**Hunt’s Algorithm**

In Hunt’s algorithm, a decision tree is grown in a recursive fashion by partitioning the training records into successively purer subsets. Let Dt be the set of training records that are associated with node t and y = {y1, y2, . . . , yc} be the class labels. The following is a recursive definition of Hunt’s algorithm.

Step 1: If all the records in Data belong to the same class yt, then t is a leaf node labeled as yt.

Step 2: If Data contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome of the test condition and the records in Dt are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node.



**Figure 5.4.** Training set for predicting borrowers who will default on loan payments.

To illustrate how the algorithm works, consider the problem of predicting whether a loan applicant will repay her loan obligations or become delinquent, subsequently defaulting on her loan. A training set for this problem can be constructed by examining the records of previous borrowers. In the example shown in Figure 5.4, each record contains the personal information of a borrower

along with a class label indicating whether the borrower has defaulted on loan payments.

The initial tree for the classification problem contains a single node with class label Defaulted =No (see Figure 5.5(a)), which means that most of the borrowers successfully repaid their loans. The tree, however, needs to be redefined since the root node contains records from both classes. The records are subsequently divided into smaller subsets based on the outcomes of the Home Owner test condition, as shown in Figure 5.5(b). The justification for choosing this attribute test condition will be discussed later. For now, we will assume that this is the best criterion for splitting the data at this point. Hunt’s algorithm is then applied recursively to each child of the root node. From the training set given in Figure 5.4, notice that all borrowers who are home owners successfully repaid their loans. The left child of the root is therefore a leaf node labelled Defaulted = No (see Figure 5.5(b)). For the right child, we need to continue applying the recursive step of Hunt’s algorithm until all the records belong to the same class. The trees resulting from each recursive step are shown in Figures 5.5(c) and (d).



**Figure** 5 **.**5Hunt’ s algorithm for inducing decision trees.

Hunt’s algorithm will work if every combination of attribute values is present in the training data and each combination has a unique class label. These assumptions are too stringent for use in most practical situations. Additional conditions are needed to handle the following cases:

1. It is possible for some of the child nodes created in Step 2 to be empty; i.e., there are no records associated with these nodes. This can happen if none of the training records have the combination of attribute values associated with such nodes. In this case the node is declared a leaf node with the same class label as the majority class of training records associated with its parent node.

2. In Step 2, if all the records associated with Dt have identical attribute values (except for the class label), then it is not possible to split these records any further. In this case, the node is declared a leaf node with the same class label as the majority class of training records associated with this node.