CLASSIFICATION AND REGRESSION TREES

Leo Breiman
University of California, Berkeley

Jerome H. Friedman
Stanford University

Richard A. Olshen
Stanford University

Charles J. Stone
University of California, Berkeley

CHAPMAN & HALL/CRC

Boca Raton London New York Washington, D.C.

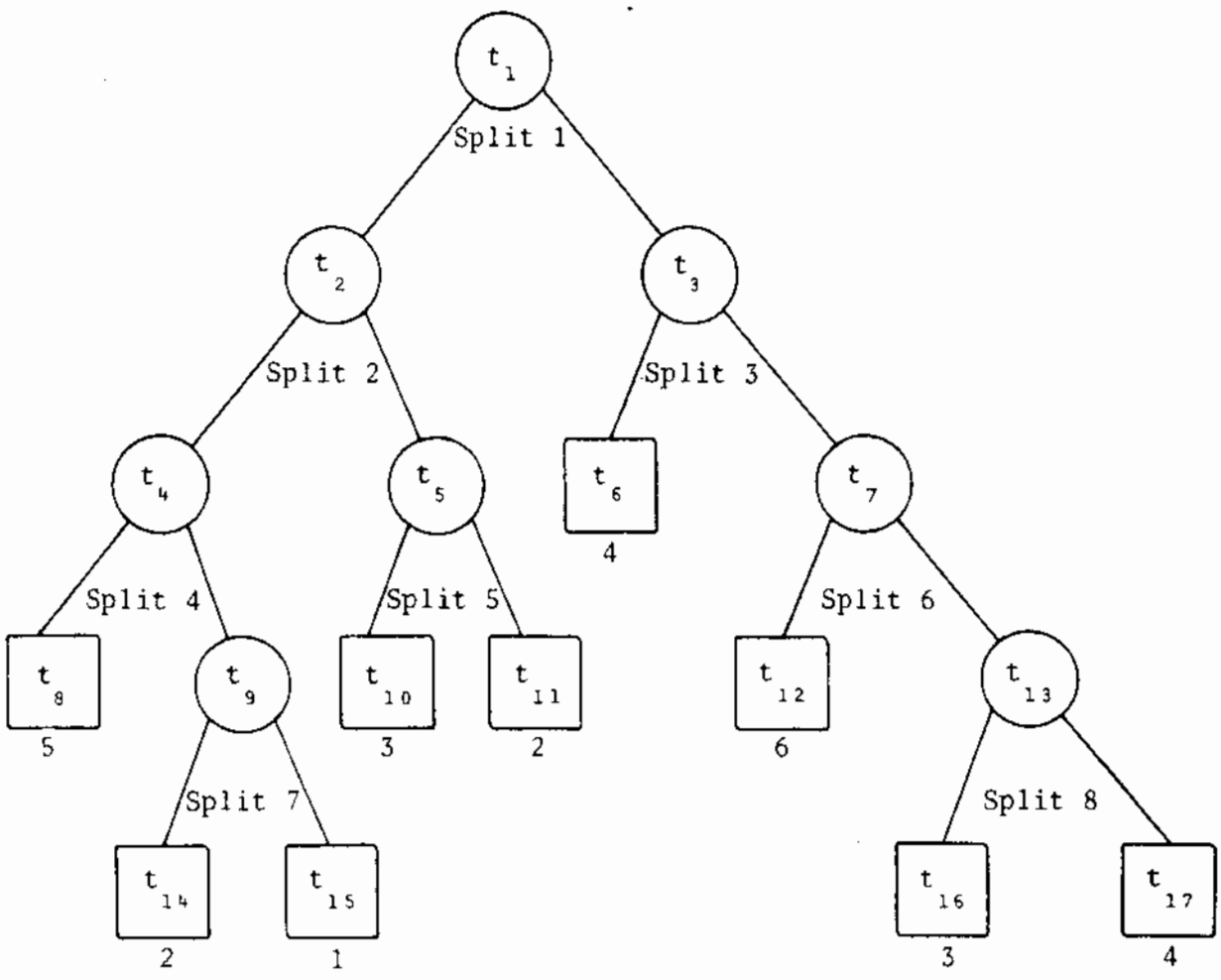


FIGURE 2.4

The crux of the problem is how to use the data $\mathcal{L}$ to determine the splits, the terminal nodes, and their assignments. It turns out that the class assignment problem is simple. The whole story is in finding good splits and in knowing when to stop splitting.

2.3 CONSTRUCTION OF THE TREE CLASSIFIER

The first problem in tree construction is how to use $\mathcal{L}$ to determine the binary splits of X into smaller and smaller pieces. The fundamental idea is to select each split of a subset so that the data in each of the descendant subsets are "purer" than the data in the parent subset.

For instance, in the six-class ship problem, denote by p_1, ..., p_6 the proportions of class 1, ..., 6 profiles in any node. For the root node t_1, $(p_1, \ldots, p_6) = \left(\frac{1}{6}, \frac{1}{6}, \ldots, \frac{1}{6}\right)$. A good spli of t_1 would be one that separates the profiles in $\mathcal{L}$ so that all profiles in classes 1, 2, 3 go to the left node and the profiles i 4, 5, 6 go to the right node (Figure 2.5).

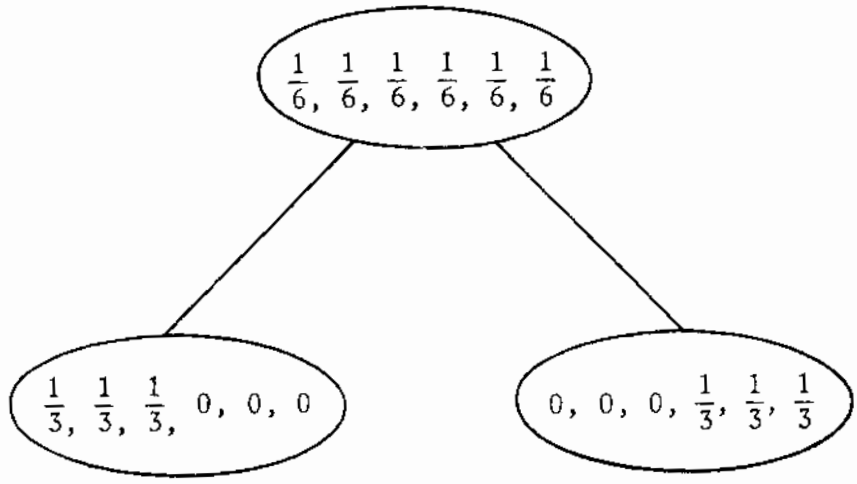


FIGURE 2.5

Once a good split of t_1 is found, then a search is made for good splits of each of the two descendant nodes t_2, t_3.

This idea of finding splits of nodes so as to give "purer" descendant nodes was implemented in this way:

1. Define the node proportions $p(j|t)$, $j = 1, \ldots, 6$, to be the proportion of the cases $\mathbf{x}_n \in t$ belonging to class j, so that

   $$p(1|t) + \cdots + p(6|t) = 1.$$

2. Define a measure $i(t)$ of the impurity of t as a nonnegative function ϕ of the $p(1|t), \ldots, p(6|t)$ such that

   $$\phi\left(\frac{1}{6}, \frac{1}{6}, \ldots, \frac{1}{6}\right) = \text{maximum},$$

   $$\phi(1, 0, 0, 0, 0, 0) = 0, \ \phi(0, 1, 0, 0, 0, 0) = 0, \ \ldots,$$

   $$\phi(0, 0, 0, 0, 0, 1) = 0$$

That is, the node impurity is largest when all classes are equally mixed together in it, and smallest when the node contains only one class.

For any node t, suppose that there is a candidate split s of the node which divides it into t_L and t_R such that a proportion p_L of the cases in t go into t_L and a proportion p_R go into t_R (Figure 2.6).

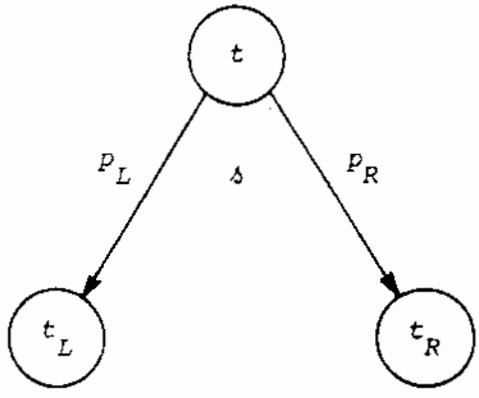


FIGURE 2.6

Then the goodness of the split is defined to be the decrease in impurity

$$\Delta i(s, t) = i(t) - p_L i(t_L) - p_R i(t_R).$$

The final step is:

3. Define a candidate set S of binary splits s at each node. Generally, it is simpler to conceive of the set S of splits as being generated by a set of questions Q, where each question in Q is of the form

   Is $\mathbf{x} \in A$?, $A \subset X$.

   Then the associated split s sends all $\mathbf{x}_n$ in t that answer "yes" to t_L and all $\mathbf{x}_n$ in t that answer "no" to t_R.

   In the ship project the node impurity was defined as

$$i(t) = -\sum_{1}^{6} p(j|t) \log p(j|t).$$

There is no convincing justification for this specific form of $i(t)$. It was selected simply because it was a familiar function having

4.3 THE MULTICLASS PROBLEM: UNIT COSTS

Two different criteria have been adopted for use in the multiclass problem with unit costs. These come from two different approaches toward the generalization of the two-class criterion and are called the

Gini criterion
Twoing criterion

4.3.1 The Gini Criterion

The concept of a criterion depending on a node impurity measure has already been introduced. Given a node t with estimated class probabilities $p(j|t)$, $j = 1, \ldots, J$, a measure of node impurity given t

$$i(t) = \phi(p(1|t), \ldots, p(J|t))$$

is defined and a search made for the split that most reduces node, or equivalently tree, impurity. As remarked earlier, the original function selected was

$$\phi(p_1, \ldots, p_J) = -\sum_j p_j \log p_j.$$

In later work the Gini diversity index was adopted. This has the form

$$i(t) = \sum_{j \neq i} p(j|t)p(i|t) \qquad (4.8)$$

and can also be written as

$$i(t) = \Big(\sum_j p(j|t)\Big)^2 - \sum_j p^2(j|t) = 1 - \sum_j p^2(j|t). \qquad (4.9)$$

In the two-class problem, the index reduces to

$$i(t) = 2p(1|t)p(2|t),$$

equivalent to the two-class criterion selected previously.

The Gini index has an interesting interpretation. Instead of using the plurality rule to classify objects in a node t, use the

rule that assigns an object selected at random from the node to class i with probability $p(i|t)$. The estimated probability that the item is actually in class j is $p(j|t)$. Therefore, the estimated probability of misclassification under this rule is the Gini index

$$\sum_{j \neq i} p(i|t)p(j|t).$$

Another interpretation is in terms of variances (see Light and Margolin, 1971). In a node t, assign all class j objects the value 1, and all other objects the value 0. Then the sample variance of these values is $p(j|t)(1 - p(j|t))$. If this is repeated for all J classes and the variances summed, the result is

$$\sum_j p(j|t)(1 - p(j|t)) = 1 - \sum_j p^2(j|t).$$

Finally, note that the Gini index considered as a function $\phi(p_1, \ldots, p_J)$ of the $p_1, \ldots, p_J$ is a quadratic polynomial with nonnegative coefficients. Hence, it is concave in the sense that for $r + s = 1$, $r \geq 0$, $s \geq 0$,

$$\phi(rp_1 + sp_1', rp_2 + sp_2', \ldots, rp_J + sp_J') \geq r\phi(p_1, \ldots, p_J) + s\phi(p_1', \ldots, p_J').$$

This ensures (see the appendix) that for any split s,

$$\Delta i(s, t) \geq 0.$$

Actually, it is strictly concave, so that $\Delta i(s, t) = 0$ only if $p(j|t_R) = p(j|t_L) = p(j|t)$, $j = 1, \ldots, J$.

The Gini index is simple and quickly computed. It can also incorporate symmetric variable misclassification costs in a natural way (see Section 4.4.2).

4.3.2 The Twoing Criterion

The second approach to the multiclass problem adopts a different strategy. Denote the class of classes by C, i.e.,

$$C = \{1, \ldots, J\}.$$

At each node, separate the classes into two superclasses,

$$C_1 = \{j_1, \ldots, j_n\},\ C_2 = C - C_1.$$

Call all objects whose class is in C_1 class 1 objects, and put all objects in C_2 into class 2.

For any given split s of the node, compute $\Delta i(s, t)$ *as though it were a two-class problem.* Actually $\Delta i(s, t)$ depends on the selection of C_1, so the notation

$$\Delta i(s, t, C_1)$$

is used. Now find the split $s^*(C_1)$ which maximizes $\Delta i(s, t, C_1)$. Then, finally, find the superclass C_1^* which maximizes

$$\Delta i(s^*(C_1), t, C_1).$$

The split used on the node is $s^*(C_1^*)$.

The idea is then, at every node, to select that conglomeration of classes into two superclasses so that considered as a two-class problem, the greatest decrease in node impurity is realized.

This approach to the problem has one significant advantage: *It gives "strategic" splits* and informs the user of class similarities. At each node, it sorts the classes into those two groups which in some sense are most dissimilar and outputs to the user the optimal grouping C_1^*, C_2^* as well as the best split s^*.

The word *strategic* is used in the sense that near the top of the tree, this criterion attempts to group together large numbers of classes that are similar in some characteristic. Near the bottom of the tree it attempts to isolate single classes. To illustrate, suppose that in a four-class problem, originally classes 1 and 2 were grouped together and split off from classes 3 and 4, resulting in a node with membership

Class:	1	2	3	4
No. cases:	50	50	3	1

Then on the next split of this node, the largest potential for decrease in impurity would be in separating class 1 from class 2.

Spoken word recognition is an example of a problem in which twoing might function effectively. Given, say, 100 words (classes), the first split might separate monosyllabic words from multisyllabic words. Future splits might isolate those word groups having other characteristics in common.

As a more concrete example, Figure 4.5 shows the first few splits in the digit recognition example. The 10 numbers within

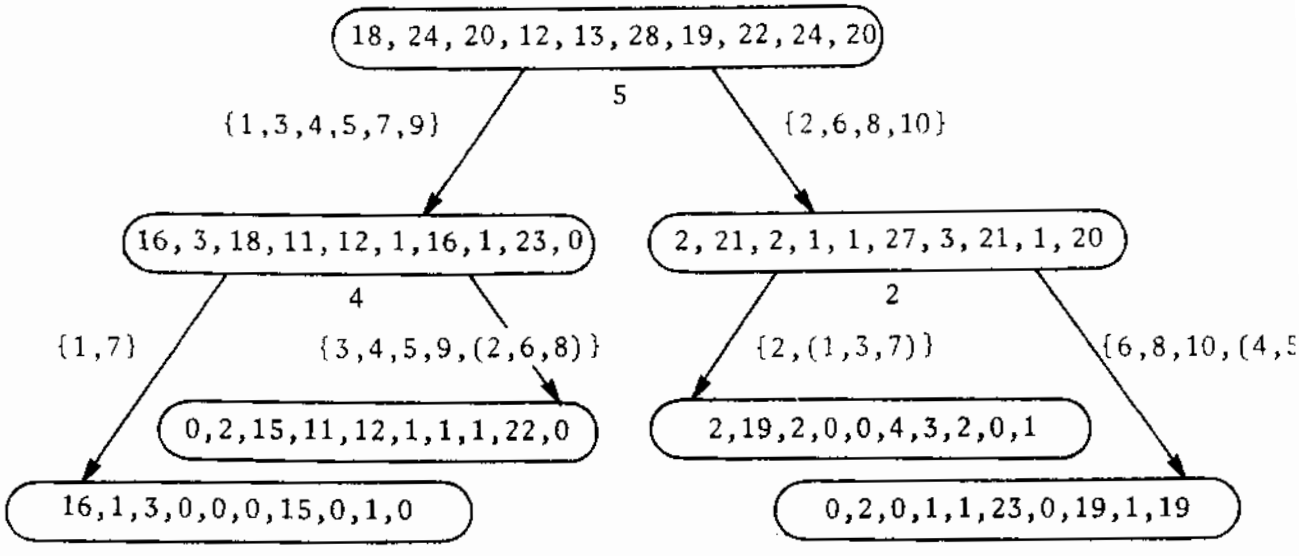
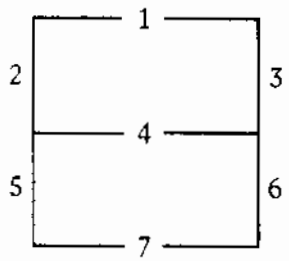


FIGURE 4.5

each node are the class memberships in the node. In each split the numbers in brackets by the split arrows are the superclasses C_1^*, C_2^*, for the split. In parentheses in the brackets are the classes whose populations are already so small in the parent node that their effect in the split is negligible. Zero populations have been ignored.

Recall that the lights are numbered as



The first split, on the fifth light, groups together classes 1, 3, 4, 5, 7, 9 and 2, 6, 8, 10. Clearly, the fifth light should be off for 1, 3, 4, 5, 7, 9 and on for the remaining digits. The next split on the left is on light 4 and separates classes 1, 7 from classes 3, 4, 5, 9. On the right, the split on light 2 separates class 2 from 6, 8, 10.

Although twoing seems most desirable with a large number of classes, it is in such situations that it has an apparent disadvantage in computational efficiency. For example, with J classes, there are 2^{J-1} distinct divisions of C into two superclasses. For $J = 10$, $2^{J-1} \simeq 1000$. However, the following result shows, rather surprisingly, that twoing can be reduced to an overall criterion, running at about the same efficiency as the Gini criterion.

THEOREM 4.10. *Under the two-class criterion* $p(1|t)p(2|t)$, *for a given split* s, *a superclass* $C_1(s)$ *that maximizes*

$$\Delta i(s, t, C_1)$$

is

$$C_1(s) = \{j: p(j|t_L) \geq p(j|t_R)\}$$

and

$$\max_{C_1} \Delta i(s, t, C_1) = \frac{p_L p_R}{4}\left[\sum_j |p(j|t_L) - p(j|t_R)|\right]^2 .$$

COROLLARY 4.11. *For any node* t *and split* s *of* t *into* t_L *and* t_R, *define the twoing criterion function* $\Phi(s, t)$ *by*