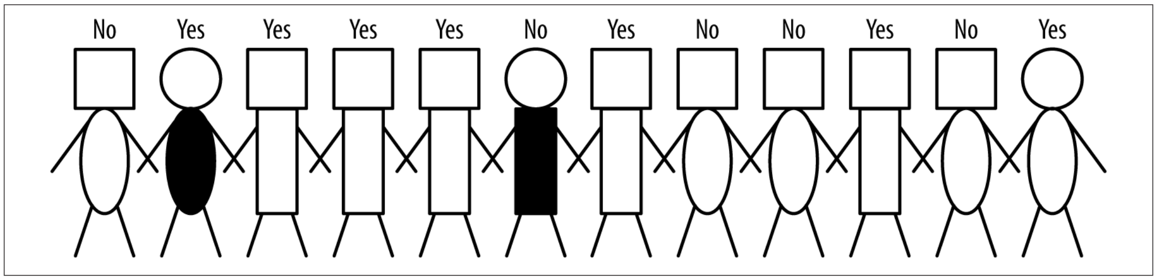

Figure 3-2. A set of people to be classified. The label over each head represents the value of the target variable (write-off or not). Colors and shapes represent different predictor attributes.

Consider just the selection of the single most informative attribute. Solving this problem will introduce our first concrete data mining technique—simple, but easily extendable to be very useful. In our example, what variable gives us the most information about the future churn rate of the population? Being a professional? Age? Place of residence? Income? Number of complaints to customer service? Amount of overage charges?

We now will look carefully into one useful way to select informative variables, and then later will show how this technique can be used repeatedly to build a supervised segmentation. While very useful and illustrative, please keep in mind that direct, multivariate supervised segmentation is just one application of this fundamental idea of selecting informative variables. This notion should become one of your conceptual tools when thinking about data science problems more generally. For example, as we go forward we will delve into other modeling approaches, ones that do not incorporate variable selection directly. When the world presents you with very large sets of attributes, it may be (extremely) useful to harken back to this early idea and to select a subset of informative attributes. Doing so can substantially reduce the size of an unwieldy dataset, and as we will see, often will improve the accuracy of the resultant model.

Selecting Informative Attributes

Given a large set of examples, how do we select an attribute to partition them in an informative way? Let's consider a binary (two class) classification problem, and think about what we would like to get out of it. To be concrete, Figure 3-2 shows a simple segmentation problem: twelve people represented as stick figures. There are two types of heads: square and circular; and two types of bodies: rectangular and oval; and two of the people have gray bodies while the rest are white.

These are the attributes we will use to describe the people. Above each person is the binary target label, *Yes* or *No*, indicating (for example) whether the person becomes a loan write-off. We could describe the data on these people as:

- Attributes:
  - head-shape: square, circular
  - body-shape: rectangular, oval
  - body-color: gray, white
- Target variable:
  - write-off: Yes, No

So let's ask ourselves: which of the attributes would be best to segment these people into groups, in a way that will distinguish write-offs from non-write-offs? Technically, we would like the resulting groups to be as *pure* as possible. By pure we mean *homogeneous with respect to the target variable*. If every member of a group has the same value for the target, then the group is pure. If there is at least one member of the group that has a different value for the target variable than the rest of the group, then the group is impure.

Unfortunately, in real data we seldom expect to find a variable that will make the segments pure. However, if we can reduce the impurity substantially, then we can both learn something about the data (and the corresponding population), and importantly for this chapter, we can use the attribute in a predictive model—in our example, predicting that members of one segment will have higher or lower write-off rates than those in another segment. If we can do that, then we can for example offer credit to those with the lower predicted write-off rates, or can offer different credit terms based on the different predicted write-off rates.

Technically, there are several complications:

1. Attributes rarely split a group perfectly. Even if one subgroup happens to be pure, the other may not. For example, in Figure 3-2, consider if the second person were not there. Then *body-color=gray* would create a pure segment (*write-off=no*). However, the other associated segment, *body-color=white*, still is not pure.
2. In the prior example, the condition *body-color=gray* only splits off one single data point into the pure subset. Is this better than another split that does not produce any pure subset, but reduces the impurity more broadly?
3. Not all attributes are binary; many attributes have three or more distinct values. We must take into account that one attribute can split into two groups while another might split into three groups, or seven. How do we compare these?
4. Some attributes take on numeric values (continuous or integer). Does it make sense to make a segment for every numeric value? (No.) How should we think about creating supervised segmentations using numeric attributes?

Fortunately, for classification problems we can address all the issues by creating a formula that evaluates how well each attribute splits a set of examples into segments, with respect to a chosen target variable. Such a formula is based on a *purity measure*.

The most common splitting criterion is called *information gain*, and it is based on a purity measure called *entropy*. Both concepts were invented by one of the pioneers of information theory, Claude Shannon, in his seminal work in the field (Shannon, 1948).

Entropy is a measure of disorder that can be applied to a set, such as one of our individual segments. Consider that we have a set of *properties* of members of the set, and each member has one and only one of the properties. In supervised segmentation, the member properties will correspond to the values of the target variable. Disorder corresponds to how mixed (impure) the segment is with respect to these properties of interest. So, for example, a mixed up segment with lots of write-offs and lots of non-write-offs would have high entropy.

More technically, entropy is defined as:

Equation 3-1. Entropy

$$entropy = - p_1 \log(p_1) - p_2 \log(p_2) - \cdots$$

Each p_i is the probability (the relative percentage) of property i within the set, ranging from $p_i = 1$ when all members of the set have property i, and $p_i = 0$ when no members of the set have property i. The … simply indicates that there may be more than just two properties (and for the technically minded, the logarithm is generally taken as base 2).

Since the entropy equation might not lend itself to intuitive understanding, Figure 3-3 shows a plot of the entropy of a set containing 10 instances of two classes, + and –. We can see then that entropy measures the general disorder of the set, ranging from zero at minimum disorder (the set has members all with the same, single property) to one at maximal disorder (the properties are equally mixed). Since there are only two classes, $p_+ = 1 - p_-$. Starting with all negative instances at the lower left, $p_+ = 0$, the set has minimal disorder (it is pure) and the entropy is zero. If we start to switch class labels of elements of the set from – to +, the entropy increases. Entropy is maximized at 1 when the instance classes are balanced (five of each), and $p_+ = p_- = 0.5$. As more class labels are switched, the + class starts to predominate and the entropy lowers again. When all instances are positive, $p_+ = 1$ and entropy is minimal again at zero.

As a concrete example, consider a set S of 10 people with seven of the *non-write-off* class and three of the *write-off* class. So:

$$\begin{aligned} p(\text{non-write-off}) &= 7 / 10 = 0.7 \\ p(\text{write-off}) &= 3 / 10 = 0.3 \end{aligned}$$

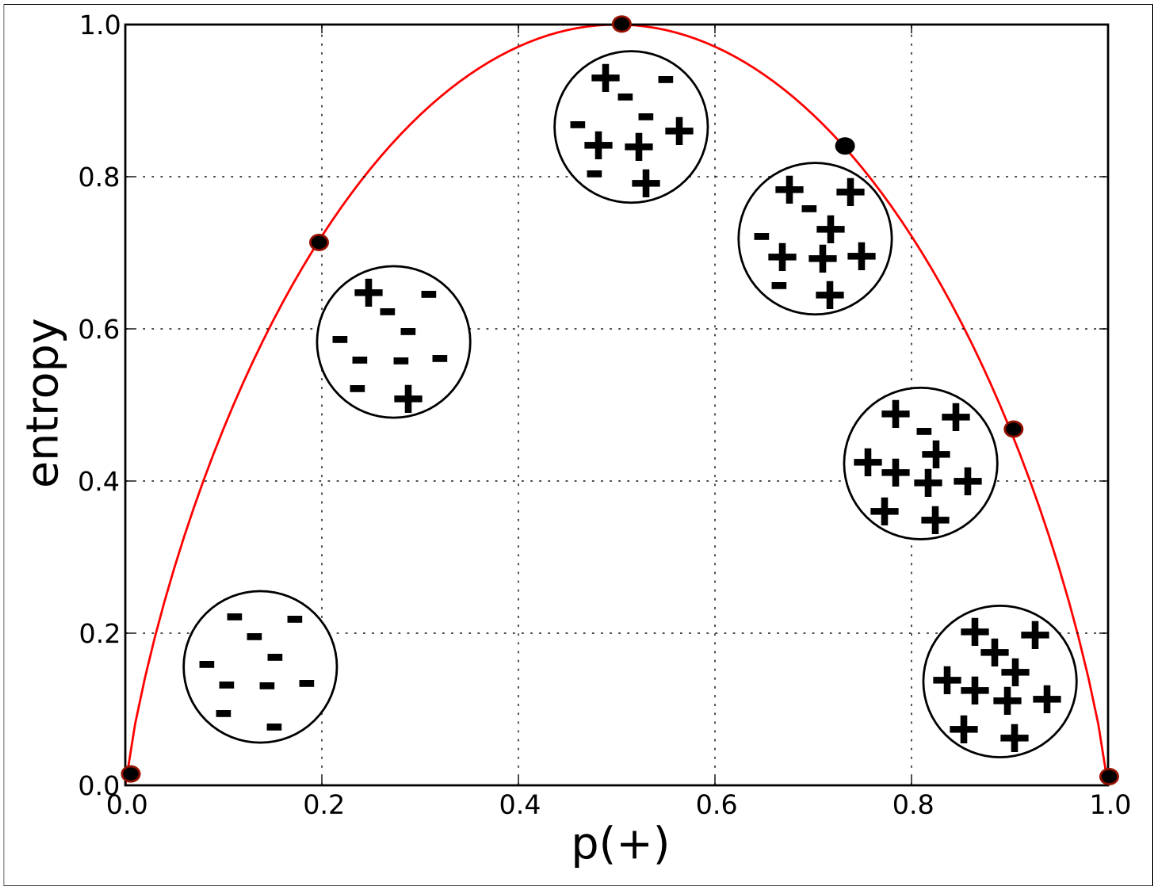

Figure 3-3. Entropy of a two-class set as a function of p(+).

$$\begin{aligned} entropy(S) &= -[0.7 \times \log_2 (0.7) + 0.3 \times \log_2 (0.3)] \\ &\approx -[0.7 \times -0.51 + 0.3 \times -1.74] \\ &\approx 0.88 \end{aligned}$$

Entropy is only part of the story. We would like to measure how *informative* an attribute is with respect to our target: how much gain in information it gives us about the value of the target variable. An attribute segments a set of instances into several subsets. Entropy only tells us how impure one individual subset is. Fortunately, with entropy to measure how disordered any set is, we can define *information gain* (IG) to measure how much an attribute improves (decreases) entropy over the whole segmentation it creates. Strictly speaking, information gain measures the *change* in entropy due to any amount of new information being added; here, in the context of supervised segmentation, we consider the information gained by splitting the set on all values of a single attribute. Let's say the attribute we split on has *k* different values. Let's call the original set of examples the *parent* set, and the result of splitting on the attribute values the *k children* sets. Thus, information gain is a function of both a parent set and of the children resulting from some partitioning of the parent set—how much information has this attribute provided? That depends on how much purer the children are than the parent. Stated in the context of predictive modeling, if we were to know the value of this attribute, how much would it increase our knowledge of the value of the target variable?

Specifically, the definition of information gain (IG) is:

Equation 3-2. Information gain

$$\begin{aligned} IG(parent, children) &= entropy(parent) - \\ &\quad [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \cdots] \end{aligned}$$

Notably, the entropy for each child (c_i) is weighted by the proportion of instances belonging to that child, $p(c_i)$. This addresses directly our concern from above that splitting off a single example, and noticing that that set is pure, may not be as good as splitting the parent set into two nice large, relatively pure subsets, even if neither is pure.

As an example, consider the split in Figure 3-4. This is a two-class problem (• and ★). Examining the figure, the children sets certainly seem "purer" than the parent set. The parent set has 30 instances consisting of 16 dots and 14 stars, so:

$$\begin{aligned} entropy(parent) &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.53 \times -0.9 + 0.47 \times -1.1] \\ &\approx 0.99 \quad \text{(very impure)} \end{aligned}$$

The entropy of the *left* child is:

$$\begin{aligned} entropy(Balance < 50K) &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.92 \times (-0.12) + 0.08 \times (-3.7)] \\ &\approx 0.39 \end{aligned}$$

The entropy of the *right* child is:

$$\begin{aligned} entropy(Balance \geq 50K) &= -[p(\bullet) \times \log_2 p(\bullet) + p(\star) \times \log_2 p(\star)] \\ &\approx -[0.24 \times (-2.1) + 0.76 \times (-0.39)] \\ &\approx 0.79 \end{aligned}$$

Using Equation 3-2, the information gain of this split is:

$$
\begin{aligned}
IG &= entropy(parent) - [p(\text{Balance} < 50\text{K}) \times entropy(\text{Balance} < 50\text{K}) \\
&\qquad + p(\text{Balance} \geq 50\text{K}) \times entropy(\text{Balance} \geq 50\text{K})] \\
&\approx 0.99 - [0.43 \times 0.39 + 0.57 \times 0.79] \\
&\approx 0.37
\end{aligned}
$$

So this split reduces entropy substantially. In predictive modeling terms, the attribute provides a lot of information on the value of the target.

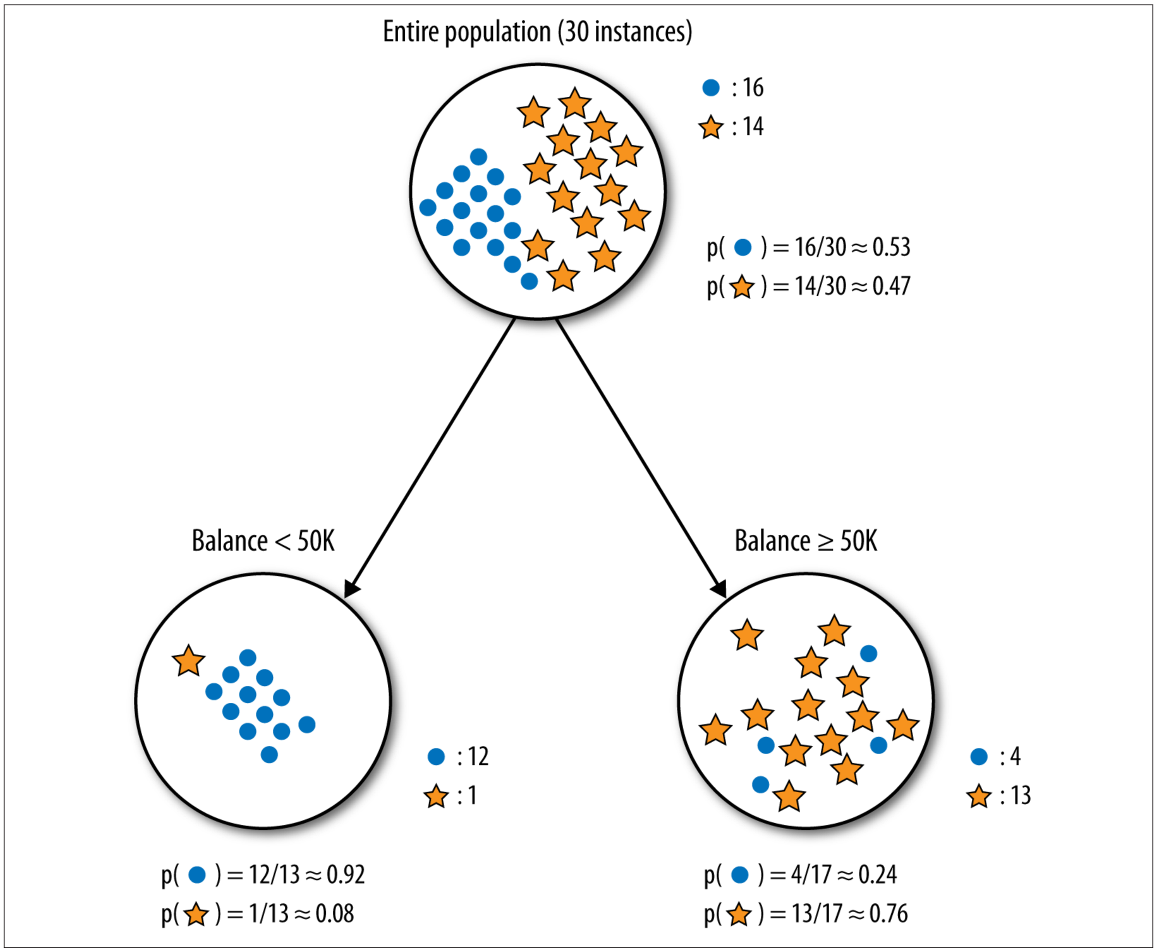

Figure 3-4. Splitting the "write-off" sample into two segments, based on splitting the Balance attribute (account balance) at 50K.

As a second example, consider another candidate split shown in Figure 3-5. This is the same parent set as in Figure 3-4, but instead we consider splitting on the attribute *Residence* with three values: OWN, RENT, and OTHER. Without showing the detailed calculations:

$$
\begin{aligned}
\mathit{entropy}(\mathit{parent}) &\approx 0.99\\
\mathit{entropy}(\text{Residence=OWN}) &\approx 0.54\\
\mathit{entropy}(\text{Residence=RENT}) &\approx 0.97\\
\mathit{entropy}(\text{Residence=OTHER}) &\approx 0.98\\
\mathit{IG} &\approx 0.13
\end{aligned}
$$

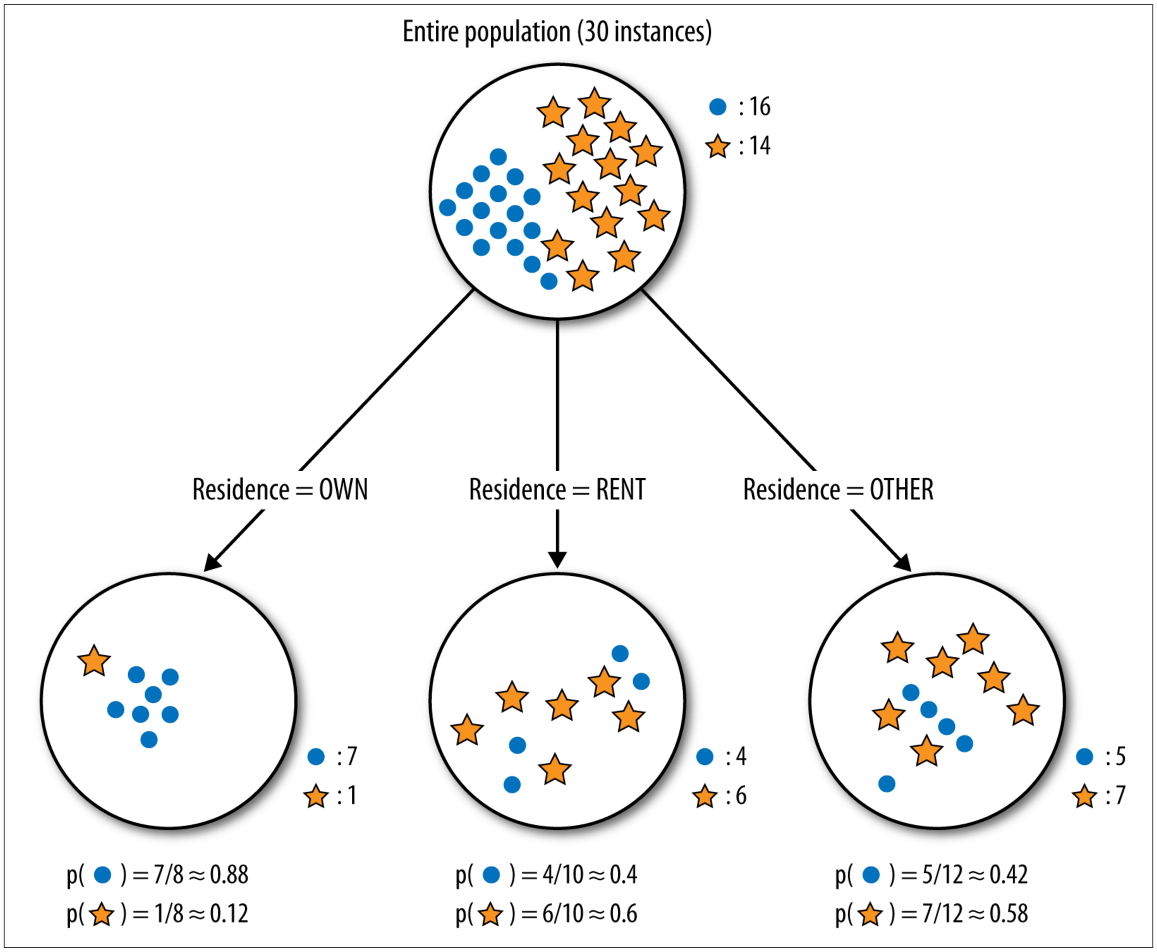

Figure 3-5. A classification tree split on the three-valued Residence attribute.

The Residence variable does have a positive information gain, but it is lower than that of Balance. Intuitively, this is because, while the one child Residence=OWN has considerably reduced entropy, the other values RENT and OTHER produce children that are no more pure than the parent. Thus, based on these data, the Residence variable is less informative than Balance.

Looking back at our concerns from above about creating supervised segmentation for classification problems, information gain addresses them all. It does not require absolute