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

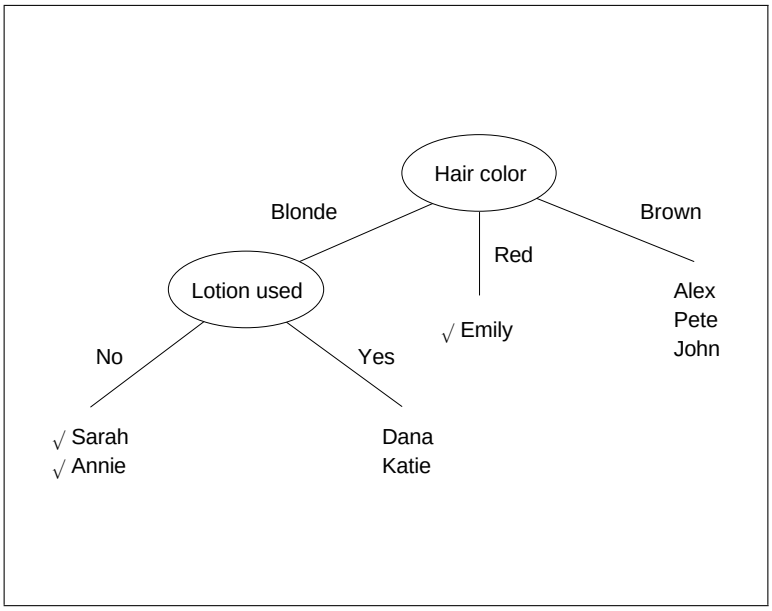


Figure 21.1 An identification tree that is consistent with the sunburn database. This tree is consistent with natural intuitions about sunburn. Each checked name identifies a person who turns red.

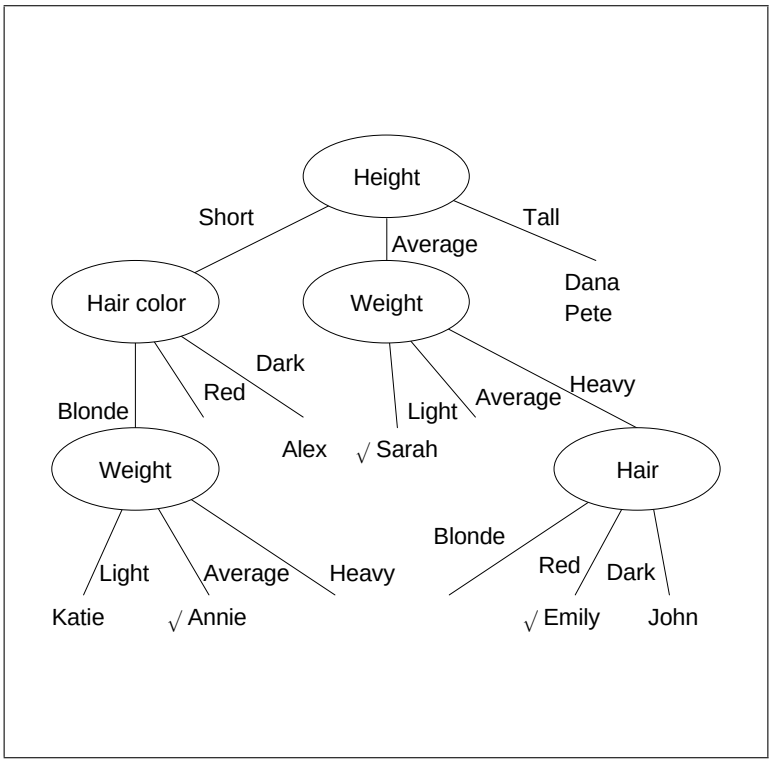


Figure 21.2 Another identification tree that is consistent with the sunburn database, albeit overly large and inconsistent with natural intuitions about sunburn. Each checked name identifies a person who turns red.

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.

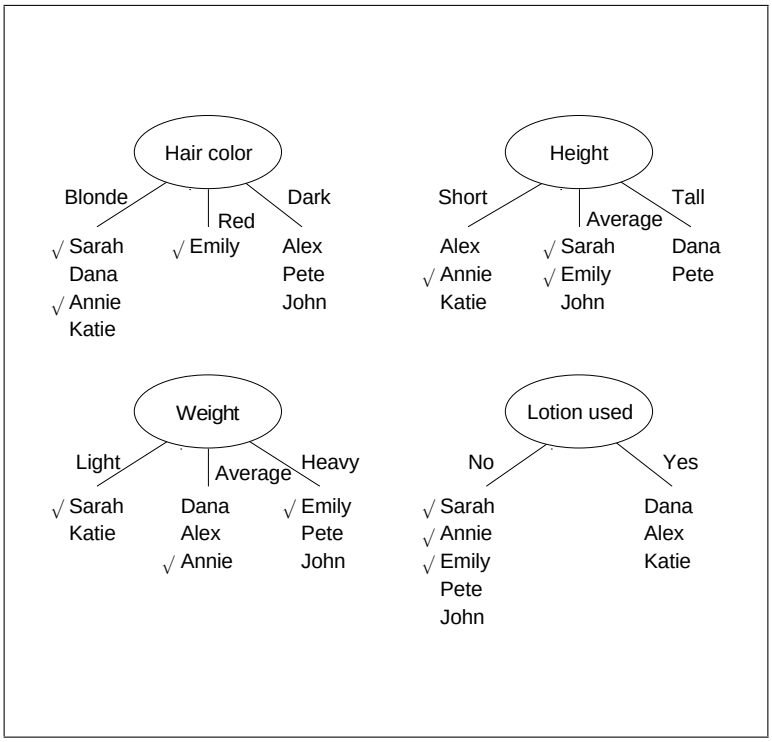


Figure 21.3 Each test divides the sunburn database into different subsets. Each checked name identifies a person who turns red. Intuition suggests that the hair-color test does the best job of dividing the database into homogeneous subsets.

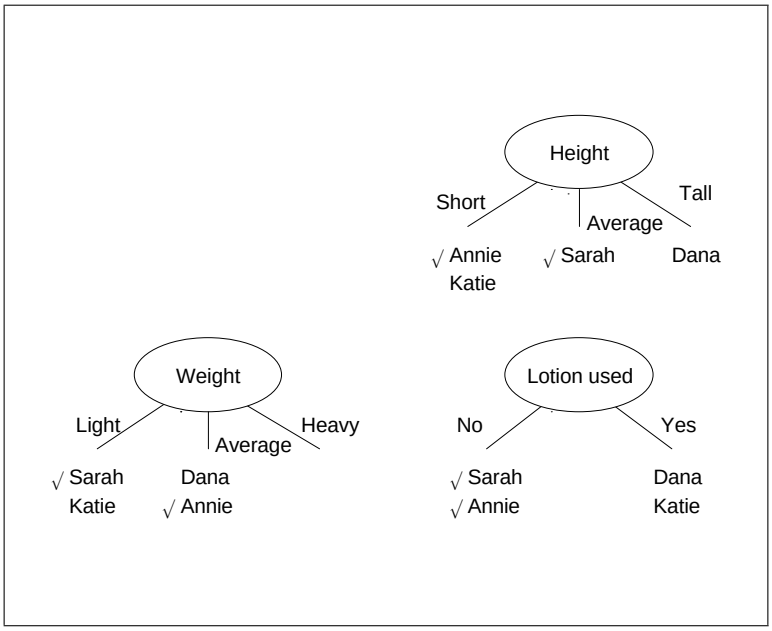


Figure 21.4 Once the blonde-haired people have been isolated, the available tests perform as shown. Each checked name identifies a person who turns red. The lotion-used test plainly does the best job of dividing the blonde-haired set consisting of Sarah, Dana, Annie, and Katie into homogeneous subsets.

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:

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

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.

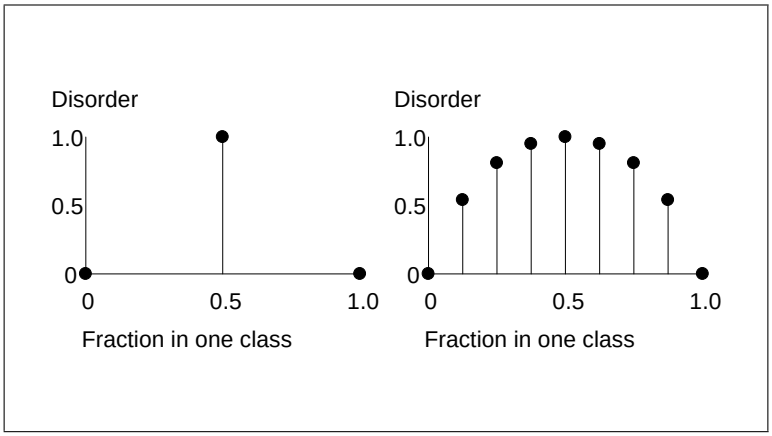


Figure 21.5 The disorder in a set containing members of two classes A and B, as a function of the fraction of the set belonging to class A. On the left, the total number of samples in both classes combined is two; on the right, the total number of samples in both classes is eight.

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.

FROM TREES TO RULES

Once an identification tree is constructed, it is a simple matter to convert it into a set of equivalent rules. You just trace each path in the identification tree, from root node to leaf node, recording the test outcomes as antecedents and the leaf-node classification as the consequent. For the sunburn illustration, the four rules corresponding to the four paths in the identification tree are as follows:

If the person's hair color is blonde
the person uses lotion
then nothing happens

If the person's hair color is blonde
the person uses no lotion
then the person turns red

If the person's hair color is red
then the person turns red

If the person's hair color is brown
then nothing happens

In the rest of this section, you learn how to simplify such rule sets so as to increase transparency and to decrease errors.

Unnecessary Rule Antecedents Should Be Eliminated

Once a rule set is devised, you can try to simplify that set by simplifying each rule and then eliminating useless rules. To simplify a rule, you ask whether any of the antecedents can be eliminated without changing what the rule does on the samples.

Two of the rules have two antecedents. For each of the two, you ask whether both antecedents are really necessary. Consider, for example, the two antecedents in the following rule:

If the person's hair color is blonde
the person uses lotion
then nothing happens

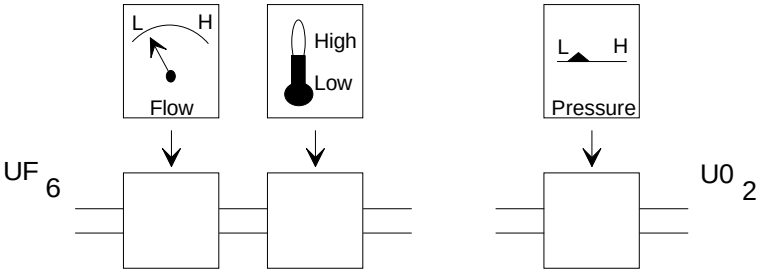
If you eliminate the first antecedent, the one about blonde hair, the rule triggers for each person who uses lotion. Three of the sample people use lotion: Dana, Alex, and Katie, none of whom turn red. Because none turn red, it cannot be that hair color matters, so the dropped antecedent that checks for blonde hair is unnecessary. Dropping that antecedent produces the following, simplified rule:

If the person uses lotion
then nothing happens

APPLICATION

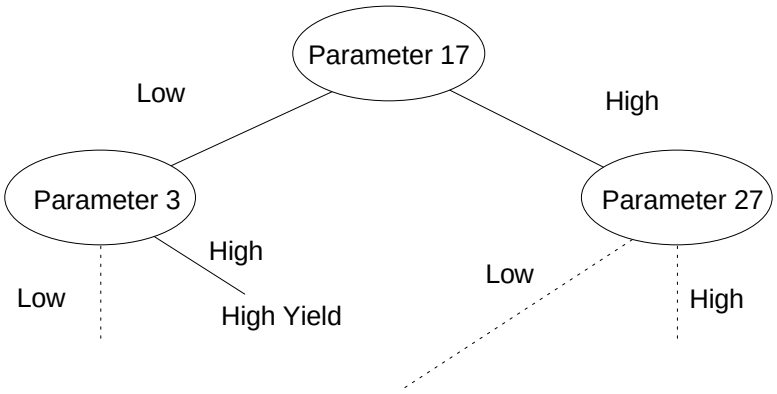
Optimizing a Nuclear Fuel Plant

Programs resembling SPROUTER can be used to identify key parameters in chemical processing. Westinghouse used such a program to improve yield at a plant in which uranium hexafluoride gas is converted into uranium-dioxide fuel pellets. Approximately six processing steps are required to do the conversion, and among these processing steps, there are approximately 30 controllable temperatures, pressures, and flow rates:



Historically, process engineers noted that yield was high on some days and low on others; of course, they wanted to control the 30 parameters so as to guarantee high yield every day. Unfortunately, no one knew quite what to do. Worse yet, nuclear fuel plants are not something with which to play, so experiments were forbidden.

Fortunately, SPROUTER was able to use plant records to build an identification tree to determine, on the basis of the parameters, when yield is high or low. In the schematic identification tree example that follows, each test decides whether a particular parameter value is high or low with respect to a threshold. Each of the thresholds is determined by SPROUTER itself so as to produce the simplest tree:



Once such a tree is in hand, it is easy to convert identification into control. You just have to find the shortest path from the root of the tree to one of the high-yield subsets. In this schematic example, you can guarantee high yield by keeping parameter 17 low and parameter 3 high. In the Westinghouse experience, this approach was a spectacular success: Their entire investment was recovered in the first half-day of improved yield.

To make such reasoning easier, it is often helpful to construct what statisticians call a **contingency table**, so called because it shows the degree to which a result is contingent on a property. In the following contingency table you see the number of lotion users who are blonde and not blonde, and the number of lotion users who are sunburned and not sunburned. The table clearly shows that knowledge about whether a person is blonde has no bearing on determining whether the person becomes sunburned given that the person uses lotion.

	No change	Sunburned
Person is blonde	2	0
Person is not blonde	1	0

Now consider the second antecedent in the same rule, the one that checks for lotion. If you eliminate it, the rule triggers whenever the person is blonde. Among the four blonde people, Sarah and Annie, neither of whom use lotion, are both sunburned; on the other hand, Dana and Katie, both of whom do use lotion, are not sunburned. Here is the contingency table:

	No change	Sunburned
Person uses lotion	2	0
Person uses no lotion	0	2

Plainly, the lotion antecedent has a bearing on the result for those people who are blonde. The samples who are blonde are not sunburned if and only if they use lotion. Accordingly, the dropped antecedent does make a difference, and you cannot eliminate it.

Now turn to the other two-antecedent rule; it triggers on blondes who do not use lotion:

If the person's hair color is blonde
the person does not use lotion
then the person turns red

As before, you explore what happens as antecedents are eliminated one at a time. Eliminating the first antecedent produces a rule that looks for people who do not use lotion. Of the five who do not, both blondes are sunburned; among the other three, one is sunburned and two are not:

	No change	Sunburned
Person is blonde	0	2
Person is not blonde	2	1

Evidently the dropped antecedent is important. Without it, you cannot be sure that a person who matches the rule is going to be burned.

Eliminating the second antecedent produces a rule that looks for people who are blonde. Of the four who are, two turn red and two do not:

	No change	Sunburned
Person uses no lotion	0	2
Person uses lotion	2	0

Again, the dropped antecedent is important. You conclude that the rule must remain as is; any simplification makes the rule fail on some of the sample people.

Finally, you need to look at the one-antecedent rules:

If the person's hair color is red
then the person turns red

If the person's hair color is brown
then nothing happens

If a rule has one antecedent and that antecedent is dropped, then, by convention, the rule is always triggered. Hence, the contingency tables for the two rules both contain all eight samples:

	No change	Sunburned
Person is red haired	0	1
Person is not red haired	5	2

	No change	Sunburned
Person is brown haired	3	0
Person is not brown haired	2	3

Repeating what you have done with two antecedent rules, you retain the red-hair antecedent in the first of these two rules, as well as the brown-hair antecedent in the second. Of course, these results are obvious in any case, for a rule with no antecedents will work correctly only if all the samples have the same result.

Unnecessary Rules Should Be Eliminated

Once you have simplified individual rules by eliminating antecedents that do not matter, you need to simplify the entire rule set by eliminating entire rules. For the sunburn illustration, the four candidate rules, one of which has been simplified, are as follows:

If the person's hair color is blonde
the person uses no lotion
then the person turns red

If the person uses lotion
then nothing happens

If the person's hair color is red
then the person turns red

If the person's hair color is brown
then nothing happens

In this example, note that two rules have consequents that indicate that a person will turn red, and that two rules have consequents that indicate that nothing will happen. You can replace the two that indicate a person will turn red with a **default rule**, one that is to be used only if no other rule applies. Because there are two possible results in the example, there are two choices:

If no other rule applies
then the person turns red

If no other rule applies
then nothing happens

In general, it makes sense to choose the default rule that eliminates as many other rules as possible; in the example, however, because both of the possible conclusions are indicated by two rules, you must use some other, tie-breaking criterion. One obvious tie breaker is to choose the default rule that covers the most common consequent in the sample set, which happens to be that nothing happens. In the example, this produces the following simplified rule set:

If the person's hair color is blonde
the person uses no lotion
then the person turns red

If the person's hair color is red
then the person turns red

If no other rule applies
then nothing happens

Another obvious tie breaker is to choose the default rule that produces the simplest rules, perhaps as measured by the total number of antecedents. In the example, this choice produces the following simplified rule set:

If the person uses lotion
then nothing happens

If the person's hair color is brown
then nothing happens

If no other rule applies
then the person turns red

In summary, to convert an identification tree into a rule set, execute the following procedure, named PRUNER:

To generate rules from an identification tree using PRUNER,

- ▷ Create one rule for each root-to-leaf path in the identification tree.
- ▷ Simplify each rule by discarding antecedents that have no effect on the conclusion reached by the rule.
- ▷ Replace those rules that share the most common consequent by a default rule that is triggered when no other rule is triggered. In the event of a tie, use some heuristic tie breaker to choose a default rule.

Fisher's Exact Test Brings Rule Correction in Line with Statistical Theory

Now let us leave the sunburn example to consider the following table, which relates presence or absence of a certain result, R, to the presence or absence of a certain property, P. Suppose that you denote the presence of the result by R_1 and its absence by R_2. Similarly, suppose you denote the presence of the property by P_1 and its absence by P_2. Then you have, in general, the following contingency table:

	R_1	R_2
P_1	l	m
P_2	n	o

Now the question is this: Do the values of l, m, n, and o indicate that knowing about P is relevant to determining R? Consider, for example, the following contingency table:

	R_1	R_2
P_1	1	0
P_2	0	1

On the surface, if you use this table to decide whether to keep an antecedent testing for P in a rule, it seems to indicate that you should keep the antecedent, because, without the antecedent, the rule would misclassify an example. But now consider the following contingency table:

	R_1	R_2
P_1	999	0
P_2	0	1

Without the antecedent testing for P, you would again misclassify a sample, but this time only one sample in 1000, rather than one in two. Is

a simplification worth an occasional error? Or is the table entry at the intersection of column P_2 and row R_2 caused by noisy measurement?

And speaking of noise, are two examples really sufficient for you to decide whether an antecedent should be retained? Should you reach the same conclusion with two contingency tables, both of which have the same numbers from a relative point of view, but one of which has 1000 times more data, as in the following pair?

	R_1	R_2
P_1	1	0
P_2	0	1

	R_1	R_2
P_1	1000	0
P_2	0	1000

After thinking about such questions, you might decide on a strategy that considers both the sizes of the entries and their relative sizes. To be conservative, if all numbers are small, you probably should get rid of an antecedent rather than treat it as though it were solidly supported. Similarly, if the ratio of l to m is the same or nearly the same as the ratio of n to o, knowing about P is not helpful, and you should probably get rid of the antecedent. On the other hand, if the numbers are large and if l/m is very different from n/o, then knowing about P is quite enlightening, and you should keep the antecedent.

To put this sort of reasoning on solid ground, you should consult a statistician, who might take you through an analysis that leads, in several steps, to **Fisher's exact test**. The following paragraphs sketch those steps.

First, think about your goal. One plausible goal is to determine whether there is a statistical dependence between the result R and the property P. Unfortunately, if there is a statistical dependence, you probably have no clue about which of an infinite number of forms that dependence might take, which means you do not know exactly for what you are to test.

Fortunately, statistical *in*dependence has only one form, making independence much easier to deal with than dependence. Accordingly, your statistician tells you to look for statistical dependence indirectly, through a double negative. Instead of trying to show that the result, R, depends on the property, P, you try to show that it is *un*likely that R does *not* depend on P. Said in another way, your goal is to decide whether your samples cast significant doubt on the independence hypothesis.[†]

Your second step is to ask about the probability of observing a particular combination, l, m, n, o, given that R is independent of P. To

[†]A statistician would say that your goal is to perform a significance test on the null hypothesis. I cannot think why.

say something about that probability, however, you have to make more assumptions, because with four things that can vary, the problem is still severely underconstrained, even with independence assumed.

The standard approach is to assume that there is a certain fixed number of samples corresponding to P_1, $S_{P_1} = l + m$, a certain fixed number corresponding to P_2, $S_{P_2} = n + o$, and a certain fixed number corresponding to R_1, $S_{R_1} = l + n$. Of course, these assumptions fix the number corresponding to R_2, $S_{R_2} = m + o$, inasmuch as $S_{P_1} + S_{P_2}$ must be equal to $S_{R_1} + S_{R_2}$.

These extra assumptions are equivalent to saying that the **marginal sums** of the contingency table are constants:

	R_1	R_2	Marginal sum
P_1	l	m	$S_{P_1} = l + m$
P_2	n	o	$S_{P_2} = n + o$
Marginal sum	$S_{R_1} = l + n$	$S_{R_2} = m + o$	$S_{P_1} + S_{P_2} = S_{R_1} + S_{R_2}$

Once you have fixed the size of the marginal sums, you are free to choose a value for only one of l or m or n or o, which then, in cooperation with the marginal sums, determines the rest.

Suppose you pick a value for l, the number of samples with result R_1 and property P_1. Then, your statistician tells you that the following probability formula, grimly full of factorials, provides the probability for your value for l given the marginal sums:

$$p(l|S_{P_1}, S_{P_2}, S_{R_1}, S_{R_2}) = \frac{\frac{S_{P_1}!}{l!(S_{P_1}-l)!} \times \frac{S_{P_2}!}{(S_{R_1}-l)!(S_{P_2}-(S_{R_1}-l))!}}{\frac{(S_{P_1}+S_{P_2})!}{S_{R_1}!(S_{P_1}+S_{P_2}-S_{R_1})!}}.$$

Note that the formula does not involve S_{R_2}, because S_{R_2} is determined by the other marginal sums.

With the formula, you can plot, as in figure 21.6, the probabilities for particular values of l given independence and $S_{R_1} = S_{R_2} = S_{P_1} = S_{P_2} = 10$.

Of course, whenever the values for S_{P_1} and S_{P_2} are unequal, the symmetry disappears—as shown, for example, in figure 21.7.

Your third step is to note that the combined probability of all extremely high and low values of l is low. In the symmetric example—the one with 20 samples—the probability that $l > 7$ is less than 0.025. Also, the probability that $l < 3$ is less than 0.025. Thus, the probability that l is outside the three-to-seven range is less than 0.05, given that the property and the result are independent.

If it is unlikely, however, that the observed value of l is outside the central range, given independence, then, if the observed value actually is outside the central range, independence must not be likely. More precisely, if you say that the property and the result are independent whenever the observed value of l is outside the central range, then the probability of

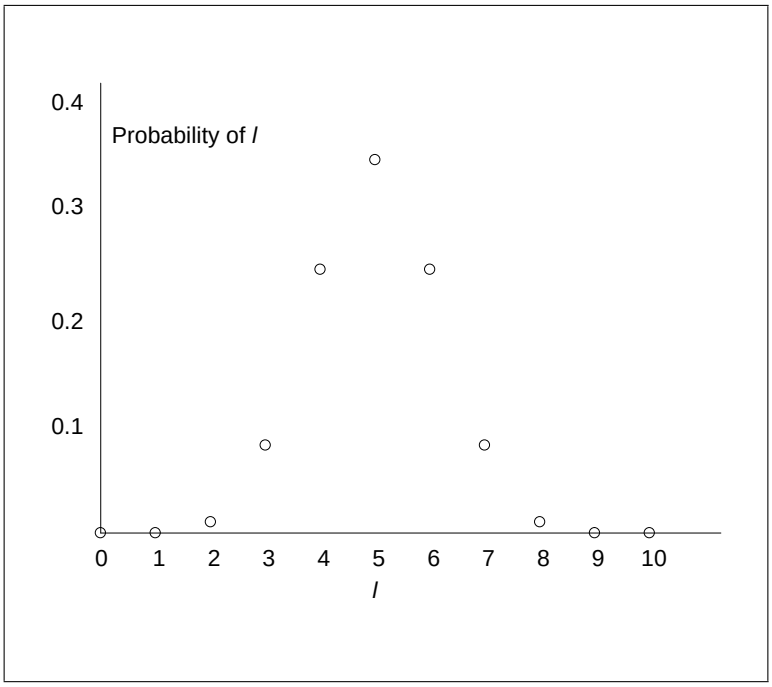


Figure 21.6 The probability of l samples exhibiting both a certain property and result, given that 10 samples have the property, 10 do not, 10 samples exhibit the result, and 10 do not, for a total of 20 samples.

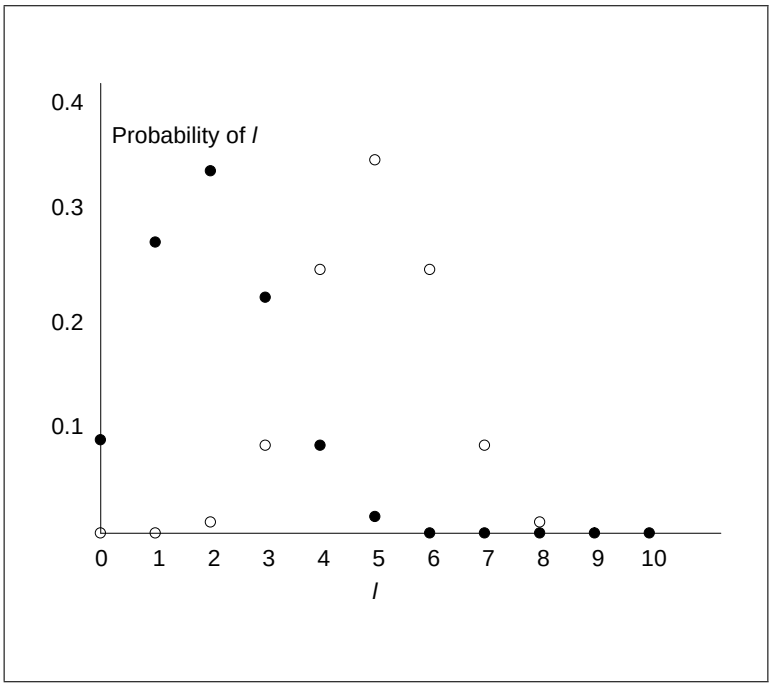


Figure 21.7 The open circles show the probability of *l* samples exhibiting both a certain property and result, given that 10 samples have the property, 10 do not, 10 samples exhibit the result, and 10 do not, for a total of 20 samples. The filled circles show the probability of *l* samples exhibiting both a certain property and result, given that 10 samples have the property, 40 do not, 10 samples exhibit the result, and 40 do not, for a total of 50 samples.

blundering when the property and the result actually are independent is less than 0.05.

Or, saying it still another way, if l lies outside the central range, you can say that the property and the result are statistically dependent with less than a five percent chance of wrongfully ruling out the independence hypothesis. Your statistician says that the observed value is significant at the 5-percent level using Fisher's exact test.

Thus, the following contingency table is seriously unlikely, given independence. Whenever you see such a table, you should retain an antecedent involving a property P.

	R_1	R_2	Marginal sum
P_1	2	8	10
P_2	8	2	10
Marginal sum	10	10	20

On the other hand, the following contingency table is reasonable, given independence. Following your statistician's line of reasoning, you should drop an antecedent involving a property P.

	R_1	R_2	Marginal sum
P_1	4	6	10
P_2	6	4	10
Marginal sum	10	10	20

Not surprisingly, when you use the test on the antecedents in the rules derived from the sunburn example, you eliminate all the antecedents, for there just are not enough data to say that there is significant evidence in favor of rejecting the conservative assumption of independence. On the other hand, were there five times as many data, with all results increased in proportion, you would reinstate all the antecedents that seemed important when reasoning without statistics.

Recall, for example, what happens when you drop the first antecedent in the following rule:

If the person's hair color is blonde
the person uses lotion
then nothing happens

Given the original data, the contingency table is as follows:

	No change	Sunburned
Person uses lotion	2	0
Person uses no lotion	0	2

With this table, l can be only 0, 1, or 2, and the probabilities are such that the central region covers all three values. There is no value of l such that the independence hypothesis is unlikely.

On the other hand, if there are five times as many data, increased in proportion, then the contingency table is as follows:

	No change	Sunburned
Person uses lotion	10	0
Person uses no lotion	0	10

With this table, l can take on any value from 0 to 10, and the probabilities are such that the central region ranges from 3 to 7. Given $l = 10$, the independence hypothesis is unlikely.

SUMMARY

- According to Occam's razor, the world is simple. Thus, the simplest explanation that covers the data is likely to be the right explanation.
- One way to recognize situations is to apply the sequence of tests dictated by an identification tree. One way to learn is to build an identification tree, keeping it simple in harmony with Occam's razor.
- One way to build a simple identification tree is to use a disorder formula, borrowed from information theory, to determine which tests to include in the tree.
- Once an identification tree is built, you usually should convert it into a simple set of rules so as to make the knowledge embedded in it more comprehensible. To do the conversion, you make a rule for each path through the tree, and then you simplify the resulting set of rules.
- To simplify a set of rules, you first eliminate unnecessary rule antecedents using Fisher's exact test. Then, you eliminate unnecessary rules.

BACKGROUND

The discussion of decision trees is based on the work of Ross Quinlan on ID3 and other decision-tree systems [1979, 1983]. Quinlan has worked out many variations on the same idea using improved measures of tree quality.

Also, Quinlan and Ronald L. Rivest have worked out an alternative approach based on finding a tree that enables identification using the minimum memory [1987].

The discussion of rule extraction from decision trees is also based on Quinlan's work [1986]. A good description of Fisher's exact test is hard to find, but some large libraries have an instructive pamphlet by Finney et al. [1963].

The nuclear-fuel plant application is the work of W. J. Leech and his associates [1986].