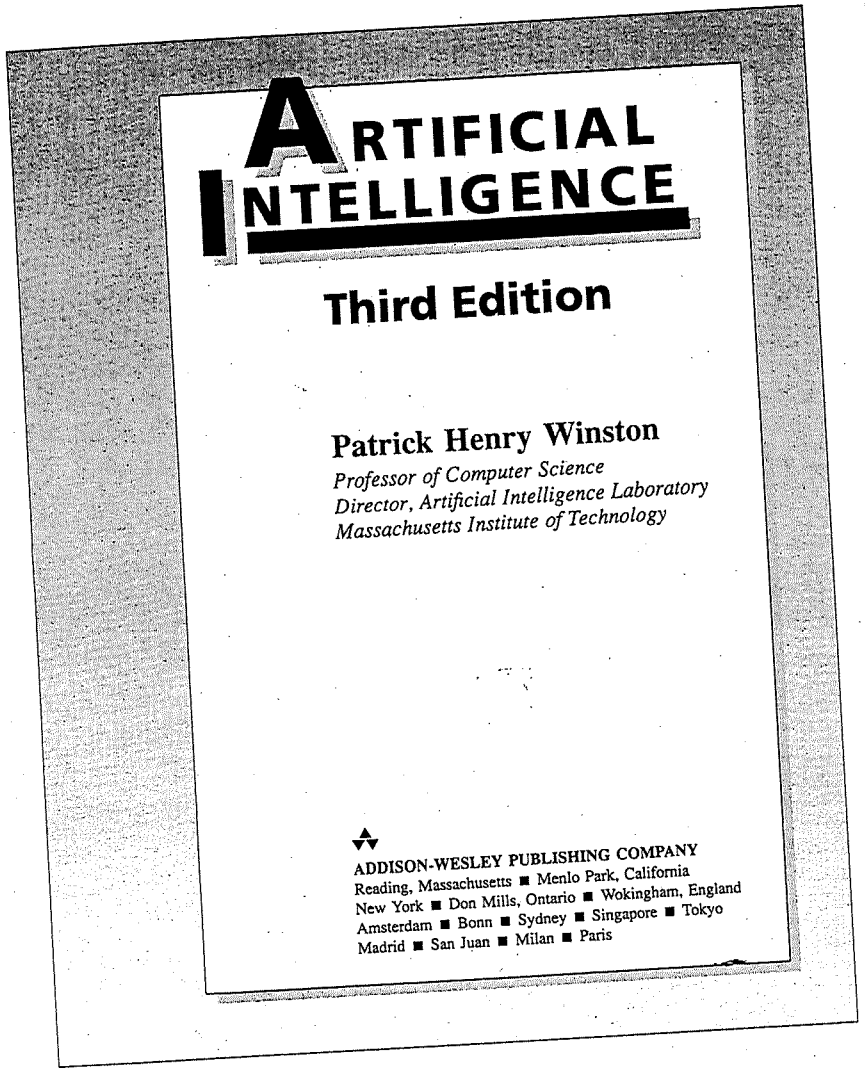
tive example, for none of the three



21 Learning by Building Identification Trees

In this chapter, you learn about a method that enables computers to learn by assembling tests into an *identification tree.* You also learn how an identification tree can be transformed into a *perspicuous set of antecedent–consequent rules.*

Identification-tree building is the most widely used learning method. Thousands of practical identification trees, for applications ranging from medical diagnosis to process control, have been built using the ideas that you learn about in this chapter.

By way of illustration, you see how the SPROUTER and PRUNER procedures construct rules that determine whether a person is likely to be sunburned, given a database of sample people and their physical attributes.

Once you have finished this chapter, you will know how to build identification trees, and how to transform them into rules.

FROM DATA TO IDENTIFICATION TREES

In this section, you learn how to build identification trees by looking for regularities in data.

The World Is Supposed to Be Simple

Imagine that you are somehow unaware of the factors that leave some people red and in pain after a few hours on the beach, while other people just turn tanned and happy. Being curious, you go to the beach and start jotting down notes. You observe that people vary in hair color, height, and weight. Some smear lotion on their bodies; others do not. Ultimately, some turn red. You want to use the observed properties to help you predict whether a new person—one who is not in the observed set—will turn red.

One possibility, of course, is to look for a match between the properties of the new person and those of someone observed previously. Unfortunately, the chances of an exact match are usually slim. Suppose, for example, that your observations produce the information that is listed in the following table:

Name	Hair	Height	Weight	Lotion	Result
Sarah	blonde	average	light	no	sunburned
Dana	blonde	tall	average	yes	none
Alex	brown	short	average	yes	none
Annie	blonde	short	average	no	sunburned
Emily	red	average	heavy	no	sunburned
Pete	brown	tall	heavy	no	none
John	brown	average	heavy	no	none
Katie	blonde	short	light	yes	none

Given that there are three possible hair colors, heights, and weights, and that a person either uses or does not use lotion, there are $3 \times 3 \times 3 \times 2 = 54$ possible combinations. If a new person's properties are selected at random, the probability of an exact match with someone already observed is $8/54 = 0.15$, or just 15 percent.

The probability can be lower in practice, because there can be many more properties and many more possible values for each of those properties. Suppose, for example, that you record a dozen unrelated properties for each observed person, that each property has five possible values, and that each property value appears with equal frequency. Then, there would be $5^{12} = 2.44 \times 10^8$ combinations, and even with a table of 1 million observations, you would find an exact match only about 0.4 percent of the time.

Thus, it can be wildly impractical to classify an unknown object by looking for an exact match between the measured properties of that unknown object and the measured properties of samples of known classification.

You could, of course, treat the data as a feature space in which you look for a close match, perhaps using the approach described in Chapter 2. But if you do not know which properties are important, you may find a close match that is close because of the coincidental alignment of irrelevant properties.

An alternative is to use the version-space method described in Chapter 20 to isolate which properties matter and which do not. But you usually have no a priori reason to believe that a class-characterizing model can be expressed as a single combination of values for a subset of the attributes—nor do you have any reason to believe your samples are noise free.

Still another alternative—the one this chapter focuses on—is to devise a property-testing procedure such that the procedure correctly classifies each of the samples. Once such a procedure works on a sufficient number of samples, the procedure should work on objects whose classification is not yet known.

One convenient way to represent property-testing procedures is to arrange the tests involved in an **identification tree**. Because an identification tree is a special kind of decision tree, the specification refers to the decision-tree specification provided in Chapter 19:

An **identification tree** is a representation

That is a decision tree

In which

▷ Each set of possible conclusions is established implicitly by a list of samples of known class.

For example, in the identification tree shown in figure 21.1, the first test you use to identify burn-susceptible people—the one at the root of the tree—is the hair-color test. If the result is blonde, then you check whether lotion is in use; on the other hand, if the hair-color result is red or brown, you need no subsequent test. In general, the choice of which test to use, if any, depends on the results of previous tests.

Thus, the property-testing procedure embodied in an identification tree is like a railroad switch yard. Each unknown object is directed down one branch or another at each test, according to its properties, like railroad cars at switches, according to their destination.

The identification tree shown in figure 21.1 can be used to classify the people in the sunburn database, because each sunburned person ends up at a leaf node alone or with other sunburned people. Curiously, however, the identification tree shown in figure 21.2 can be used as well, even though it contains tests that have nothing to do with sunburn susceptibility. The identification tree in figure 21.1 seems more reasonable because you know that hair color and exposure are reasonably congruent with sunburn susceptibility.

The identification tree in figure 21.1 seems to us to be better than the one in figure 21.2, but how can a program reach the same conclusion

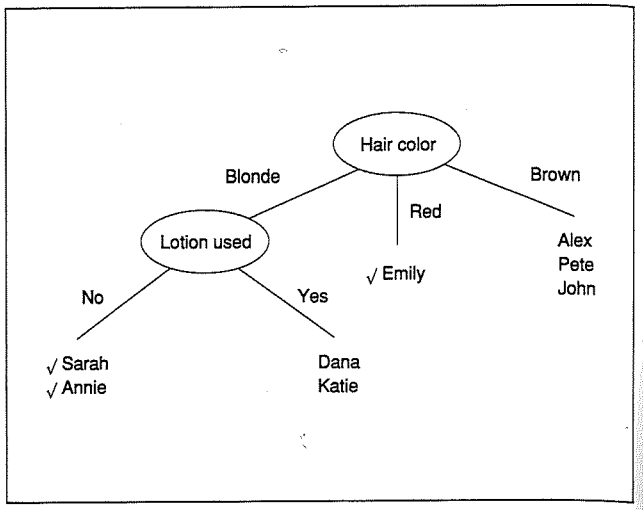


Figure 21.1 An identification tree that is consistent with the sunburn database. This tree is consistent with natural intuitions about sunburn. Each checked name identifies a person who turns red.

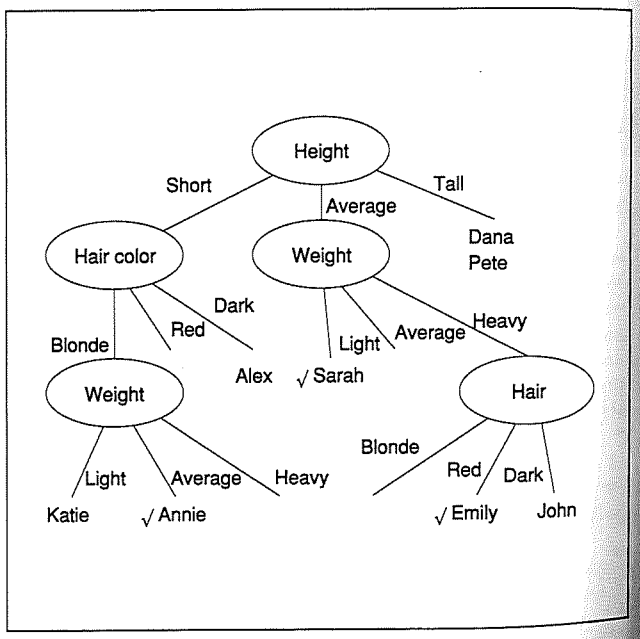


Figure 21.2 Another identification tree that is consistent with the sunburn database, albeit overly large and inconsistent with natural intuitions about sunburn. Each checked name identifies a person who turns red.

without any prior knowledge of what lotion does or how hair color relates to skin characteristics? One answer is to presume a variation on Occam's razor:

Occam's razor, specialized to identification trees:

▷ The world is inherently simple. Therefore the smallest identification tree that is consistent with the samples is the one that is most likely to identify unknown objects correctly.

Thus, the identification tree in figure 21.1, being smaller than the one in figure 21.2, is the tree that is more likely to identify sunburn-susceptible people.

Consequently, the question turns from *which is the right identification tree* to *how can you construct the smallest identification tree?*

Tests Should Minimize Disorder

Unfortunately, it is computationally impractical to find the smallest possible identification tree when many tests are required, so you have to be content with a procedure that tends to build small trees, albeit trees that are not guaranteed to be the smallest possible.

One way to start is to select a test for the root node that does the best job of dividing the database of samples into subsets in which many samples have the same classification. For each set containing more than one kind of sample, you then select another test in an effort to divide that inhomogeneous set into homogeneous subsets.

Consider, for example, the sunburn database and the four candidates for the root test. As shown in figure 21.3, the weight test is arguably the worst if you judge the tests according to how many people end up in homogeneous sets. After you use the weight test, none of the sample people are in a homogeneous set. The height test is somewhat better, because two people are in a homogeneous set; the lotion-used test is still better, because three people are in homogeneous sets. The hair-color test is best, however, because four people—Emily, Alex, Pete, and John—are in homogeneous sets. Accordingly, you use the hair-color test first.

The hair-color test leaves only one inhomogeneous set, consisting of Sarah, Dana, Annie, and Katie. To divide this set further, you consider what each of the remaining three tests does to the four people in the set. The result is shown in figure 21.4.

This time, there can be no doubt. The lotion-used test divides the set into two homogeneous subsets, whereas both the height and weight tests leave at least one inhomogeneous subset.

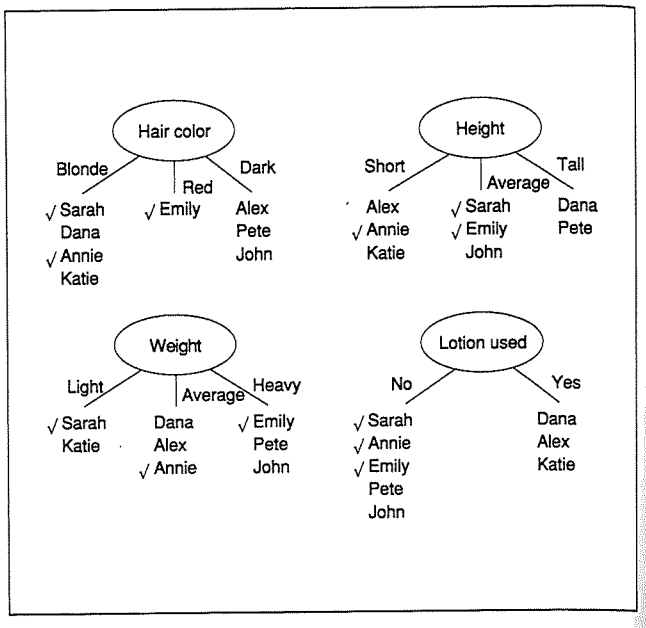


Figure 21.3 Each test divides the sunburn database into different subsets. Each checked name identifies a person who turns red. Intuition suggests that the hair-color test does the best job of dividing the database into homogeneous subsets.

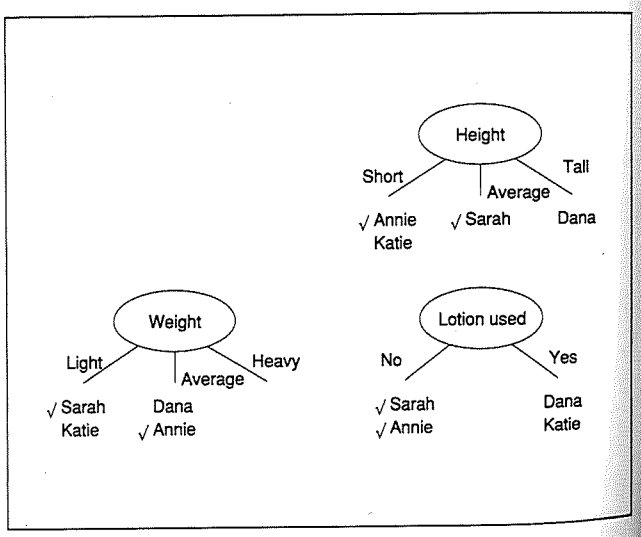


Figure 21.4 Once the blonde-haired people have been isolated, the available tests perform as shown. Each checked name identifies a person who turns red. The lotion-used test plainly does the best job of dividing the blonde-haired set consisting of Sarah, Dana, Annie, and Katie into homogeneous subsets.

Information Theory Supplies a Disorder Formula

For a real database of any size, it is unlikely that any test would produce even one completely homogeneous subset. Accordingly, for real databases, you need a powerful way to measure the total disorder, or inhomogeneity, in the subsets produced by each test. Fortunately, you can borrow the formula you need from information theory:

$$\text{Average disorder} = \sum_b \left(\frac{n_b}{n_t}\right) \times \left(\sum_c -\frac{n_{bc}}{n_b} \log_2 \frac{n_{bc}}{n_b}\right),$$

where

n_b is the number of samples in branch b,

n_t is the total number of samples in all branches,

n_{bc} is the total of samples in branch b of class c.

To see why this borrowed formula works, first confine your attention to the set of samples lying at the end of one branch b. You want a formula involving n_b and n_{bc} that gives you a high number when a test produces highly inhomogeneous sets and a low number when a test produces completely homogeneous sets. The following formula involving n_{bc} and n_b does the job:

$$\text{Disorder} = \sum_c -\frac{n_{bc}}{n_b} \log_2 \frac{n_{bc}}{n_b}.$$

Although there is nothing sacred about this disorder formula, it certainly has desirable features, which is why information-theory experts use a similar formula to measure information.†

To get a feel for the desirable features of the disorder formula, suppose that you have a set that contains members of just two classes, class A and class B. If the number of members from class A and the number of members from class B are perfectly balanced, the measured disorder is 1, the maximum possible value:

$$\begin{aligned}\text{Disorder} &= \sum_c -\frac{n_{bc}}{n_b} \log_2 \frac{n_{bc}}{n_b} \\ &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} \\ &= \frac{1}{2} + \frac{1}{2} \\ &= 1.\end{aligned}$$

†In information theory, the disorder formula is sacred: It is the only formula that satisfies certain general properties. The requirements imposed by heuristic tree building are not so stringent, however.

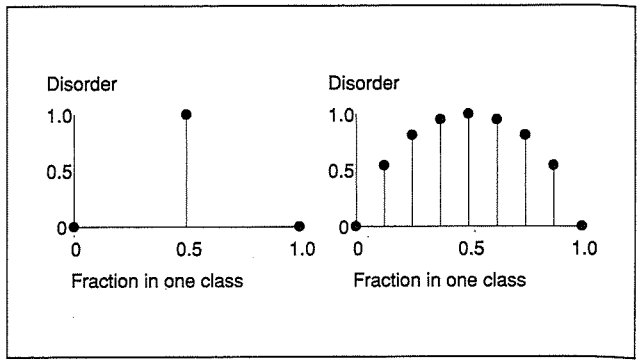


Figure 21.5 The disorder in a set containing members of two classes A and B, as a function of the fraction of the set belonging to class A. On the left, the total number of samples in both classes combined is two; on the right, the total number of samples in both classes is eight.

On the other hand, if there are only As or only Bs, the measured disorder is 0, the minimum possible value, because, in the limit, as x approaches zero, $x \times \log_2(x)$ is zero:

$$\begin{aligned}\text{Disorder} &= \sum_c -\frac{n_{bc}}{n_b} \log_2 \frac{n_{bc}}{n_b} \\ &= -1 \log_2 1 - 0 \log_2 0 \\ &= -0 - 0 \\ &= 0.\end{aligned}$$

As you move from perfect balance and perfect homogeneity, disorder varies smoothly between zero and one, as shown in figure 21.5. The disorder is zero when the set is perfectly homogeneous, and the disorder is one when the set is perfectly inhomogeneous.

Now that you have a way of measuring the disorder in one set, you can measure the average disorder of the sets at the ends of the branches under a test. You simply weight the disorder in each branch's set by the size of the set relative to the total size of all the branches' sets. In the following formula, n_b is the number of samples that the test sends down branch b, and n_t is the total number of samples in all branches:

$$\text{Average disorder} = \sum_b \frac{n_b}{n_t} \times (\text{Disorder in the branch } b \text{ set}).$$

Substituting for the disorder in the branch b set, you have the desired formula for average disorder.

Now you can compute the average disorder produced when each test is asked to work on the complete sample set. Looking back at figure 21.3, note that the hair-color test divides those people into three sets. In the blonde set, two people turn red and two do not. In the red-haired set, there is only one person and that person turns red. In the brown-haired set, all three people are unaffected.

Hence, the average disorder produced by the hair-color test when the complete sample set is used is 0.5:

$$\begin{aligned}\text{Average disorder} &= \frac{4}{8}\left(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right) \\ &\quad + \frac{1}{8} \times 0 \\ &\quad + \frac{3}{8} \times 0 \\ &= 0.5.\end{aligned}$$

Working out the result for the other tests yields the following results:

Test	Disorder
Hair	0.5
Height	0.69
Weight	0.94
Lotion	0.61

Because the hair test clearly produces the least average disorder, the hair test is the first that should be used, which is consistent with the previous informal analysis. Similarly, once the hair test is selected, the choice of another test to separate out the sunburned people from among Sarah, Dana, Annie, and Katie is decided by the following calculations:

Test	Disorder
Height	0.5
Weight	1
Lotion	0

Thus, the lotion-used test is the clear winner. Using the hair test and the lotion-used tests together ensures the proper identification of all the samples.

In summary, to generate an identification tree, execute the following procedure, named SPROUTER:

To generate an identification tree using SPROUTER,

- ▷ Until each leaf node is populated by as homogeneous a sample set as possible:
  - ▷ Select a leaf node with an inhomogeneous sample set.
  - ▷ Replace that leaf node by a test node that divides the inhomogeneous sample set into minimally inhomogeneous subsets, according to some measure of disorder.
