DISTRIBUTION BASED TREES ARE MORE ACCURATE

Nong Shang
School of Public Health
University of California
shang@stat.berkeley.edu

Leo Breiman
Statistics Department
University of California
leo@stat.berkeley.edu

ABSTRACT

Classification trees are attractive in that they present a simple and easily understandable structure. But on many data sets their accuracy is far from optimal. Much of this lack of accuracy is due to their instability--small changes in the data can lead to large changes in the resulting tree. This instability is the reason that combining many trees by voting can lead to dramatic decreases in test set error (Breiman[1995]). But combining trees loses the simple structure. To keep the simple structure and improve accuracy, a way must be found to reduce the instability in the construction. If we knew the true probability distribution of the inputs and outputs, then the splits in the tree could be based on this distribution and give more accuracy then the splits based on a finite data set. So we turn the tree procedure around--instead of basing the splits on the data, the data is used to estimate the input-output probability distribution and the splits are then based on this estimate. We give the details of this construction. The experimental results on a number of well-know data sets indicate that this procedure has potential for producing much more accurate trees.

1. Introduction

Classification and regression trees ala CART and C4.5 are sensitive to small changes in the data. A small change may result in a substantially altered tree. Because of this, tree predictors have high variance (Breiman [1995], [1996]) which results in inflated test set error rates. Tree construction proceeds by recursively splitting the data. Thus the sample size available for determining the splits decreases rapidly with the depth of the tree. The result is increasingly noisy and poorly determined splits.

Denote the output, numerical or a class label, by Y and the multivariate inputs by **X.** If we knew the joint probability distribution of Y,**X** then we could determine the optimal splits at each node using this distribution. We would have, essentially, infinite sample size to use in tree construction. This suggests the following possibility--

*Instead of using the given data set to determine the splits, use the data set to estimate the Y,**X** distribution and then use this estimate to determine the splits.*

This idea works and gives some dramatic decreases in test set error as compared with the standard method of tree construction. In this paper, we focus on classification trees, but the same methods (actually simplified) hold for regression. In classification, we construct a separate density estimate for each class using a kernel density estimate of the Y,**X** distribution. Then optimal splits are found in each node by using the estimated distributions to compute the Gini criterion for a grid of splits points on each variable. We will refer to this procedure as DB-CART where DB stands for distribution based.

Note that because we are using the estimated probability and not the data points to split with, that when a parent is split, the influence of a single data point may be divided between the

two children nodes. Thus, the splits in the lower nodes may be influenced by probability contributions from many data points.

Tree construction, consisting of the determination of node splits and node class probabilities is based on the estimated probabilities. The rest of the procedure is standard. Using the estimates a large tree is grown and a sequence of pruned subtrees determined. To select one of these subtrees, either a test set or 10-fold cross-validation is used. If a large enough test set is available, it is run down the sequence of pruned trees and the tree with lowest test set error selected. If not, 10% of the data is deleted, a tree grown on the other 90% using estimated probabilities, and the 10% run down it to give test set error estimates. Then a different 10% is let out, and so on.

The layout for this paper is as follows: Section 2 has experimental results concerning error rates of DB-CART compared to CART on a number of well-known data sets. In Section 3 we review how splits and node class probabilities would be determined if we knew the Y,**X** distribution. Section 4 gives the details of the probability estimates and Section 5 has comments.

2. **Experimental Results for DB-CART Compared to CART**

DB-CART was compared to CART on some well-known data sets summarized in table 1.

Table 1 Summary of Data Sets

Data Set	#Classes	#Predictors	#training	#test
Waveform	3	21	300	3000
Vowel	11	10	990	
Ionosphere	2	33	351	
Sonar	2	60	208	
Diabetes	2	8	786	
Glass	6	9	214	
Breast Cancer	2	30	569	

All of these data sets are in the UCI repository and are documented there. Waveform is artificial data. Test set errors were computed as follows: for the waveform data a 300 case training set and 3000 case test set were independently generated 10 times. The test set error rate was computed as the average over the 10 runs. In the other data sets the data was divided at random in 90%-10%. The tree was grown and pruned on the 90%, and the 10% then used to measure test set error. This was repeated 50 times and the test set errors averaged. The results are given in Table 2.

Table 2 Test Set Error(%) for CART and DB-CART

Data Set	CART Error	DB-CART Error	Decrease
Waveform	28.4	24.7	13%
Vowel	21.8	10.0	54%
Ionosphere	11.1	8.7	22%
Sonar	32.1	18.2	43%
Diabetes	26.3	25.6	3%
Glass	28.6	29.4	-3%
Breast Cancer	6.5	3.8	42%

Using the estimated distribution to construct trees can produce surprisingly large decreases in error rates. But in two of the data set, glass and diabetes, there is no significant difference. We are studying these two to try and understand the problem. Because using the estimated distribution emulates a larger sample size, the trees grown by DB-CART are generally larger than those grown by CART. The comparison of sizes is in Table 5.

Table 5 Number of Terminal Nodes in Trees Constructed by CART and DB-CART

Data Set	CART	DB-CART
Waveform*	15	49
Vowel	107	244
Ionosphere	6	16
Sonar	6	41
Diabetes	11	33
Glass	11	26
Breast Cancer	7	40

3. **Using the Distribution to Construct the Tree**

Suppose we know the joint distribution of the random vector Y,**X** where Y is the class label and **X** is a vector of M predictor variables, i.e. **X** = (X_1, ... ,X_M). That is, suppose we know P(Y=j, **X**$\in$ **dx**). Let P(j)=P(Y=j). A node **t** of the tree corresponds to **X**$\in$ I where I is a multidimensional rectangle.

Suppose we want to split I by a split of the form $X_m \le c$. Let

$$P_L = P(X_m \le c | \mathbf{X} \in \mathbf{I}) \qquad P_R = P(X_m > c | \mathbf{X} \in \mathbf{I})$$
$$q_{jL} = P(j | X_m \le c, \mathbf{X} \in \mathbf{I}) \qquad q_{jR} = P(j | X_m > c, \mathbf{X} \in \mathbf{I})$$

The optimal split is defined to be the one minimizing the Gini criterion (see Breiman et. al [1984]). This is equivalent to maximizing

$$G = P_L \sum_j q_{jL}^2 + P_R \sum_j q_{jR}^2 \tag{3.1}$$

Look at the first term

$$P(X_m \le c | \mathbf{X} \in \mathbf{I}) \sum_j P^2(j | X_m \le c, \mathbf{X} \in \mathbf{I})$$

This equals

$$[\sum_j P^2(X_m \le c, \mathbf{X} \in \mathbf{I} | j) P^2(j)] / P(\mathbf{I}) P(X_m \le c, \mathbf{X} \in \mathbf{I}) \tag{3.2}$$

Note that

$$P(X_m \le c, \mathbf{X} \in \mathbf{I}) = \sum_j P(X_m \le c, \mathbf{X} \in \mathbf{I} | j) P(j)$$

The second term in (2.1) has an expression similar to (2.2) but uses $X_m > c$. A search is made over all m,c to find the values that maximize (21.). Call these m*,c*. Now the new left and right nodes become

$$\mathbf{t}_L = \mathbf{I} \amalg \{X_{m^*} \le c^*\} \qquad \mathbf{t}_R = \mathbf{I} \amalg \{X_{m^*} > c^*\}$$

and the process is iterated.

Using the probability distribution to construct the tree is equivalent to having an infinite number of samples available from the Y,**X** distribution. By known consistency proofs (see Breiman et.al [1984]), trees constructed this way converge toward the minimal obtainable error rate as they are grown larger.

4. **Estimating the Distribution.**

But given a finite training data set T, the distribution isn't known. However, it can be estimated. Let $T = \{(j_n, \mathbf{x}_n), n = 1, \ldots, N\}$, L_j the set of all indices n such that $j_n = j$, and N_j the number of instances in class j. We will estimate the density of $P(\mathbf{dx} \mid j)$ by a kernel density estimate:

$$\hat{f}_j(\mathbf{x}) = \frac{1}{N_j} \sum_{n \in Lj} K(\mathbf{x} - \mathbf{x}_n)$$

4.1 Numerical Variables

To begin with, normalize all numerical variables to have range 0 to 1. Let f(x) be a Gaussian density with mean zero and standard deviation h > 0, and define

$$K(\mathbf{x}) = \prod_m f(x_m) \tag{4.1}$$

where the product is over all numerical variables. Thus, the cumulative distribution function $F_j(\mathbf{x})$ of the estimated density is

$$F_j(\mathbf{x}) = \frac{1}{N_j} \sum_{n \in Lj} \prod_m F((x_m - x_{m,n}) / h)$$

where F is the standard N(0,1) cumulative distribution function. For any interval I = [a,b], let F(I)=F(a)-F(b). If $\mathbf{I} = I_1 \otimes I_2 \otimes \ldots \otimes I_M$ then

$$P_j(\mathbf{I}) = \frac{1}{N_j} \sum_{n \in Lj} \prod_m F((I_m - x_{m,n}) / h) \tag{4.2}$$

Define

$$q(n, \mathbf{I}) = \prod_m F((I_m - x_{m,n}) / h)$$

Now using these estimated probabilities, we want to evaluate the Gini (3.1) for splits of the form $\{X_m \leq c\}$ Denote by I(m,c) the interval $\{x_m \leq c\}$, and let $R_n(m,c)$ be the ratio

$$R_n(m,c) = F((I_m \cap I(m,c) - x_{m,n}) / h)) / F((I_m - x_{m,n}) / h)$$

Then

$$P_j(\mathbf{I} \cap I(m,c)) = \frac{1}{N_j} \sum_{n \in Lj} q(n,\mathbf{I}) R_n(m,c) \qquad (4.3)$$

Putting P(j)=Nj/N and using the (4.3) estimate in (3.2) gives the first term in the Gini. The second term is gotten by using the complement of I(m,c) in the expression for $R_n(m,c)$.

3.2 Non-numeric Variables

Suppose x takes on non-numerical values we label as 1,2, ..., I. The density $f(x,x_n)$ that appears in the product (4.1) is then a probability distribution on these values. This distribution is defined in the following way: let the proportion of categories 1,2, ... ,I for instances in class j be given by p(i,j). Suppose x_n is in class j and $x_n = i0$. Then , for $0<q<1$, set

$$f(x, x_n) = \begin{cases} (1-q) + q \cdot p(i0, j), & if \ x = i0 \\ q \cdot p(i, j) \qquad\qquad , & if \ x = i \neq i0 \end{cases}$$

4.3 Maximization search

If the variable x has many distinct values, then the search over splits of the form $\{x \leq c\}$ is confined to a grid of 100 equally spaced c values. If x has only a few distinct values, then the search is confined to splits for which c is at the midpoint between two x-values.

4.4 Setting parameter values.

For numerical variables, the value of h needs to be selected, and for non-numerical, the value of q. This is done by cross-validation. Our current procedure is to search over a small grid of h and q values values to find those giving the lowest cross-validation error rates. This is expensive computationally since for each grid point examined, a tree is grown and its error rate estimated using 10-fold cross-validation. One current research priority is to find a faster method for locating optimal parameter values.

5. **Comments**

Given the training data $T = \{(j_n, \mathbf{x}_n), n = 1, \ldots, N\}$, if we estimate the $Y, \mathbf{X}$ distribution by the empirical distribution that puts weight 1/N at each of the points $(j_n, \mathbf{x}_n)$, then using this estimate gets us back to CART. If an oversmoothed estimate is used, then it's too far away from the "true" distribution and the accuracy drops. Density estimates in high dimensional spaces are known to be quite noisy. In particular, kernel density estimates do not perform well. But the evidence seems to be that the accuracy of CART is significantly enhanced by using an estimated distribution.

We believe that the reason is that the relevant estimates needed are low dimensional. CART splits are univariate. So what is needed are fairly good univariate density estimates together with rough estimates of interactions. Kernel density estimation does this well enough to produce increased test set accuracy. We are encouraged by these preliminary results and will explore further along this direction. Our distribution estimate is admittedly a first approximation and we plan to see if it can be improved. Its possible, for instance, that variable kernel estimates will provide better accuracy. But the results to date are encouraging in that they shown that accuracy can be improved without loss in simplicity and interpretability.

References

Breiman, L., Friedman, J.,Olshen, R., and Stone, C. [1984] Classification and Regression Trees, Wadsworth

Breiman, L.[1995] Bagging Predictors, in press, Machine Learning

Breiman, L.[1996] Bias, Variance, and Arcing Classifiers, submitted to Annals of Statistics, ftp ftp.stat.berkeley.edu pub/breiman/arcall.ps