Tree-Structured Classification Via Generalized Discriminant Analysis

WEI–YIN LOH and NUNTA VANICHSETAKUL*

The problem of constructing classification rules that can be represented as decision trees is considered. Each object to be classified has an associated **x** vector containing possibly incomplete covariate information. Tree construction is based on the information provided in a "learning sample" of objects with known class identities. The **x** vectors in the learning sample may have missing values as well. Procedures are proposed for each of the components of classifier construction, such as split selection, tree-size determination, treatment of missing values, and ranking of variables. The main idea is recursive application of linear discriminant analysis, with the variables at each stage being appropriately chosen according to the data and the type of splits desired. Standard statistical techniques used as basic building blocks include analysis of variance, linear and canonical discriminant analysis, and principal component analysis. A new method of tree-structured classification is obtained by assembling the pieces. This method can accommodate prior probabilities as well as unequal misclassification costs and can yield trees with univariate, linear combination, or linear combination with polar coordinate splits. The method is compared with the CART method of Breiman, Friedman, Olshen, and Stone (1984). Some of the operational differences are that the new method (a) can have multiple splits per node, (b) is nonrandomized, (c) uses a direct stopping rule, (d) handles missing values by estimation, (e) allows both ordered and unordered variables in the same linear combination split, (f) is not invariant of monotone transformations of the individual variables, and (g) is computationally faster. Simulation experiments suggest that the two methods have comparable classification accuracy. The Boston housing data are analyzed in a classification context for illustration.

KEY WORDS: Cross-validation; Decision tree; Discriminant analysis; Machine learning; Misclassification; Missing values; Principal components; Recursive partitioning.

1. INTRODUCTION

1.1 The Problem and Classical Solutions

We have a k vector of measurements $\mathbf{x} = (x_1, \ldots, x_k)'$ on an object belonging to one of J classes, and wish to predict its class label. A "learning sample" of n other objects whose **x** vectors and class labels are known is assumed available. This problem has been variously called discrimination (Hand 1981), identification (Gordon 1981, p. 3), and classification (Breiman, Friedman, Olshen, and Stone 1984). We adopt the latter term. Its scope is very broad and examples include the following: (a) remote sensing of crops using high altitude photographs, (b) medical diagnosis based on health history and vital measurements, (c) speech recognition via waveform data, (d) ship identification from radar profiles, (e) analysis of chemical compounds via mass spectra, and (f) weather prediction using past data (e.g., see Breiman et al. 1984; Hand 1981).

To predict the class of an object a classification rule is needed. This is constructed using the information in the learning sample and any given prior probabilities and misclassification costs. When the class probability densities are known, the best rule is the maximum likelihood or Bayes rule. Most practical methods, however, explicitly or implicitly estimate the densities from the data.

The earliest and best-known method is linear discriminant analysis (LDA). If the **x** vectors come from normal distributions with equal covariance matrices, LDA is asymptotically Bayes. When equal priors and constant misclassification costs are assumed and the data are sphericized, LDA partitions the **x** space into J disjoint portions, with each portion containing one sample class mean such that every point in a portion is closer to the mean it contains than to the other means (e.g., see Gnanadesikan 1977, chap. 4). Other methods, such as density estimation and nearest-neighbor techniques, estimate the class densities nonparametrically. Although effective, these methods have been criticized (see Breiman et al. 1984, p. 17) for their (a) dependence on the metric $\|\mathbf{x}\|$ used; (b) inability to treat categorical variables and missing observations naturally; (c) high computational cost, since the learning sample must be recalled every time a new object is classified; and (d) limited function as "black boxes," yielding little information about the data.

1.2 The CART Method

The CART method of Breiman et al. (1984) addresses these criticisms by having a binary decision tree as a classifier. The latter is obtained by recursively partitioning the learning sample, which induces a corresponding partition of the **x** space. At each stage the sample is split according to the answers to questions such as "Is $x_i \le c$?" (univariate split), "Is $\sum_i a_i x_i \le c$?" (linear combination split), and "Does $x_i \in A$?" (if x_i is a categorical variable). The method searches over essentially all constants c, coefficients $\{a_i\}$, and subsets A to find the best split, with the

* Wei–Yin Loh is Associate Professor, Department of Statistics, University of Wisconsin, Madison, WI 53706. Nunta Vanichsetakul is Instructor, Department of Statistics, Kasetsart University, Bangkok 10900, Thailand. Loh's research was partially supported by National Science Foundation Grants MCS 83-00140 and DMS 85-02303, U.S. Air Force Office of Scientific Research Grant ISSA 86-0068, and U.S. Army Contract DAAG 29-80-C-0041. Vanichsetakul's research was partially supported by the University of Wisconsin Graduate School. This article was revised during Loh's IBM Junior Faculty Research Fellowship at the T. J. Watson Research Center. All of the computations were done on the University of Wisconsin Department of Statistics computers. The authors thank an associate editor and two referees for their helpful comments.

Journal of the American Statistical Association
September 1988, Vol. 83, No. 403, Theory and Methods

goodness of a split measured by how much it decreases the impurity of the subsamples.

Splitting stops at a node when it is pure or contains less than a prespecified number of cases. The bottom nodes are then recombined or "pruned upwards" to give the final tree. The amount pruned is determined by cross-validation (CV) using a cost-complexity function that balances the apparent error rate with the tree size. A by-product of this procedure is a CV estimate of the error rate. Besides the best split, a set of surrogate or alternative splits on other variables are obtained at each node to direct cases with missing values down the tree. Finally, a ranking of the importance or discriminatory power of the variables is calculated, with the importance of a variable measured by how effectively the surrogate splits based on it decrease tree impurity. Classifying new objects is therefore rapid, the learning sample is not needed afterwards, and the tree structure provides additional information about the data.

1.3 The Proposed Methods

Though flexible and powerful, CART has some less desirable properties. (a) Being based on sort-and-search principles, it can be slow with large data sets. (b) It is typically not more accurate than LDA (Breiman et al. 1984, sec. 5.7). Further, the dual use of CV for error estimation and tree construction means that (c) the CV estimate is not genuine, (d) run-time is not saved if the estimate is waived, and (e) the tree is randomized if fewer than n-fold CV is used, because its size then depends on the random-number seed used to form the CV samples (10- or 25-fold CV is common for practical reasons).

We propose and investigate alternative procedures for each of the main steps of classification-tree construction. The goal is an algorithm sharing the best features of LDA and CART, namely the speed of linear techniques and the visual information of decision trees. One immediate possibility for a tree with linear combination splits is to recursively partition the $\mathbf{x}$ space using linear discriminant functions. But this inevitably leads to almost singular covariance matrices in the subsamples as they become more homogeneous and reside in subspaces. Further, some simple data sets are not amenable to linear partitions, such as two spherically distributed classes with one class containing the other (see Sec. 3.4). We address the first problem by a dimensional reduction through principal components, and the second by transforming to polar coordinate splits with a suitably chosen origin. To obtain univariate splits (which is really where tree-structured representation is humanly comprehensible), we use univariate F ratios for variable selection and linear discriminant analysis to partition the selected coordinate axis. Finally, to avoid a randomized solution (and save computations when CV error estimation is not desired), we employ a direct stopping rule.

2. METHODOLOGICAL DETAILS

2.1 Priors and Misclassification Costs

Let $\{\pi(j), j = 1, \ldots, J\}$ be the class priors, either estimated or given. The estimated posterior probabilities $\{p(j \mid t), j = 1, \ldots, J\}$ at node t are then $p(j \mid t) = p(j, t)/\sum_i p(i, t)$, where N_j is the number of class j objects in the sample, $N_j(t)$ is the number of class j objects in t, and $p(j, t) = \pi(j)N_j(t)N_j^{-1}$. Let $C(i \mid j)$ be the cost of misclassifying a class j object as class i, and assume that $C(i \mid j) = 0$ if $i = j$, and is nonnegative otherwise. The CART method deals with unequal misclassification costs in two ways. One (symmetric Gini) makes the cost matrix symmetric. The other (altered priors) converts the misclassification costs into unit costs by altering the priors to

$$\pi'(j) = C(j)\pi(j) \Big/ \sum_i C(i)\pi(i), \tag{1}$$

where $C(j) = \sum_i C(i \mid j)$.

We propose a third option for dealing with unequal misclassification costs that is intimately related to our use of discriminant functions to split a node. It is called the *normal theory* option (described in Sec. 2.2). It does not give linear splits with nonconstant misclassification costs.

2.2 Splitting Rule

We use a modified version of LDA to generate linear combination splits. To avoid near-singular covariance matrices, a principal component analysis of the correlation matrix is done at each node. Linear discriminant functions are calculated from those principal components whose eigenvalues exceed β times the largest eigenvalue (β is user-specified and is .05 in our examples).

Specifically, a split is selected at node t via the discriminant functions

$$d_j(\mathbf{y}) = \hat{\boldsymbol{\mu}}_j'\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{y} - \tfrac{1}{2}\hat{\boldsymbol{\mu}}_j'\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_j + \ln\{p(j \mid t)\}, \tag{2}$$

where $\mathbf{y}$ denotes a vector in the space of the larger principal components, $\hat{\boldsymbol{\mu}}_j$ is the sample mean vector of the jth class, and $\hat{\boldsymbol{\Sigma}}$ is the pooled estimate of the covariance matrix at the node. Each node is split into J subnodes, and an object is channeled into the ith subnode if the latter minimizes the estimated expected misclassification cost:

$$\sum_{j=1}^{J} C(i \mid j)\exp\{d_j(\mathbf{y})\} = \min_{1 \le m \le J} \sum_{j=1}^{J} C(m \mid j)\exp\{d_j(\mathbf{y})\}.$$

This is the aforementioned normal theory option. [It is the optimal strategy if $\mathbf{y} = \mathbf{x}$ and we have normal densities in the node (see Anderson 1984). Greer (1979) and Vapnik (1982, chap. 10) discussed optimal strategies for restricted families of rules according to various optimality criteria. Their strategies require linear and discrete programming for implementation.] The same formulas apply to the altered-priors option as well, except that then $C(i \mid j)$ would be taken as 1 for $i \neq j$, and the priors changed according to (1).

A univariate split is found in two steps. First, we find the variable x_{i^*} (say) with the largest F ratio of between- to within-classes variance. If this F ratio is greater than or equal to F_0 (a user-specified threshold), we use the aforementioned discriminant functions to split the node, with the quantities in (2) referring to values along this coordinate only. (We usually use $F_0 = 4$, because it coincides with the F-to-enter value in the stepwise discriminant analysis program in BMDP.) If the largest F ratio is less than F_0 (suggesting that the class means are close relative to the spreads), we look for splits based on dispersion: Each x_i is transformed into $z_i = |x_i - \bar{x}_i|$, where $\bar{x}_i$ is the node mean for the ith coordinate, and the F ratios based on the z_i's are computed. Suppose that $z_{i^{**}}$ has the largest F ratio, say $F_{i^{**}}$. If $F_{i^{**}} \geq F_0$, the node is split using (2) with $\mathbf{y} = z_{i^{**}}$, and so on; this typically produces $2J - 1$ subnodes, because whenever the condition $a \leq |x - c| < b$ along a branch is a union of two disjoint intervals, it is split into two branches, each of the form $d < x < e$. Otherwise, if $F_{i^{**}} < F_0$, the node is split into two according as $x_{i^*} \leq$ or $> \bar{x}_{i^*}$ (a guard against premature stopping).

The preceding steps are ineffective when there is angular or radial symmetry in the node. Therefore, we propose polar coordinate splits as a third split option. With this, the same steps are followed, except that a transformation to polar coordinates is made whenever spherical symmetry is detected. Let $\mathbf{y}$ again be the vector consisting of the larger principal component coordinates. Suppose that y_{i^*} has the largest F ratio F'_{i^*} of between- to within-class variances among the y_i's. If $F'_{i^*} \geq F_0$, a linear combination split is used at the node. Otherwise, each y_i is transformed into $w_i = |y_i - \bar{y}_i|$. Suppose that $w_{i^{**}}$ has the largest F ratio, say $\tilde{F}_{i^{**}}$. If $\tilde{F}_{i^{**}} < F_0$, we simply split the node in two according as $y_{i^*} \leq$ or $> \bar{y}_{i^*}$, again to avoid early stopping—an example where this is effective is two-dimensional data distributed at the corners of a square with sides parallel to the axes, such that the data for one class are located at a pair of diagonally opposite corners and that for the second class at the remaining corners. Otherwise, if $\tilde{F}_{i^{**}} \geq F_0$, then some spherical symmetry probably exists among the x variables, and a split on polar coordinates could be effective. The symmetry, however, may not be present in every variable. To weed out the "noise" variables, Levene's (1960) homogeneity test of the x_i variances is performed. (Consider data in three dimensions, with the classes uniformly distributed along concentric cylindrical shells around one axis; the variable corresponding to this axis would be found insignificant by Levene's test and identified as noise.) Suppose that m variables are found significant. If $m = 1$, the node is split on $w_{i^{**}}$. If $m = 0$, all of the x variables are transformed to polar coordinates $(r, \theta_1, \ldots, \theta_{k-1})$. If $m > 1$, only the significant x variables are transformed to polar coordinates $(r, \theta_1, \ldots, \theta_{m-1})$. In each of the latter two cases, the best univariate split from the w's, r and θ's is found. The formulas of Watson (1983, pp. 5–6) for the average and dispersion of a set of angles are used in computing the F ratios for the θ variables.

Categorical variables are transformed into ordered ones. If x_i, say, is a categorical variable with c categories, it is converted into $c - 1$ dummy variables. The largest *discriminant coordinate* (CRIMCOORD value; also called *canonical variate,* see Gnanadesikan 1977, p. 86), say u_i, from this $(c - 1)$-dimensional space is obtained. Each $(c - 1)$-dimensional dummy vector is mapped into a one-dimensional u_i. Finally, x_i is replaced by u_i in the sample. The 0–1 nature of the dummy variables makes it straightforward to reexpress the split "Is $u_i \geq c$?" to the form "Does $x_i \in A$?", where A is a subset of categories. Using only the first CRIMCOORD ensures that the dimension of the variable-space is not increased.

The CART method handles categorical variables by searching through all possible subsets A for each x_i. Because it does not convert a categorical variable to an ordered one, the two variable types cannot appear together in a linear combination split; that is, a split is either on a linear combination of ordered variables or on a single categorical variable. The method proposed here can mix variable types.

2.3 Stopping and Node Assignment Rules

The CART method uses CV to prune a large tree to its eventual size. We adopt the direct stopping rule: Stop splitting if either the node apparent error rate does not decrease with splitting, or there is at most one class in the node with sample size $\geq$ MINDAT. MINDAT is user-specified and analogous to a corresponding parameter in CART. Specifically, let $t_1, \ldots, t_J$ be the daughter nodes of t if it is split, and let $l(t)$ denote the class that will be assigned to node t if it is declared terminal. The first part of the rule stops splitting at t if

$$\sum_{i=1}^{J} C(l(t) \mid i) p(i \mid t) \leq \sum_{j=1}^{J} \left\{ \sum_{i=1}^{J} C(l(t_j) \mid i) p(i \mid t_j) \right\}.$$

Following CART, we assign node t to class i if the latter minimizes the estimated expected misclassification cost, that is, if $\sum_{j=1}^{J} C(i \mid j) p(j \mid t) = \min\{\sum_{j=1}^{J} C(m \mid j) p(j \mid t) : 1 \leq m \leq J\}$. [Another assignment method was given in Vanichsetakul (1986).]

3. EXAMPLES

This section compares a FORTRAN implementation of the proposed method [called FACT (fast algorithm for classification trees)] with the commercial CART (1984) computer program (also coded in FORTRAN) in five simulated situations. Equal priors and (except for Sec. 3.6) unit misclassification costs are assumed. All of the data in both the learning and test samples are complete (i.e., without missing observations). Unless stated otherwise, the Gini criterion is used in CART, and MINDAT = 5, except for the waveform recognition problem, where it is 10. The timings include tenfold CV for both methods. The times are the central processing unit times on a Pyramid 90X superminicomputer with a floating point accelerator and running 4.2 BSD UNIX.

The first example consists of normally distributed data, the second and third examples are from Breiman et al. (1984), the fourth is from Friedman (1977), and the fifth has multiple-valued categorical variables.

3.1 Normal Discrimination Problem

There are 10 variables and 3 classes. The data consist of normally distributed $\mathbf{x}$ vectors with identity covariance matrix and mean vectors $(0, 0, 0, \ldots, 0)$, $(3, 0, 0, \ldots, 0)$, and $(3/2, 3\sqrt{3}/2, 0, \ldots, 0)$. Table 1 gives the results. The error rates for the two methods are not significantly different, but the new method is 16 times faster

Table 1. Example 3.1: Normal Problem

	Univariate splits			*Linear combination splits*		
Criterion	*CART*	*FACT*	*SE/Speed*	*CART*	*FACT*	*SE/Speed*
CV error estimate	.15	.13	±.02	.11	.11	±.02
Test-sample error estimate	.15	.16	±.01	.15	.14	±.01
Run time	3.5 m	12.7 s	16.5	34 m	69.7 s	29.3

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART. $F_0 = 4$, $k = 10$, $J = 3$, and $n = 300$. There are 3,000 test samples, equal priors, unit misclassification costs, complete samples, and 10-fold CV. The asymptotic Bayes rate = .1153.

with univariate splits and 30 times faster with linear combination splits. Stepwise LDA using BMDP7M (Dixon et al. 1983) took 22s and gave a test-sample error estimate of .12 ± .01.

3.2 Waveform Recognition Problem

This example is from Breiman et al. (1984, pp. 49–55). There are 3 classes and 21 variables. Each class consists of a random convex combination of two triangular waveforms, with noise added. Specifically,

$$\begin{aligned} x_i &= uh_1(i) + (1 - u)h_2(i) + \varepsilon_i, && \text{Class 1} \\ &= uh_1(i) + (1 - u)h_3(i) + \varepsilon_i, && \text{Class 2} \\ &= uh_2(i) + (1 - u)h_3(i) + \varepsilon_i, && \text{Class 3,} \end{aligned}$$

where $i = 1, \ldots, 21$, u is a uniform random number on (0, 1), the ε_i's are independent standard normal noise variables, and the h_j's are shifted triangular waveforms such that $h_1(i) = \max(6 - |i - 11|, 0)$, $h_2(i) = h_1(i - 4)$, and $h_3(i) = h_1(i + 4)$. Breiman et al. reported a test-sample estimate of .14 for the Bayes error rate. Table 2 shows that the proposed method is better than CART for both accuracy and speed. [Breiman et al. (p. 134) reported a test-sample estimate of .20 for CART with linear combination splits. This differs from our .30. Another learning sample gave estimates of .24 for CART and .21 for our method.]

3.3 Digit Recognition Problem

This is another example of Breiman et al. (1984). There are seven 0–1 variables, indicating if a light is on or off in the seven lines of a digital display. (See the top left corner of Fig. 1.) Thus $J = 10$. Each light has probability .1 of not doing what it is supposed to do, independently of the others.

The learning sample comes from the CART demonstration package. The variables are specified as categorical and univariate splits are used. Figures 1 and 2 show the two trees (the CART tree is obtained with the "twoing" criterion). Although the proposed method splits every node into 10 subnodes, the 0–1 nature of the variables forces all of the samples down only two subnodes at a time. Table 3 shows the accuracy and speed. The slightly higher estimates of error for CART are probably due to one class not being assigned a terminal node. (See Table 10, Sec. 5, for the results with another learning sample.)

Table 2. Example 3.2: Waveform Problem

	Univariate splits			*Linear combination splits*		
Criterion	*CART*	*FACT*	*SE/Speed*	*CART*	*FACT*	*SE/Speed*
CV error estimate	.31	.31	±.03	.24	.21	±.02
Test-sample error estimate	.31	.27	±.01	.30	.20	±.01
Run time	7.6 m	25.2 s	18.1	55.2 m	2.4 m	23

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART. $F_0 = 4$, $k = 21$, $J = 3$, and $n = 300$. There are 2,000 test samples, equal priors, unit misclassification costs, complete samples, and 10-fold CV.

3.4 Spherical Distribution Problem

This example from Friedman (1977) has 2 classes and 10 variables. The first four variables of one class are distributed uniformly within a four-dimensional spherical slab centered at the origin, with inner radius 3.5 and outer radius 4.0; the last six variables are independent standard normal. The variables in the other class are distributed as 10-dimensional multivariate normal centered at the origin, with identity covariance matrix. Thus the last six variables are noise and the first class almost completely surrounds the second in the space containing the first four variables.

Table 4 shows that CART is more accurate than the proposed method if univariate splits are used. The better accuracy of CART is due to the "build a large tree, then prune" approach. With linear combination and polar coordinate splits, however, the proposed method was able

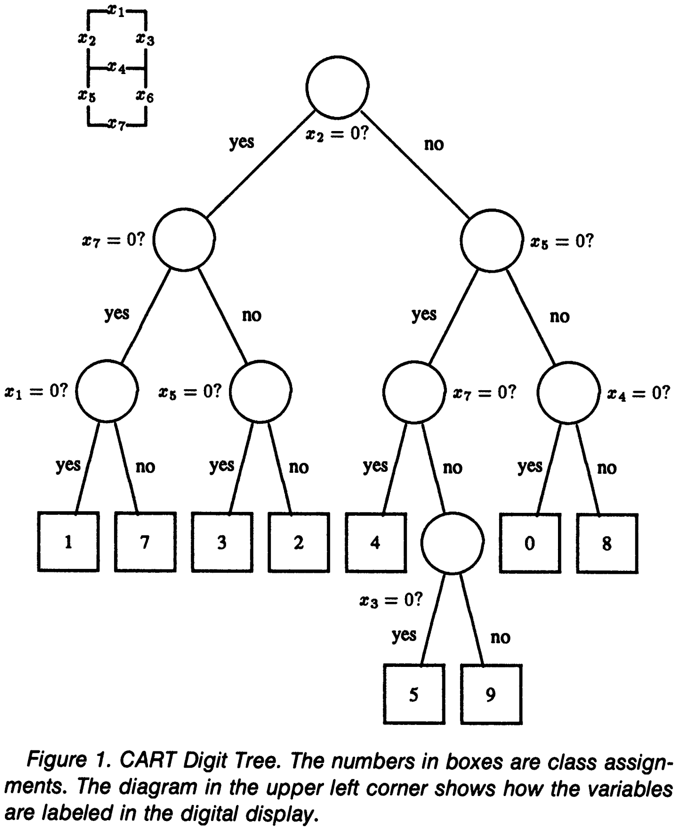

Figure 1. CART Digit Tree. The numbers in boxes are class assignments. The diagram in the upper left corner shows how the variables are labeled in the digital display.

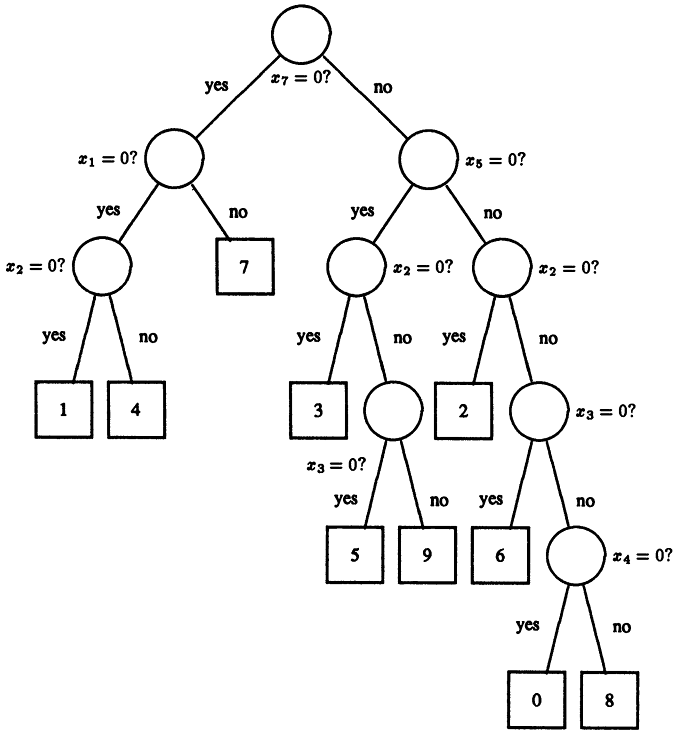

Figure 2. Digit Tree With the Proposed Method. The numbers in boxes are class assignments.

to detect the spherical structure and the tree split only once, on a radius. Note that CART is worse with linear combination splits than with univariate splits. This may be the result of pruning and the one-step optimality of the procedure; that is, the tree with linear combination splits is overpruned, and the best univariate split at the root node subsequently yielded better splits than the corresponding best linear combination split.

3.5 Categorical Variable Problem

This problem has three classes and five categorical variables. Variables $x_1, \ldots, x_5$ have 8, 3, 3, 3, and 10 categories, respectively. Each variable takes values 1, 2, and so forth, up to the number of its categories. The class distributions are

$$\text{Class 1: } \Pr(x_1 = i, x_2 = j, x_3 = k) = 1/72,$$

$$\text{Class 2: } \Pr(x_1 = i, x_2 = j, x_3 = k) = (i + j + k)/612,$$

$$\text{Class 3: } \Pr(x_1 = i, x_2 = j, x_3 = k) = (8 - i + j + k)/540.$$

Variables x_4 and x_5 are uniformly distributed noise. Table 5 shows that the two methods give close to the asymptotic Bayes rate. The program CART is slightly more accurate, but much slower.

3.6 Unequal Misclassification Costs

Table 3. Example 3.3: Digit Problem

	Univariate splits		
Criterion	*CART*	*FACT*	*SE/Speed*
CV error estimate	.31	.30	±.03
Test-sample error estimate	.36	.32	±.01
Run time	77.2 s	4.9 s	15.8

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART. F_0 = 4, k = 7, J = 10, and n = 200. There are 5,000 test samples, equal priors, unit misclassification costs, complete samples, and 10-fold CV. The asymptotic Bayes rate = .26.

We use two cost matrices given by Breiman et al. (1984, sec. 4.5) to examine the effect of unequal costs on the waveform problem. Table 6 gives the results, with cost matrix

$$C(i \mid j) = \begin{bmatrix} 0 & 5 & 5 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \qquad (3)$$

where i is the column and j the row index. Because the misclassification costs are constant for each row in (3), our method gives the same tree under either option. It is from 16 to 23 times faster than CART with univariate splits, and 13 to 18 times faster with linear combination splits. Our method is as accurate as CART for univariate splits, but its misclassification cost is about half CART's for linear combination splits. Table 7 gives qualitatively similar results, with the symmetric cost matrix

$$C(i \mid j) = \begin{bmatrix} 0 & 3 & 3 \\ 3 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix}. \qquad (4)$$

4. VARIABLE IMPORTANCE RANKING

A tree structure can sometimes look deceptively simple, and may cause one to think that only those variables that appear in the splits are important. Real data often exhibit proxy phenomena, where two or more variables measure essentially the same thing. With linear combination splits, this leads to difficulties with interpretation of the coefficients. With univariate splits, this produces the masking problem, because only one variable can appear at a time in a split.

Breiman et al. (1984) introduced the useful idea of an importance ranking of the variables via surrogate splits, to detect masking. Suppose that $s(t)$ is the best split at

Table 4. Example 3.4: Spherical Problem

	Univariate splits					
Criterion	*CART*	*FACT*	*SE/Speed*	*CART, LC*	*FACT, LC and P*	*SE/Speed*
CV error estimate	.08	.18	±.01	.12	.06	±.01
Test-sample error estimate	.09	.19	±.005	.12	.05	±.004
Run time	15 m	69.5 s	12.9	2 h	87 s	80

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART, LC is linear combination, and P is polar coordinate splits. F_0 = 6, k = 10, J = 2, and n = 1,000. There are 5,000 test samples, equal priors, unit misclassification costs, complete samples, and 10-fold CV. The asymptotic Bayes rate = .0063.

Table 5. Example 3.5: Categorical Problem

Criterion	Univariate splits CART	Univariate splits FACT	SE/Speed
CV error estimate	.59	.61	±.03
Test-sample error estimate	.57	.60	±.01
Run time	13.6 m	37.5 s	21.8

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART. $F_0 = 4$, $k = 5$, $J = 3$, and $n = 300$. There are 5,000 test samples, equal priors, unit misclassification costs, complete samples, and 10-fold CV. The asymptotic Bayes rate = .5795.

node t. A surrogate split $s_i(t)$ based on x_i is that split on x_i that best predicts $s(t)$. The importance of a variable is measured by how well these surrogate splits decrease tree impurity.

Instead of surrogate splits, we use ratios of variances to measure the importance of variables. If $F_{i,t}$ denotes the ratio of the between-classes to the total variance of variable x_i at node t, and $p(t)$ is the estimated probability that an object will fall into node t (i.e., the proportion of samples in t), then the importance of x_i is defined to be proportional to $\sum_t F_{i,t}p(t)$, with the sum over all nonterminal nodes and the proportionality constant chosen to make the largest importance value 100.

Table 8 shows that the two methods rank the variables differently for the digital problem, and suggests that no single variable is much more important than the others. Table 9 gives the ranking for the spherical distribution problem. Both methods are clearly able to identify the last six variables as noise.

5. MISSING OBSERVATIONS

Surrogate splits are used in CART to direct cases with missing observations down a tree. We replace missing values in the learning sample with class means estimated from the nonmissing values (see BMDPAM, from Dixon et al. 1983) prior to tree construction. To classify a new object with missing values, we propose two solutions, depending on the type of split. For a tree with linear combination or polar coordinate splits, all missing values are replaced at the root node by the respective coordinates of the class centroid closest to the object, in the space of the nonmissing coordinates. The distance of $\mathbf{x}$ from the jth centroid in node t is defined by

$$d(\mathbf{x}, j, t) = -\ln\{p(j \mid t)\} + \tfrac{1}{2} \sum_i \{\ln s_{ji}^2(t) + s_{ji}^{-2}(t)[x_i - \bar{x}_{j,i}(t)]^2\}, \quad (5)$$

where $\bar{x}_{j,i}(t)$ and $s_{ji}^2(t)$ are the sample mean and variance of the ith variable in the jth class in t, and the sum is over the nonmissing variables. The motivation is maximum likelihood, assuming normal densities. Categorical variables are included in (5) through their largest CRIMCOORD values. To estimate a missing categorical value x_i, the corresponding largest CRIMCOORD u_i is first estimated via (5). Then, x_i is replaced by the category whose largest CRIMCOORD is closest to u_i. The same procedure is used for a tree with univariate splits, except that a missing value is estimated not at t_0 but at the node in which it is first needed; the sum in (5) is then over all of the nonmissing variables, as well as any previously estimated variables.

Table 6. Waveform Problem

Criterion	CART AP	CART SG	FACT AP	FACT NT	SE/Speed
		Univariate splits			
CV error estimate	.35	.34	.63	.63	±.05
Test-sample error estimate	.54	.56	.54	.54	±.03
Run time	9.2 m	10.2 m	26.5 s	33.9 s	16–23
		Linear combination splits			
CV error estimate	.32	.32	.42	.42	±.06
Test-sample error estimate	.61	.61	.33	.33	±.03
Run time	1.1 h	1.5 h	5.1 m	5.1 m	13–18

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART, AP is altered priors, SG is symmetric Gini, and NT is normal theory. $F_0 = 4$, $k = 21$, $J = 3$, and $n = 300$. There are 2,000 test samples, equal priors, complete samples, and 10-fold CV. The cost matrix is Equation (3).

Table 7. Waveform Problem

Criterion	CART AP	CART SG	FACT AP	FACT NT	SE/Speed
		Univariate splits			
CV error estimate	.61	.59	.79	.55	±.07
Test-sample error estimate	.79	.72	.69	.72	±.03
Run time	6.8 m	10 m	48.2 s	46.6 s	8–13
		Linear combination splits			
CV error estimate	.53	.55	.55	.53	±.06
Test-sample error estimate	.61	.72	.47	.47	±.02
Run time	49.3 m	1.5 h	6.8 m	6.1 m	7–15

NOTE: SE is standard error and "speed" is the relative speed of FACT to CART, AP is altered priors, SG is symmetric Gini, and NT is normal theory. $F_0 = 4$, $k = 21$, $J = 3$, and $n = 300$. There are 2,000 test samples, equal priors, complete samples, and 10-fold CV. The cost matrix is Equation (4) of Section 3.6.

Table 10 shows the effect of missing values on the digit

Table 8. Variable Importance Ranking for the Digit Problem

CART Variable	CART Importance	FACT Variable	FACT Importance
x4	100	x3	100
x2	71	x5	82
x5	69	x7	80
x1	40	x4	76
x3	30	x1	54
x7	21	x2	49
x6	20	x6	40

Table 9. Variable Importance Ranking for the Spherical Problem

Univariate splits							
CART		FACT		Linear, CART		Linear and Polar, FACT	
Variable	Importance	Variable	Importance	Variable	Importance	Variable	Importance
× 4	100	× 2	100	× 2	100	× 4	100
× 2	91	× 1	92	× 3	98	× 3	75
× 3	85	× 3	79	× 4	91	× 2	66
× 1	61	× 4	62	× 1	89	× 1	64
× 8	12	× 7	1.1	× 9	13	× 7	1.1
× 9	11	× 8	.6	× 5	12	×10	.5
× 5	9	×10	.4	× 7	10	× 8	.2
×10	8	× 5	.3	×10	9	× 6	.1
× 7	7	× 6	.3	× 8	8	× 5	.1
× 6	5	× 9	.1	× 6	7	× 9	.0

recognition problem for a learning sample different from that of Table 3. Both the learning and test samples have $p\%$ of their values randomly deleted, with $p = 0, 5, 10, 25$. The test-sample estimates indicate that CART is better. Table 11 shows the results for the waveform recognition problem. Neither method dominates with univariate splits here, but the proposed method is more effective with linear combination splits.

As in Breiman et al. (1984), two further experiments were conducted to investigate the separate effects of missing values in the learning and test samples for the digit recognition problem. The first experiment consists of having $p\%$ of the data missing from the learning sample but not the test sample. The second is the opposite, with missing values in the test sample only. Table 12 indicates that for both methods, missing values are more damaging if they occur in the test sample than in the learning sample (see Breiman et al. 1984, table 5.5).

The tentative overall conclusion is that CART handles missing values better with univariate splits, and the proposed method is better with linear combination splits.

6. DISCUSSION

Our results show that the proposed method is fast and accurate relative to CART. We now discuss some implications of the differences in strategies.

1. *Splitting by F Ratios Versus Sorting.* The CART approach of exhaustively searching for splits allows it to select from a much larger class of splits than the proposed method. Nevertheless, CART's greater power in this respect is offset by one disadvantage. By optimizing a split on the data, CART may sometimes find things that are not really there. The method of splitting by F ratios alleviates this problem by restricting the class of splits. To make up for the restriction, simple transformations such as absolute deviations and polar coordinates are included. This may still be deficient in sufficiently complex situations. But because the sample is recursively partitioned, the handicap may not be crippling for most applications, and the simulations seem to bear this out—it is not crucial for every node to be split optimally, since subsequent splits have a chance to compensate (see the last sentence of Sec. 3.4).

Table 10. Cross-Validation and Test-Sample Error Estimates for the Digit Problem

	CART		FACT	
p	CV	TS	CV	TS
0	.29	.30	.31	.32
5	.33	.34	.31	.35
10	.36	.36	.35	.39
25	.47	.46	.52	.51

NOTE: TS is test sample and SE is standard error. The value p is the percentage missing in learning and test samples. $F_0 = 4$, $n = 200$, SE (CV) = .03, and SE (TS) = .01. There are univariate splits, equal priors, unit costs, and 10-fold CV.

2. *Top-Down Stopping Versus Pruning.* The stopping rule proposed here is similar to those in the AID and THAID algorithms (see Fielding 1977; Morgan and Messenger 1973), which are precursors of CART. The obvious advantage of the "build a large tree, then prune" approach is that it insures against stopping too early. A simple example is a two-dimensional two-class data set where the samples of one class are distributed on the white squares of a checkerboard and those of the second class on the black squares. This structure would be discovered only after a large tree is built, and a top-down stopping rule such as the one proposed here would not find it. One difficulty that this example illustrates is that although the CART approach may eventually find good splits in a highly complex situation, the resulting tree may not be easy to interpret. The numerous levels of nesting would require a fair amount of human scrutiny and intelligence to reveal the checkerboard structure of the data.

3. *Missing Value Estimation Versus Surrogate Splits.* The idea of surrogate splits is conceptually excellent. Besides solving the problem of missing values, surrogates can help identify the nodes where masking of specific variables occurs. Practical difficulties, however, can affect the way it is implemented. First, when linear combination splits are used, it is impractical to find at *every* node the best surrogate linear combination split for each possible subset of variables not a superset of those in the original split. (The CART program computes only univariate surrogate splits regardless of the split option.) Second, even the use of univariate surrogate splits for univariate splits may be insufficient. Figure 3 shows a two-class two-variable data set

Table 11. Cross-Validation and Test-Sample Error Estimates for the Waveform Problem

	Univariate splits				Linear combination splits			
	CART		FACT		CART		FACT	
p	CV	TS	CV	TS	CV	TS	CV	TS
0	.31	.31	.31	.27	.24	.30	.22	.20
5	.35	.33	.29	.31	.34	.31	.25	.20
10	.31	.35	.27	.35	.33	.34	.23	.20
25	.38	.37	.28	.42	.40	.40	.29	.22

NOTE: TS is test sample and SE is standard error. The value p is the percentage missing in learning and test samples. F_0 = 4, n = 300, SE (CV) = .03, and SE (TS) = .01. There are equal priors, unit costs, and 10-fold CV.

for which a binary split on x_1 is optimal. Nevertheless, no surrogate split on x_2 of the form "Is $x_2 \leq c$?" is effective. In contrast the missing-value algorithm in Section 5 predicts x_1 from x_2 satisfactorily. For surrogate splits to be effective here, they must contain more than just binary partitions of the x_2 axis.

4. *Categorical Variables.* Although the CART method of splitting on categorical variables appears natural, it may favor such variables over ordered ones. For example, an ordered variable x_1 taking m distinct sample values generates $m - 1$ possible splits of the sample of the form "Is $x_1 \leq \xi$?". On the other hand, an m-valued categorical variable x_2 generates $2^{m-1} - 1$ distinct splits of the form "Does $x_2 \in A$?". Therefore, when all other things are equal, x_2 is more likely to be split than x_1. Transforming a categorical variable to a CRIMCOORD variable ameliorates this condition by reducing the number of splits.

5. *Multiple Versus Binary Splits.* The use of binary splits has the following advantages: (a) the split can be characterized simply with a yes–no type question, (b) categorical variables can be handled as naturally as ordered variables, and (c) the idea of surrogate splits is more straightforward to implement than if each node is split into varying pieces. On the other hand, binary splitting can produce a highly nested tree, which may be difficult to comprehend because the brain needs to keep track of the many levels of conditioning (see Mingers 1986, p. 21). Multiple splitting is a double-edged solution: It may reduce the level of nesting, but it can also attenuate the subsamples so quickly that interesting information is not shown because the tree is too short. Section 7 contains another argument for multiple splits.

6. *Cross-Validation.* The principal benefits of separating CV from tree construction are that (a) a randomized solution is avoided, and (b) execution time is substantially decreased if the CV error estimate is skipped (e.g., in exploratory analyses).

Table 12. Test-Sample Error Estimates for the Digit Problem

	First experiment, p% missing in learning sample		Second experiment, p% missing in test sample	
p	CART	FACT	CART	FACT
0	.30	.32	.30	.32
5	.32	.32	.33	.35
10	.30	.32	.36	.39
25	.31	.31	.44	.50

NOTE: F_0 = 4 and standard error = .01. There are univariate splits, equal priors, unit costs, and 10-fold CV.

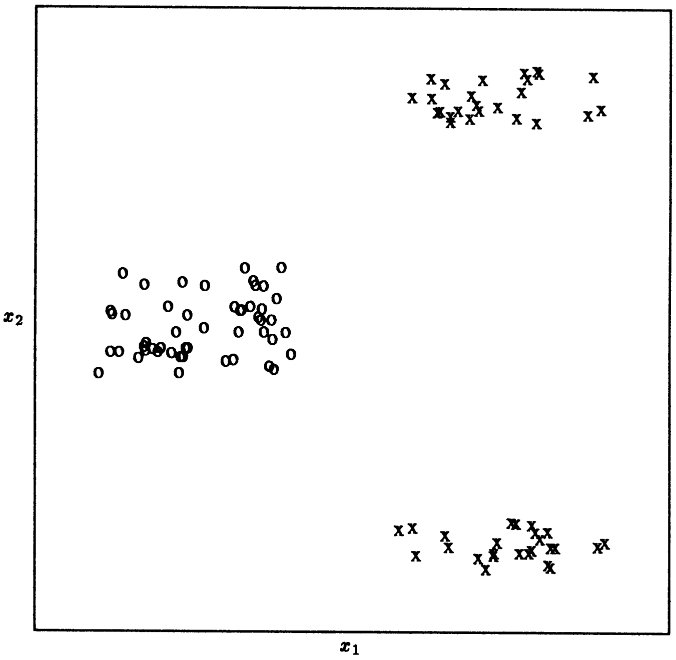

Figure 3. Data With Two Variables and Two Classes (o, x). The best split is on x_1; estimation of a missing x_1 value using x_2 and the algorithm in Section 5 is more effective than using a surrogate split based on x_2.

7. *Other Differences.* The relative merits of the other ingredients are not clear. The ability to mix ordered and categorical variables in a linear combination split removes to some extent the dichotomy between the two classes of variables. The richer class of splits that results may also make the method more powerful than CART when linear combination splits are requested. We do not know which method of dealing with unequal misclassification costs is best. Neither do we know which method of ranking the importance of variables is superior; both appear satisfactory from the simulation experiments, and ours is cheaper because it does not use surrogate splits. The implications of invariance versus noninvariance to transformations are discussed in the next two sections.

7. TREE INTERPRETATION AND THE EFFECT OF TRANSFORMATIONS: THE BOSTON HOUSING DATA

So far, we have emphasized accuracy and speed. We now examine the differences in interpretation between the trees from the two methods and the effect of transformations on them by analyzing the 1970 Boston housing data, reported by Harrison and Rubinfeld (1978). There are 506 cases (census tracts) and 14 variables, including the median value of homes in thousands of dollars (MV). The others are as follows: CRIM, crime rate; DIS, the weighted distance to employment centers; ZN, the percentage of land zoned for lots; CHAS, 1 if on Charles River, 0 otherwise; AGE, the percentage built before 1940; B, $(Bk - .63)^2$, Bk = the proportion of blacks in the population; INDUS, the percentage of nonretail business;

RAD, accessibility to radial highways; RM, the average number of rooms; NOX, nitrogen oxide concentration; TAX, tax rate; LSTAT, the percentage of lower-status population; and P/T, the pupil/teacher ratio.

To set this up as a classification problem, we categorize MV into three classes of roughly equal size: low (Class 1) if ln(MV) ≤ 9.84, high (Class 3) if ln(MV) > 10.075, and medium (Class 2) otherwise. Figures 4 and 5 show the two trees with univariate splits, and Table 13 gives their variable importance rankings. Both trees split on LSTAT first. This variable effectively splits the sample into two pieces in Figure 4 and five pieces in Figure 5. The next variables split are RM and NOX (CART), and RM and AGE (ours). The CART method splits NOX if LSTAT > 14.4, and ours splits AGE if LSTAT > 15.7. This observation and the importance rankings suggest that NOX and AGE are proxies for each other if LSTAT > 15.

Five of the predictor variables were transformed by Harrison and Rubinfeld (1978), namely, ln(LSTAT), RM^2, NOX^2, ln(RAD), and ln(DIS) (see Belsley, Kuh, and Welsch 1980, p. 231; Breiman et al. 1984, p. 218). Figure 6 and Table 13 show the results with these transformations. (Table 13 also gives the order in which BMDP7M entered the variables.) The tree is now shorter, and one of the cut points at the root node in Figure 6 is the same as the CART cut point in Figure 4. After transformations the CART tree is indistinguishable from Figure 4, to the accuracy shown (e.g., the cut point at the root node is 14.39 instead of 14.40), because CART is almost invariant of monotone transformations in the ordered variables when univariate splits are used. We say "almost" because although the learning sample is invariantly partitioned, the cut points at the nodes are not invariant (each is a midpoint between two ordered data values). Nevertheless, except for transformations that are highly nonlinear over the intervals between ordered data values containing the cut points, the CART tree is practically invariant.

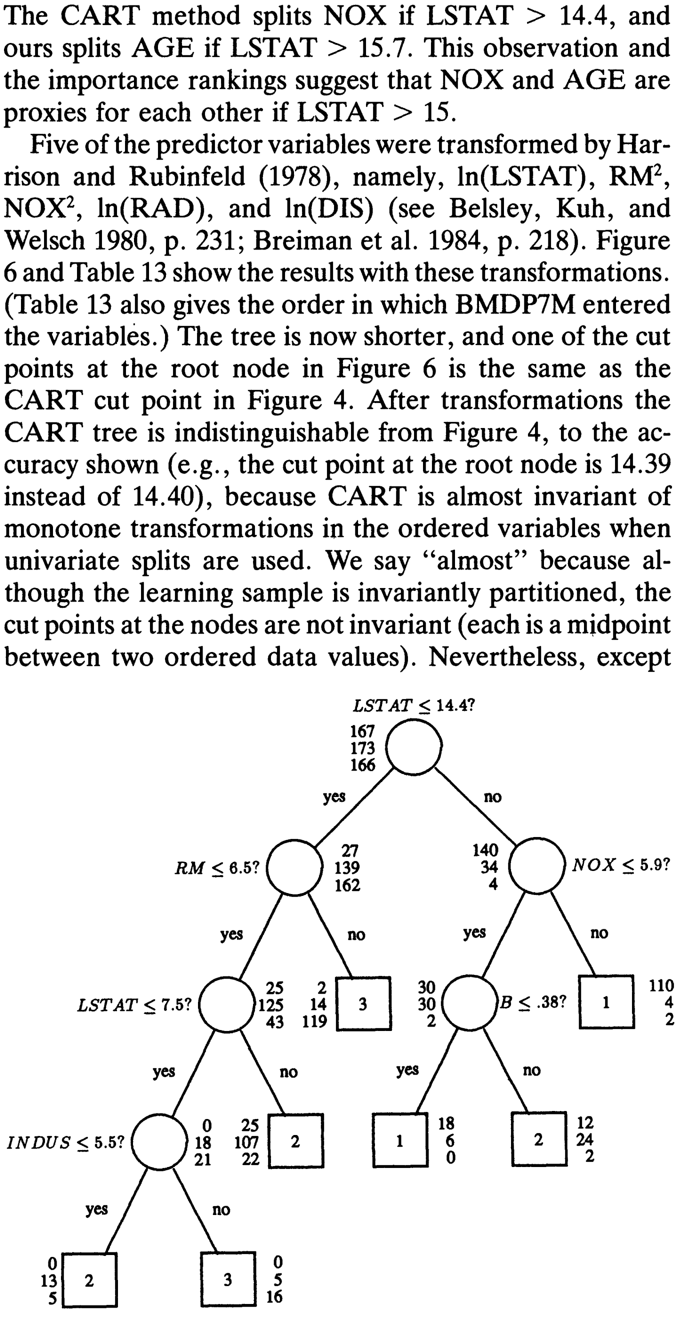

Figure 4. Boston Housing Tree From CART. This includes pruning by 10-fold cross-validation, estimated priors, and equal misclassification costs. The numbers in boxes are class assignments. Triples beside nodes are the node compositions; for example, there are 167 Class 1, 173 Class 2, and 166 Class 3 cases in the root node.

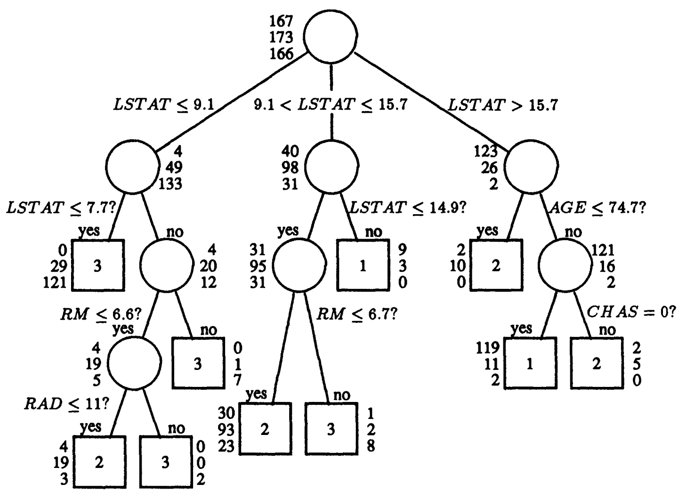

Figure 5. Boston Housing Tree With the Proposed Method. F_0 = 4. There are estimated priors and equal misclassification costs. The numbers in boxes are class assignments. Triples beside nodes are the node compositions; for example, there are 167 Class 1, 173 Class 2, and 166 Class 3 cases in the root node.

Table 14 shows how the learning sample is classified by the three trees. The apparent error rates are similar, with the CART tree being slightly lower. The CV error estimates are also similar, .24 ± .02, .26 ± .02, and .24 ± .02 for Figures 4, 5, and 6, respectively. The three trees gave identical predictions for 416 (82.2%) of the sample cases, of which 355 were correctly predicted.

Sensitivity to transformations has its advantages and disadvantages. When the goal is data exploration, a method sensitive to transformations obviously allows many more views of the data structure through its trees than one that is insensitive. Variability among the trees can help uncover masked variables as well as signal noisy data. The tradeoff is difficulty in deciding which tree to select. This may be done on the basis of smallest CV error estimate, which is then not unlike the use of CV in CART to select its "right-sized" tree, except that selection is manual rather than automatic and the trees are not necessarily nested. A method insensitive to monotone transformations may be preferred if the objective is to construct a tree-structured classifier but one does not want to worry about what transformations to use. This distinction between CART and the proposed method vanishes with linear combination splits.

Because our method is built from linear discriminant analysis, it may be a good idea to always include transformations (such as those of Hoaglin, Mosteller, and Tukey 1983) that improve the symmetry of the sample marginal distributions. The present example is one illustration.

Tree interpretation may be easier when the root node is split into as many subnodes as the number of classes. The node compositions in Figures 5 and 6 show that the first split typically produces three subtrees, each with one class dominant. Subsequent splits on a subtree merely try to separate minority cases. (Later nodes do not have three

Table 13. Variable Importance Ranking: Boston Housing Data

		FACT					
CART		Untransformed		Transformed		BMDP7M	
Variable	*Importance*	*Variable*	*Importance*	*Variable*	*Importance*	*Untransformed*	*Transformed*
LSTAT	100	LSTAT	100	ln(LSTAT)	100	LSTAT	ln(LSTAT)
RM	69	RM	44	RM^2	39	RM	RM^2
P/T	62	NOX	34	AGE	30	P/T	P/T
INDUS	61	AGE	32	NOX^2	29	NOX	*B*
AGE	58	TAX	31	INDUS	28	DIS	NOX^2
CRIM	55	INDUS	31	TAX	28	*B*	ln(DIS)
DIS	54	P/T	24	P/T	21	AGE	ZN
NOX	53	RAD	20	ln(DIS)	18	ZN	AGE
TAX	53	CRIM	18	CRIM	16		
RAD	36	*B*	17	*B*	15		
B	34	ZN	14	ln(RAD)	14		
ZN	12	DIS	14	ZN	11		
CHAS	5	CHAS	3	CHAS	1		

splits each, because the *F* ratios are weighted by the estimated node class priors and so can yield empty subnodes.) Thus to understand where most of the cases in a particular class go, the tree may be read top–down. For example, in Figure 6 the sample is split into three groups according to the value of LSTAT: (a) housing values are high (Class 3) in more affluent tracts (left branch of tree); (b) values in less affluent tracts (right branch) are either low (Class 1) or medium (Class 2), depending on age; and (c) tracts with average values of LSTAT (middle branch) are mostly white: Where this is the case, housing values are largely determined by the number of rooms. A bottom–up approach is required to interpret the CART tree in Figure 4.

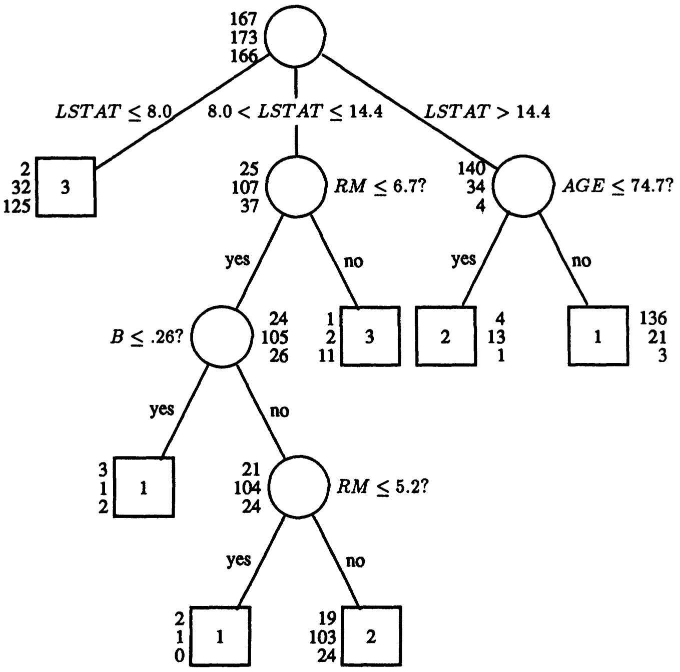

Figure 6. Boston Housing Tree With the Proposed Method Using Transformed Variables. F_0 = 4. There are estimated priors and equal misclassification costs. The numbers in boxes are class assignments. Triples beside nodes are the node compositions; for example, there are 167 Class 1, 173 Class 2, and 166 Class 3 cases in the root node.

The CART and FACT trees took 8.5 minutes and 30 seconds, respectively, to build.

8. CONCLUSION

In the examples the speed of the proposed method relative to CART ranges from a minimum of 7 (Table 7) to a maximum of 80 (Table 4), with a median of 16. As noted in point (d) of Section 1.3, these ratios are multiplied by a factor of 10 if the CV estimate of error is not required (because 10-fold CV was used). The test-sample error estimates show that neither method dominates on accuracy: CART wins 8 times and ours 17 times, with 3 ties. The better accuracy of one over the other is usually not significant, but when it is (Tables 2, 4, 6, and 11) there is an apparent pattern: CART is better with univariate splits, and ours is better with linear combination splits.

These observations are admittedly based on a handful of examples, and the usual words of caution about generalizations are in order. On the theoretical side, Breiman et al. (1984, theorem 12.19) proved that under mild regularity conditions, rules based on recursive partitioning are Bayes risk consistent. We have not yet identified situations in which these conditions hold for the methods proposed here. Like CART (Breiman et al., p. 327), no theoretical justifications are claimed for our splitting and stopping rules.

Tree-structured methods that allow univariate and nonunivariate splits are really two tools in one: a classification rule and data-exploratory technique. Nonunivariate splits

Table 14. Classifications of Learning Sample by the Three Trees

	CART			FACT 1			FACT 2		
Actual	*1*	*2*	*3*	*1*	*2*	*3*	*1*	*2*	*3*
1	128	37	2	128	38	1	141	23	3
2	10	144	19	144	127	32	23	116	34
3	2	29	135	2	26	138	5	25	136
Total	140	210	156	144	191	171	169	164	173
	81.3% correct			77.7% correct			77.7% correct		

NOTE: "FACT 1" and "FACT 2" refer to the trees in Figures 5 and 6, respectively. Column and row headings are the predicted and actual class labels, respectively.

would be the choice when the emphasis is on classification accuracy. The CART and our methods are then no different from non-tree-structured methods, because trees with such splits are typically difficult to comprehend. When the purpose is data exploration, however, a tree with univariate splits offers another way of data summary [called *data-base compression* by computer scientists; e.g., see Michie (1982, p. 223)] that supplements correlation matrices and two-dimensional scatterplots. In this case, computational efficiency and sensitivity to transformations may be desirable (e.g., compare ordinary regression vs. regression based on ranks).

Further details on the proposed method (including class probability trees and alternative terminal node assignment rules) are provided in Loh and Vanichsetakul (1986) and Vanichsetakul (1986). Wolberg, Tanner, Loh, and Vanichsetakul (1987) applied it to a medical diagnosis problem.

More information on the FACT computer program may be obtained from the first author.

[*Received January 1986. Revised July 1987.*]

REFERENCES

Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis* (2nd ed.), New York: John Wiley.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics,* New York: John Wiley.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees,* Belmont, CA: Wadsworth.

Dixon, W. J., Brown, M. B., Engelman, L., Frane, J. W., Hill, M. A., Jennrich, R. I., and Toporek, J. D. (1983), *BMDP Statistical Software,* Berkeley: University of California Press.

Fielding, A. (1977), "Binary Segmentation: The Automatic Interaction Detector and Related Techniques for Exploring Data Structure," in *The Analysis of Survey Data, Volume I: Exploring Data Structures,* eds. C. A. O'Muircheartaigh and C. Payne, New York: John Wiley, pp. 221–258.

Friedman, J. H. (1977), "A Recursive Partitioning Decision Rule for Nonparametric Classification," *IEEE Transactions on Computers,* 26, 404–408.

Gnanadesikan, R. (1977), *Methods for Statistical Data Analysis of Multivariate Observations,* New York: John Wiley.

Gordon, A. D. (1981), *Classification,* London: Chapman & Hall.

Greer, R. L. (1979), "Consistent Nonparametric Estimation of Best Linear Classification Rules/Solving Inconsistent Systems of Linear Inequalities," Technical Report 129, Stanford University, Dept. of Statistics.

Hand, D. J. (1981), *Discrimination and Classification,* New York: John Wiley.

Harrison, D., and Rubinfeld, D. L. (1978), "Hedonic Prices and the Demand for Clean Air," *Journal of Environmental Economics and Management,* 5, 81–102.

Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (1983), *Understanding Robust and Exploratory Data Analysis,* New York: John Wiley.

Levene, H. (1960), "Robust Tests for Equality of Variances," in *Contributions to Probability and Statistics,* ed. I. Olkin, Palo Alto, CA: Stanford University Press, pp. 278–292.

Loh, W.-Y., and Vanichsetakul, N. (1986), "Tree-Structured Classification Via Generalized Discriminant Analysis," Technical Report 781, University of Wisconsin, Dept. of Statistics.

Michie, D. (1982), "The State of the Art in Machine Learning, " *Introductory Readings in Expert Systems,* ed. D. Michie, New York: Gordon & Breach Science Publishers, pp. 208–229.

Mingers, J. (1986), "Inducing Rules for Expert Systems," *The Professional Statistician,* 5, 19–24.

Morgan, J. N., and Messenger, R. C. (1973), "THAID: A Sequential Analysis Program for the Analysis of Nominal Scale Dependent Variables," technical report, University of Michigan, Institute for Social Research.

Vanichsetakul, N. (1986), *Tree-Structured Classification Via Recursive Discriminant Analysis,* unpublished Ph.D. thesis, University of Wisconsin, Dept. of Statistics.

Vapnik, V. (1982), *Estimation of Dependencies Based on Empirical Data,* New York: Springer-Verlag.

Watson, G. S. (1983), *Statistics on Spheres,* New York: John Wiley.

Wolberg, W. H., Tanner, M. A., Loh, W.-Y., and Vanichsetakul, N. (1987), "Statistical Approach to Fine Needle Aspiration Diagnosis of Breast Masses," *Acta Cytologica,* 31, 737–741.

Comment

LEO BREIMAN and JEROME H. FRIEDMAN*

We regard the publication of this article in the *Journal of the American Statistical Association* as an important forward step, indicating an interest by the mainstream statistical community in the tree-structured approach to regression and classification. This certainly would not have been possible 10 years ago, when CART was originally developed. We also applaud the interest shown by Loh and Vanichsetakul in this methodology. We never regarded the CART approach as being the last word in recursive partitioning technology, which has evolved over the past 25 years beginning with the pioneering work of Morgan, Sonquist, Messenger and others at the University of Michigan. This evolution has taken place in the social sciences and the fields of electrical engineering, pattern recognition, and most recently Artificial Intelligence. We feel that the statistical community can contribute greatly to advances in this area.

From a purely technical point of view, however, we cannot regard the approach presented in this article as a step forward in this evolution. It uses a splitting rule derived by analogy to separation of normal distributions with equal covariance matrices, and F ratios to decide when to split and when to stop splitting. To this, special devices are added: If the F ratio based on separation of means is

* Leo Breiman is Professor, Department of Statistics, University of California, Berkeley, CA 94720. Jerome H. Friedman is Professor, Department of Statistics, and Group Leader, Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94305.

not large enough, then the predictor variables are transformed and the splits are based on dispersion. Polar coordinate splits are introduced to deal with possible radial symmetry; Levene's test of homogeneity of variances of the variables is performed to weed out noise variables. Categorical variables are handled by the introduction of 0–1 dummy variables, and missing data are handled by analogy to maximum likelihood, assuming normal densities.

The principal motivation given for this is computational. By sacrificing CART's thorough nonparametric approach, it is possible to greatly increase execution speed. This may be important in academic settings, where computational resources are often scarce. In industrial applications, however, the cost of collecting and organizing the data is usually much greater than even the most computationally intensive statistical-analysis procedures. In situations where one has an interest in the information to be learned from the data (as opposed to simply using it as a test bed for trying out procedures), scrimping on the analysis phase is being penny wise and pound foolish. We have seldom encountered industrial problems for which computing time for statistical data analysis was an issue. Issues such as accuracy and interpretability are much more important. Also, the less-thorough approach remains viable only to the extent that it is much faster than the thorough one. As algorithmic improvements are made to the latter, the motivation for the former diminishes. (A new, faster version of CART will soon be released.)

In addition to the issue of computational speed, this article offers many opinions concerning the efficacy of the approach presented. Unfortunately, we have difficulty finding any with which we can basically agree. Lack of space precludes a complete discussion of all of the issues, so we will only touch on a few. One general theme throughout the article is that splitting based on linear combinations of the variables is generally superior to univariate splits. Our experience has been to the contrary. Besides the obvious interpretability of models based on univariate splits (which is the biggest single advantage of the tree-structured approach) we have found that in most applications where recursive partitioning has higher accuracy than traditional methods, that advantage is achieved through univariate rather than linear combination splitting. Linear combination splitting was introduced in Friedman (1977). There, linear discriminant functions were applied recursively in a manner similar to that described in this article. This was later supplanted by the far more thorough algorithm implemented in CART. In both cases,·the principal reason for introducing linear combination splits was to make recursive partitioning competitive with classical linear procedures in settings appropriate for the latter. It does this to some extent, but in these situations one is usually better off with the appropriate classical procedure. Although linear combination splitting has strong intuitive appeal, it does not seem to achieve this promise in practice except for the settings mentioned previously. Most of the examples used for comparisons in this article, however, are taken from just these settings. The overwhelming majority of CART users prefer univariate splitting in practice even when computation is not an issue.

Another opinion seen throughout this article is that multiway splitting is generally superior to binary splitting. Proposals for recursive partitioning using multiway splits have appeared in the past (see Henrichon and Fu 1969). As Friedman (1977) argued, multiway splitting does not make as effective use of the conditional information potentially present in the tree as does binary splitting. This article also asserts that multiway splits are more easily interpreted. We do not think so, but interpretability is clearly in the eye of the beholder.

The two ways that the procedure presented in this article (FACT) gains speed over CART is by using an ad hoc top–down stopping rule, and by not implementing surrogate splits. Top–down stopping rules were used in all of the early predecessors of CART (AID, THAID, etc.). They were highly criticized (with good reason) on this point, and this was the principal reason for their lack of acceptance in the statistical community. The optimal complexity tree-pruning algorithm (based on cross-validatory choice) implemented in CART is probably the most important contribution of Breiman et al. (1984) to the evolution of tree-structured methodology. It tends to produce right-sized trees reliably.

In the course of the research that led to CART, almost two years were spent experimenting with different stopping rules. Each stopping rule was tested on hundreds of simulated data sets with different structures. Each new stopping rule failed on some data set. It was not until a very large tree was built and then pruned, using cross-validation to govern the degree of pruning, that we obtained something that worked consistently. The procedure presented in this article faces a similar predicament. It works well in four cases and does poorly on the fifth.

A large part of this article is concerned with trying to persuade the reader that one does not gain accuracy with CART through its thorough but computationally intensive nonparametric approach. This is done by comparing the two procedures on several examples. These mostly originate from Breiman et al. (1984). The simulated data examples used in Breiman et al. were deliberately chosen to have a simple and intuitive structure, on which almost any classification procedure would do fairly well. Unfortunately, this article uses these as serious test beds for accuracy comparisons, along with an example with normally distributed data, an example where there is some simple spherical symmetry, and an example with five categorical variables. The Boston housing data, which is essentially a 3–4 variable data set, is also used.

As Breiman et al. (1984) repeatedly emphasized, CART's most marked superiority to traditional methods is in the analysis of complex nonlinear data sets with many variables. The data sets used in the examples here are not of this type, but (with one exception) are fairly linear and amenable to classical linear procedures. The CART method has been applied to a variety of complex data bases in different fields (see Breiman et al. for some examples);

CART's performance on these contributes mainly to its attractiveness.

Still, to look at the examples actually used, the claim that in the univariate case both procedures have comparable accuracy is suspect. In four of the five examples the accuracy is comparable. In the spherical example CART has a test set accuracy of .09, compared with .19 for FACT. The authors attribute the better accuracy of CART here to the "build a large tree, then prune" approach. This is one of the few points on which we agree. It is true that the polar coordinate split does well in this particular case, because of its good match to the spherical symmetry built into the example. It is not likely that nonlinear decision boundaries will exhibit such nice symmetry in practice.

The article expresses several odd disparaging opinions concerning CART:

1. *Categorical Variables.* Because FACT (with its multiple splits) cannot handle categorical variables in a clean, elegant way, it is forced to use the usual linear regression trick of coding a C-value categorical variable into C-dummy 0–1 variables, and including the latter in linear combination splits. It is then commented, "The method proposed here can mix variable types" (p. 717). In the summary the authors give another unsubstantiated argument in favor of their recoding method.

2. *Transformations of Variables.* Because of CART's nonparametric approach, the models produced are invariant under all univariate monotone transformations of any or all of the predictor variables. There appears to be almost universal agreement that this is an attractive feature. The FACT method does not have this property. The article attempts to turn this into a virtue by stating "When the goal is data exploration, a method sensitive to transformations obviously allows many more views of the data structure through its trees than one that is insensitive" (p. 723). We leave it to the reader to sort out the logic in this statement.

3. *Cross-Validation.* The cross-validation used in CART is criticized on several grounds. For example, "the tree is randomized . . . because its size then depends on the random-number seed used to form the cross-validation samples" (p. 716). In this day of bootstrapping and data resampling, this criticism is a strange anachronism. Of course, the question is "how dependent?" Data are, by the very nature of the statistical enterprise, random. How much does changing the random-number seed in bootstrapping or cross-validation change the results as compared to the inherent randomness of the data? Our experience, based on many simulations, is that the relative effect is minor.

Another criticism is that "the dual-use of cross-validation . . . means that the CV estimate is not genuine" (p. 716). As pointed out in Breiman et al. (1984, p. 81), this was thoroughly investigated through many simulations, and it was concluded that this is not an issue.

4. Surrogate splits in CART are mainly used to handle missing data values. Based on the *simulations,* this article concludes that CART's missing-value algorithm is better. Later (in the summary) the authors attempt to explain why theirs *should* be superior.

It is important to keep in mind that for most problems there is a wide variety of sensible (and sometimes even not sensible) methods that have comparable accuracy. In these situations, methods can be evaluated by their interpretability and the insight they provide. On this basis it is difficult to beat simple nonparametric binary recursive partitioning represented by parsimonious trees. To the extent that CART has achieved popular appeal, it is due to this aspect rather than its increased accuracy in some situations. This tends to be especially true if one wants to present the results to nonclassically trained statisticians.

One does occasionally encounter classification problems for which the (optimal) decision boundaries are highly nonlinear (or nonquadratic). In these cases recursive partitioning has the potential to achieve substantially higher accuracy than classical approaches. It is, however, in just these situations that CART's thorough (time-consuming) nonparametric approach (namely, investigating all potential splits and bottom–up optimal complexity tree pruning) is essential. (The spherical example in this article provides a simple illustration of this.) Of course, one seldom knows in advance the precise nature of the underlying class boundaries. One must therefore choose whether to use a computationally fast procedure that has difficulty detecting and dealing with these (perhaps infrequent) occurrences, or to apply a computationally intensive but thorough method knowing that it will often provide no better performance than the faster methods, because of the underlying simplicity of the problem.

What one pays for with the computing time spent for computationally intensive nonparametric procedures (such as CART, bootstrapping, etc.) is insurance for those situations where one's expectations (assumptions) are violated. An emerging trend in modern statistics has been methods that substitute computation for unverifiable assumptions. This is motivated by the fact that the cost for computation is decreasing roughly by a factor of two every year, whereas the price paid for incorrect assumptions is remaining the same.

We thank the editors of JASA for inviting us to make these comments. We hope this publication, by bringing recursive partitioning methodology to the attention of statisticians, will stimulate research and development on what we consider an important area of data analysis.

ADDITIONAL REFERENCE

Henrichon, E. G., Jr., and Fu, K. S. (1969), "A Nonparametric Partitioning Procedure for Pattern Classification," *IEEE Transactions on Computers,* 18, 614–624.

Rejoinder

WEI–YIN LOH and NUNTA VANICHSETAKUL

A major difficulty with the evaluation of any tree-structured method is that there are at least three different criteria that can be used, namely, classification accuracy, computational speed, and interpretability of the trees. When we first thought of the basic idea for our method, we were sure it would run fast, but we did not know if it would be accurate. The results reported in the article, as well as others we have obtained since then, indicate that its accuracy is much better than we had hoped. Of course, it is possible that we are entirely misled by these examples, but we doubt it. We admit that we do not have as extensive experience with our method as Breiman and Friedman have with CART, although the FACT program will be almost three years old by the time our article is published.

The discussants may be correct in saying that in industrial settings computational efficiency is not a prime consideration, but surely this must depend on the size of the data set and the number of times a method is applied.

Interpretability of a tree is much harder to define, because it must ultimately depend on the consumer. What one person finds interpretable may make no sense to another. A tree is only interpretable if the consumer knows what each variable measures, on its own as well as relative to the other variables. Perhaps the most that a computer program can be expected to do, as far as this issue is concerned, is to allow univariate splits.

Another factor that makes it almost impossible to say whether one method produces more interpretable trees than another is that there are often many trees that seem to describe a data set equally well—compare the CART and FACT trees for the housing and digit problems in our article, for example. After all, a tree is only a sequential way of summarily describing a data set. Just as there are many ways to describe a picture in a sequence of steps, there ought to be different but equally correct ways to summarize data with a tree structure.

The lack of a definition of correctness is not a handicap. On the contrary, the consumer can often learn more about data from a set of different trees, all of which make sense, than from a single one. And that is where we feel that a method that is noninvariant to transformations can be advantageous.

We now address some of the discussants' more specific comments.

1. It is not true that our approach cannot be made invariant of monotone transformations in the individual variables. Just as in all rank-based methods, one can simply recode the data into ranks, run FACT on the ranked observations, and then retransform back to the original units (using linear interpolation between ranks if necessary). The result would be no less invariant than CART's.
2. Tree selection by cross-validation pruning or some other method is almost mandatory for CART, because of its splitting approach. A similar kind of selection is possible with FACT by first getting a set of FACT trees (each possibly using a different set of variable transforms), along with their associated cross-validation estimates of error. The final tree can then be selected on the basis of its error estimate *and* its interpretability. Given the speed superiority of our method over CART, the increased computer time required could still be less than that for one CART run.
3. We believe that splitting based on linear combinations of the variables should generally yield better classification accuracy than univariate splits. The reason the discussants find this to be false in the case of CART could be either (a) because its search algorithm is getting trapped in local maxima (see Breiman et al. 1984, p. 132, last sentence), or (b) because using univariate surrogate splits to predict a linear combination split when there are missing values is poor strategy (see Sec. 6, item 3 of our article).
4. The discussants argue that the way cross-validation is used in CART is similar to bootstrapping in general. We feel there are two big differences. First, bootstrappers typically use hundreds or thousands of bootstrap samples, compared to the 10 or 25 cross-validation samples typical in CART. Second, the major use of the bootstrap has been in estimating the error of a solution, rather than in constructing the solution itself, which is why CART uses cross-validation. A randomized estimate of error seems more acceptable than a decision tree with a random number of nodes.

We thank all concerned for the opportunity to add these comments. An improved version of FACT, with many additional features, should be ready for distribution by the time this gets into print.