As a matter of fact, if it were possible to obtain the tree with the minimal number of leaves, i.e. exactly one leaf for each of the k classes, then each node test will isolate some classes from others. If an attribute-value pair were not really correlated with the classes it isolates, then the probability of it causing a total class separation partition is lower than that for an overlapping classes partition. If the goal is to generate a tree with a smaller number of leaves, then we should expect the selection measure to be especially sensitive to total class separation. The case illustrated in Figure 1 gets worse when the number of classes grows (see [6] for further examples). The primary task of a classification learning algorithm is to separate differing classes from each other as much as possible, while separating as few examples of the same class as possible. If a partition is selected that does not separate differing classes apart, then the learning algorithm must find later partitions that do. Hence, by separating classes, the algorithm avoids postponing an action that sooner or later must be taken. The price for not taking it sooner is the likelihood of increasing the number of leaves in the tree. It is worthwhile noting that if the learning problem has exactly two classes, then the entropy heuristic no longer suffers from this problem, since class purity and class separation become the same. For this reason, in the binary class case, we believe that entropy is a good measure for classification.

The above discussion clearly illustrates that the information entropy heuristic is not sensitive to class separation, if there are more than two classes in the problem. Since a partition is evaluated by averaging the impurity of its component blocks, the entropy measure can only detect class separation indirectly. We argue that a "proper" selection measure for classification should compare the class vectors of the partition blocks directly. Other evidence for the inappropriateness of the entropy measure originates in our empirical experience with ID3, ID3-IV, GID3, and GID3*. ID3-IV uses the *gain ratio* rather than entropy as a selection measure (see [13].) GID3 and GID3* differ from ID3 and ID3-IV in that rather than branching on every value of the selected attribute, they branch on a few individual values while grouping the rest in one default branch. All these algorithms perform better than ID3. For example, we have very strong empirical evidence that GID3* consistently produces better trees than ID3 [5, 6]. How does this reflect on the merit of the entropy measure as a selection criterion? Our answer to this question takes the form of an argument illustrating that for any node in the tree, ID3 always selects the partition having the minimum entropy among all possible partitions using the selected attribute. In particular, we shall show that the partition selected by GID3* *is always* one that has an equivalent or *higher* entropy than the partition selected by ID3. Hence, if following the minimum entropy heuristic is a good idea, then ID3 should in principle generate better decision trees. Yet the empirical evidence points strongly to the contrary.

Given a data set at a node during decision tree generation, the partition π_1 selected by ID3 to partition the data at the node is a refinement of the partition selected by GID3* for the same node. This is obvious from the fact that GID3* branches on a subset of the individual values branched on by ID3. In addition, Given a data set at a node during decision tree generation, the partition π_1 selected by ID3 to partition the data at the node has the same or lower class information entropy than the corresponding partition π_2 selected by GID3* for the same node, as shown by:

Lemma 1 *If the partition induced on a set S by attribute A is a refinement of the partition induced on S by attribute A', then the class information entropy of the former is lower than that of the latter: $E(A, S) \leq E(A', S)$.*

Proof: See Lemma B.0.1 in [6]. □

Although ID3 always selects partitions having lower entropies than those selected by GID3*, empirical experience clearly indicates that this consistently leads to worse decision trees. The interpretation of this proposition is that although information entropy captures reasonable properties that make it useful for attribute selection, whatever it captures is not geared towards generating good trees. Whence, a method that consistently does not minimize entropy (GID3*) leads to the discovery of better trees. Other evidence for the correctness of this claim is presented as part of our discussion of the binary tree hypothesis (to be stated later).

A similar situation occurs with ID3-IV [13]. Although the *IV* measure and the gain-ratio were introduced by Quinlan as an *ad hoc* solution to overcome the problems with very fine partitions (large number of values), ID3-IV seems to outperform ID3. This gives us another instance of an algorithm that purposely avoids minimal entropy partitions and yet produces better trees.

Another reason to suspect the suitability of information entropy to the task of attribute selection derives from results in communication and information theory. It has been shown that the entropy minimization heuristic tends to yield decision trees with near-minimal average depth [8]. Although this may be a desirable property from a communication/vector quantization application perspective, we have strong empirical evidence indicating that trees with low average depth tend to have a large number of leaves and a higher error rate for the data sets encountered in a large variety of domains [4, 6]. We therefore claim that, from a machine learning perspective, the information entropy heuristic aims at an inappropriate goal: minimizing the average depth of a tree.

A consequence of Lemma 1 and entropy's preference for finer partitions is that the entropy measure is insensitive to within-class fragmentation: the separation of examples having the same classes. For example, consider two partitions of a set of examples, one (π_2) being a refinement of the other (π_1). Further assume that both partitions are pure, i.e. each of their component blocks consists of examples of the same class. Both evaluate equally under entropy—the average entropy of each partition being zero. However, the partition π_2 necessarily fragments examples from some class while partition π_1 does not. The extra fragmentation simply results in more leaves.

The information entropy measure suffers from two additional deficiencies. The first occurs in cases where the training set contains a majority of examples from one class, the *Gain* measure is then necessarily depressed away from its possible maximum *for all possible partitions.* This is because the entropy of each set, prior to partitioning, is low. In such situations various partitions approach each other in merit when evaluated under entropy.

The second defeciency with the entropy measure is what is referred to as the *information paradox* in [15]. The basic problem is that a set with a given class probability vector evaluates identically to another set whose class vector is a permutation of the first. Thus if one of the subsets of a set has a different majority class than the original but the distribution of classes is simply permuted, entropy will not detect the change. However, a major change in the dominant class is generally considered as evidence that the attribute value is actually relevant to classification. In general, entropy, like all other members of the family impurity measures, is insensitive to permutations in the components of the class probability vector. The reason for this is that it measures the impurity of each set in isolation. So the information paradox is detectable only indirectly when we average the entropies of the two sets in the partition.

In summary, we have listed above seven properties that we take as indications that the information entropy measure is not particularly well-suited for use as an attribute selection measure. It is our hypothesis that a measure that is better behaved than entropy, will lead to the generation of better decision trees. We shall formulate such a measure.

Binary Decision Trees

We have shown in [6] that for every decision tree, there exists a binary decision tree that is logically equivalent to it. Thus, exploring strictly binary trees does not reduce the space of possible decision trees that one may discover. At this stage we make a further claim:

The Binary Tree Hypothesis : For a top-down, non-backtracking, decision tree generation algorithm, if the algorithm applies a proper attribute selection measure, then selecting a single attribute-value pair at each node and thus constructing a binary tree, rather than selecting an attribute and branching on all its values simultaneously, is likely to lead to a decision tree with fewer leaves.

We have no formal proof of this hypothesis: only informal analysis and an empirical evaluation of it. Due to space constraints, we do not include the analysis. The empirical results supporting this hypothesis are given later in the paper as part of our comparison of all the algorithms. The interested reader is referred to the detailed presentation in [6].

What are the ramifications of this hypothesis on decision tree generation algorithms? We claim that it establishes a reasonable strategy for designing decision tree generation algorithms in general. Rather than branching on all attribute values of a selected attribute, branch on a single one—the "best" value. When we formulated the GID3* algorithm, we were considering the problem of deciding which subset of values of an attribute are relevant for classification when the information entropy is used as a measure of merit. The discussion presented above gives a different answer to the attribute-value selection problem: Rather than deciding which subset of values of an attribute is relevant based on whatever measure is being used, always select one value at a time; however, insure that the measure of attribute-value pair merit is a proper measure for classification.

Properties Desirable in a Selection Measure

Now that we have decided to generate strictly binary decision trees in which each branch test specifies a single attribute-value pair, we turn our attention to the design of a proper selection (merit) measure for attribute-value pairs. Rather than following the tradition of impurity measures and defining desirable properties with respect to a single set, we shall specify desirable properties with respect to a *partition on a set*.

Given a test τ on an attribute A and a set of training examples S, τ induces a binary partition on the set S into: $S = S_\tau \cup S_{\neg\tau}$, where $S_\tau = \{e \in S | e \text{ satisfies } \tau\}$, and $S_{\neg\tau} = S \sim S_\tau$.

We propose that a selection measure should in principle satisfy the properties:

1. It is maximum when the classes in S_τ are disjoint with the classes in $S_{\neg\tau}$ (*inter-class separation*).
2. It is minimum when the class distribution in S_τ is identical to the class distribution in $S_{\neg\tau}$.
3. It favours partitions which keep examples of the same class in the same block (*intra-class cohesiveness*).
4. It is sensitive to permutations in the class distribution.
5. It is non-negative, smooth (differentiable), and symmetric with respect to the classes.

This defines a family of measures, called C-SEP (for Class SEParation), for evaluating binary partitions.

By keeping as many examples of the same class together, we are aiming at leaves with high example support. This leads to better predictors and to a smaller number of leaves. Recall that the number of leaves and the expected error rate are related to each other, as shown in [4, 6], and that the training support per leaf serves as a semantic (vs. syntactic), estimate of rule generality [6].

Note that the conditions listed above force members of the family to *compare* class distributions directly since many of the properties are not detectable if each class vector is evaluated in isolation (c.f. impurity measures). The heuristic that is instantiated by selection measures that are members of the impurity measures family (including entropy) may be summarized as:

> *Favor the partition for which, on average, the distribution of classes in each block is most uneven.*

On the other hand, members of the C-SEP family of measures represent the following heuristic:

> *Favor the partition which separates as many different classes from each other as possible, and keeps examples of the same class together.*

Formulating a Selection Measure

For a k-vector V, ||V|| denotes the *magnitude* of V:

$$||\mathbf{V}|| = \sqrt{\sum_{i=1}^{k} V_i^2}.$$

Let $\mathbf{V}_1$ be the class vector of S_τ and $\mathbf{V}_2$ be the class vector of $S_{\neg\tau}$. In order to measure class overlap/separation directly, what should be done is to examine the "angle" between the two class vectors. In general k-space, what we need is a measure of the degree of *orthogonality* of the two vectors. Two vectors are orthogonal when their non-zero components do not overlap. We implicitly assume that the test τ is a *meaningful test* in that it induces a non-trivial partition on S, i.e., we implicitly assume that $S_\tau \neq S_{\neg\tau} \neq \emptyset$. Since our vectors are in $\mathcal{N}^k$, the angle is at a maximum when it is 90° and is minimum when it is 0°.

One measure of the angle is to take its cosine. The cosine of the angle between two vectors $\mathbf{V_1}$ and $\mathbf{V_2}$, $\theta(\mathbf{V_1}, \mathbf{V_2})$ is given by:

$$\cos\theta(\mathbf{V_1}, \mathbf{V_2}) = \frac{\mathbf{V_1} \circ \mathbf{V_2}}{\|\mathbf{V_1}\| \cdot \|\mathbf{V_2}\|}$$

where '$\circ$' represents the inner (dot) product:

$$\mathbf{V_1} \circ \mathbf{V_2} = \sum_{i=1}^{k} V_{1i} V_{2i}.$$

It is minimum when the two vectors are orthogonal and is maximum when they are *parallel*. Two vectors are parallel when one is a constant multiple of the other, whence the angle between them is zero.

Selection Measure ORT : For a set S of training examples and a test τ inducing a binary partition on S into S_τ and $S_{\neg\tau}$ having class vectors $\mathbf{V_1}$ and $\mathbf{V_2}$, respectively, the *orthogonality measure* is defined as

$$\mathrm{ORT}(\tau, S) = 1 - \cos\theta(\mathbf{V_1}, \mathbf{V_2}).$$

Note that this measure takes values in the range [0, 1]. A maximum value indicates that the vectors are orthogonal. We now show that the orthogonality measure possesses the desirable properties listed above.

Proposition 1 *The ORT measure possesses the following properties:*

1. *If the class probability vectors of the two sets in the partition are identical, then ORT is minimum.*
2. *If $ORT(\tau, S) = 0$ then the class probability vectors of the two sets in the partition of S are identical.*
3. *The ORT measure is maximum iff the classes in S_τ are disjoint from the classes that appear in $S_{\neg\tau}$.*
4. *The measure ORT favours partitions which keep like classes in the same subset.*

Proof: See [6]. □

Note that condition 2 for minimum is equivalent to the condition under which information gain is minimum:

Corollary 1 $Gain(\tau, S) = 0 \iff ORT(\tau, S) = 0.$

Proof: See [6]. □

Although the conditions under which *Gain* and ORT are minimum are equivalent, the conditions for maximum are not. This is where ORT possesses desirable properties that information entropy does not. Actually, the conditions under which ORT achieves its maximum value, 1, are more general than those under which entropy is as its minimum, 0:

Corollary 2 *Given a test τ that induces a binary partition on a set S of training examples containing more than one class, then $E(\tau, S) = 0 \implies ORT(\tau, S) = 1$.*

Proof: See [6]. □

Empirical Evaluation

We now turn to verifying empirically that the ORT measure indeed results in better trees. We name the binary tree algorithm that uses the ORT measure O-BTree. We turn our attention to comparing the performance of O-BTree with that of ID3, ID3-IV, GID3*, and ID3-BIN. ID3-BIN is simply the ID3 algorithm modified to branch on a single attribute-value pair at each node (hence generating strictly binary trees). We have earlier claimed that ID3-BIN should consistently outperform ID3 and ID3-IV. The results of this comparison will also demonstrate this fact as a side effect.

The data sets used for empirical evaluation were of two type: synthetic and real-world. The synthetic data was used since we have complete control over it so it can serve as a clean controlled test. A set of rules was constructed manually for diagnosing a well-understood portion of the Reactive Ion Etching (RIE) process in semiconductor manufacturing. The rules were verified physically and semantically by the domain experts. This set of rules was used to generate random examples. Each rule specifies the values of only a few of the available attributes. Since attributes not appearing in a rule's precondition are considered irrelevant to the classification task, random values are generated for those attributes in order to obtain examples. The goal of the learning program is to attempt to rediscover, or approximate, the original set of rules. This establishes a reference point for comparing the performance of the learning algorithms. The learning task contains 8 attributes (all discrete) and 6 classes. The classes are roughly equally likely. Hence the error rate of a "naive" classifier that always guesses the most common class for any example should not be lower than 83%.

In order to eliminate random variation, 10 independent experiments were conducted on 10 independently generated random RIE data sets[1]. The performance is measured by number of leaves and error rate. The error rates were collected by classifying examples in a separate fixed test set of 1886 examples.

The results reported will all be in terms of ratios relative to GID3* performance. Figure 2 shows the relative performance for the RIE random experiments. In this domain, better trees were generated by O-BTree. One note to make here is that although GID3* and O-BTree managed to discover trees with the minimal number of leaves on many occasions, the trees were actually different. As a matter of fact, O-BTree was able to discover the original (optimal) tree on some trials. This tree has a zero error rate.

The second type of data used consisted of a real-world application data set from semiconductor manufacturing (HARR90), and some publicly available data having only

[1] A training set consists of 150 examples on the average.

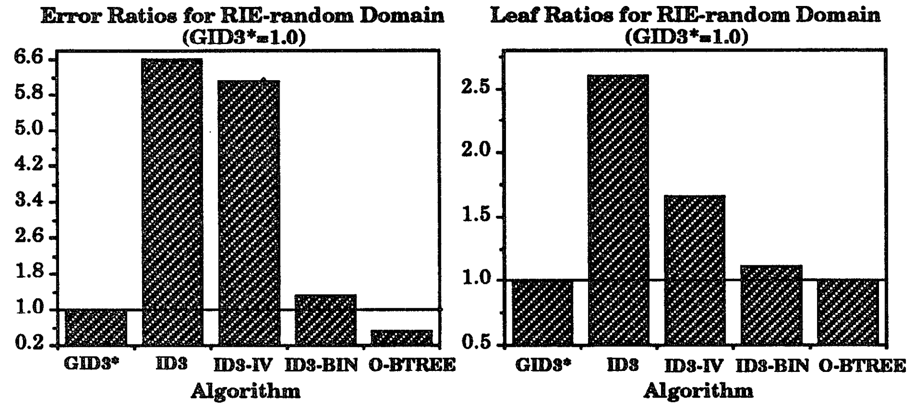

Figure 2: Performance of Various Algorithms (Relative to GID3*) in RIE Domain.

discrete attributes. These were the foreign imports automobile insurance data (AUTO), the soybean disease data sets (SOYBEAN) and the mushroom classification task (MUSHRM). We avoid using data that have continuous-valued attributes in order to avoid confusing the issue of attribute selection and continuous-valued attribute discretization which is performed prior to selection. We would like to isolate the effect of attribute selection as much as possible. The results for these domains are shown in Figure 3. A data set worth noting here is the SOYBEAN data. This data happens to have attribute-value pairs that induce total class separation at the very root of the tree. Although this did not decrease the number of leaves, the error rate of the tree generated by O-BTree was a little over half the corresponding error rate for ID3-BIN. We may therefore conclude that the ORT measure lead to more appropriate choices than those made by minimizing the information entropy at any stage (ID3-BIN).

Concluding Remarks

We have proposed a new family of measures, C-SEP, for use in selecting attributes during decision tree generation. The selection measures that are widely used in the decision tree induction literature typically use a selection measure from the family of impurity measures. We illustrated why we believe that the C-SEP family is more appropriate in the context of top-down decision tree generation. The main difference between the two families of measures is that impurity measures examine each subset in a partition separately without particular regard to cross-subset class overlap. They detect overlap indirectly by averaging over all the subsets. While indirect detection works well in the two-class case, it deteriorates as the number of classes grows.

Since members of the C-SEP family of measures evaluate a partition by comparing two class vectors, they require that partitions be strictly binary. This does not limit the expressive power of the trees produced. Furthermore, we hypothesize and informally argue that binary decision tree generation is likely to produce better trees than ID3-type branching. This hypothesis is verified empirically by comparing the performance of ID3 with its binary tree generating counterpart, ID3-BIN.

The family C-SEP, with the ORT measure being representative of its members, was defined using the same approach used in defining the family of impurity measures: specify where the maxima and minima should occur, and hope, with some assumptions of smoothness, that the behavior of the measure on the cases in-between the minima and maxima is “reasonable." The term “reasonable” implicitly means: correctly captures the aspects that make a partition good. The reason we make this implicit assumption is that, we, as designers of these measures, do not really know how to evaluate the partitions that constitute neither minimal nor maximal partitions according to the measure(s) being considered. Given a choice among several impure partitions, the entropy heuristic calls for favoring the partition in which the average class distribution is most uneven. On the other hand, faced with the same choices, the ORT measure favours the partition for which unlike classes overlap least and like classes overlap most. However, we have no clear reason to favor one measure over the other in those regions, since we do not know which is the better partition in the first place. What we did in this paper is point out that for some regions that are “well-understood” by us, the impurity measures fail to detect good partitions. This failure was a consequence of the fact that impurity measures are defined on single sets. We corrected this aspect by defining a new family of measures.

The proper approach to the selection measure design problem must first answer this question in some justifiable way: When can we say that one partition is better than another, meaning that it will eventually lead (or is likely to lead) to a tree with fewer leaves given the training data?

The intent of this paper is to point out that the problem of attribute selection is an important, and not yet adequately addressed problem. Future work here seems impossible without a formalism which allows us to answer the general question of what constitutes a better partition when generating a tree in a non-backtracking framework. We know how to answer the question if we perform a full lookahead search, but that is computationally infeasible. If there are

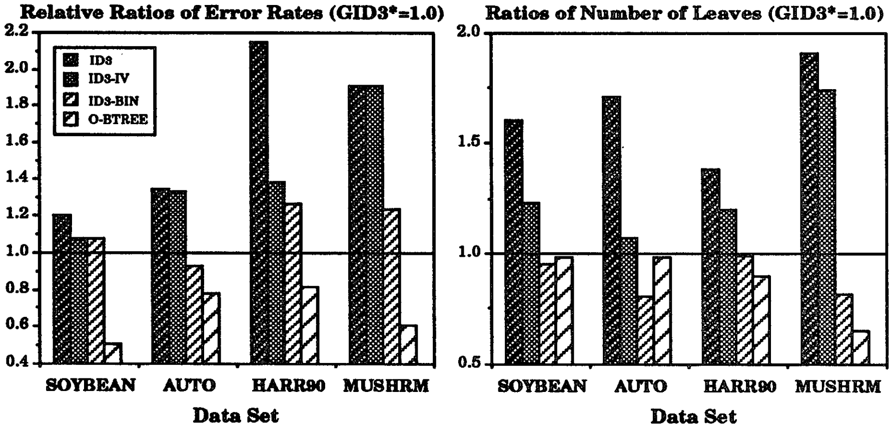

Figure 3: Performance of Various Algorithms (Relative to GID3*) over Several Domains.

K_v attribute-value pairs in the problem, and the minimal tree has n leaves, then an exhaustive search requires exploring at least $\binom{K_v}{n-1}$ possible trees, since the binary tree has $n-1$ internal (decision) nodes. A good selection measure should not be expected to find the minimal tree since this would make P=NP, a fact that we generally consider unlikely to be true. However, this does not rule out the possibility of solving the problem in the sense of formulating a measure that leads to minimal or near-minimal trees with high probability. Such an investigation is, of course, left as future work.

Acknowledgments

Thanks to R. Uthurusamy for valuable comments on this paper. The work described in this paper was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth & Brooks.

[2] Cheng, J., Fayyad, U.M., Irani, K.B., and Qian, Z. (1988). "Improved decision trees: A generalized version of ID3." *Proceedings of the Fifth International Conference on Machine Learning* (pp. 100-108). San Mateo, CA: Morgan Kaufmann.

[3] Clark, P. and Niblett, T. (1989). "The CN2 induction algorithm." *Machine Learning, 3*, 261-284.

[4] Fayyad, U.M. and Irani, K.B. (1990). "What should be minimized in a decision tree?" *Proceedings of the Eighth National Conference on Artificial Intelligence AAAI-90* (pp. 749-754). Cambridge, MA: MIT Press.

[5] Fayyad, U.M. and Irani, K.B. (1991). "A Machine Learning Algorithm (GID3*) for Automated Knowledge Acquisition: Improvements and Extensions." *General Motors Research Report CS-634*. Warren, MI: GM Research Labs.

[6] Fayyad, U.M. (1991). *On the Induction of Decision Trees for Multiple Concept Learning*. PhD dissertation, EECS Dept., The University of Michigan.

[7] Fayyad, U.M., Doyle, R., Weir, N. and Dgorgovski, S. (1992). "An automated data classification technique for reducing a very large astronomy data set." *Proceedings of ISY Conf. on Earth and Space Science Information Systems*. Pasadena, CA: NASA-JPL.

[8] Goodman, R. and Smyth, P. (1988) "Decision tree design from a communication theory standpoint." *IEEE Transactions on Information Theory* 34:5.

[9] Irani, K.B., Cheng, J., Fayyad, U.M., and Qian, Z. (1990). "Applications of Machine Learning Techniques in Semiconductor Manufacturing." *Proceedings of The S.P.I.E. Conference on Applications of Artificial Intelligence VIII* (pp. 956-965). Bellingham, WA: SPIE: The International Society for Optical Engineers.

[10] Irani, K.B., Cheng, J., Fayyad, U.M. and Qian, Z. (1992) "A Machine Learning Approach to Diagnosis and Control with Applications in Semiconductor Manufacturing." A chapter in *Intelligent Modeling, Diagnosis, and Control of Manufacturing Processes*. B. Chu and S. Chen (Eds). World Scientific Publishing.

[11] Lewis, P.M. (1962). "The characteristic selection problem in recognition systems." *IRE Transactions on Information Theory, IT-8*, 171-178.

[12] Mingers, J. (1989) "An empirical comparison of selection measures for decision-tree induction." *Machine Learning* 3:4 (319-342).

[13] Quinlan, J.R. (1986). "Induction of decision trees." *Machine Learning 1*, 81-106.

[14] Quinlan, J.R. (1990). "Probabilistic decision trees." In *Machine Learning: An Artificial Intelligence Approach, Volume III*, Y. Kodratoff & R. Michalski (Eds.) San Mateo, CA: Morgan Kaufmann.

[15] Smyth, P. and Goodman, R.M. (1991) "An information theoretic approach to rule induction from databases." *IEEE Transactions on Knowledge and Data Engineering*, to appear.