Trees as Sets of Rules

Before moving on from the interpretation of classification trees, we should mention their interpretation as logical statements. Consider again the tree shown at the top of Figure 3-15. You classify a new unseen instance by starting at the root node and following the attribute tests downward until you reach a leaf node, which specifies the instance's predicted class. If we trace down a single path from the root node to a leaf, collecting the conditions as we go, we generate a rule. Each rule consists of the attribute tests along the path connected with AND. Starting at the root node and choosing the left branches of the tree, we get the rule:

```
IF (Balance < 50K) AND (Age < 50) THEN Class=Write-off
```

We can do this for every possible path to a leaf node. From this tree we get three more rules:

```
IF (Balance < 50K) AND (Age ≥ 50) THEN Class=No Write-off
IF (Balance ≥ 50K) AND (Age < 45) THEN Class=Write-off
IF (Balance ≥ 50K) AND (Age < 45) THEN Class=No Write-off
```

The classification tree is equivalent to this rule set. If these rules look repetitive, that's because they are: the tree gathers common rule prefixes together toward the top of the tree. Every classification tree can be expressed as a set of rules this way. Whether the tree or the rule set is more intelligible is a matter of opinion; in this simple example, both are fairly easy to understand. As the model becomes larger, some people will prefer the tree or the rule set.

Probability Estimation

In many decision-making problems, we would like a more informative prediction than just a classification. For example, in our churn-prediction problem, rather than simply predicting whether a person will leave the company within 90 days of contract expiration, we would much rather have an estimate of the probability that he will leave the company within that time. Such estimates can be used for many purposes. We will discuss some of these in detail in later chapters, but briefly: you might then rank prospects by their probability of leaving, and then allocate a limited incentive budget to the highest probability instances. Alternatively, you may want to allocate your incentive budget to the instances with the highest expected loss, for which you'll need (an estimate of) the probability of churn. Once you have such probability estimates you can use them in a more sophisticated decision-making process than these simple examples, as we'll describe in later chapters.

There is another, even more insidious problem with models that give simple classifications, rather than estimates of class membership probability. Consider the problem of estimating credit default. Under normal circumstances, for just about any segment of the population to which we would be considering giving credit, the probability of write-off will be very small—far less than 0.5. In this case, when we build a model to estimate the classification (write-off or not), we'd have to say that for each segment, the members are likely not to default—and they will all get the same classification (not write-off). For example, in a naively built tree model every leaf will be labeled "not write-off." This turns out to be a frustrating experience for new data miners: after all that work, the model really just says that no one is likely to default? This does *not* mean that the model is useless. It may be that the different segments indeed have very different probabilities of write-off, they just all are less than 0.5. If instead we use these probabilities for assigning credit, we may be able reduce our risk substantially.

So, in the context of supervised segmentation, we would like each segment (leaf of a tree model) to be assigned an estimate of the probability of membership in the different classes. Figure 3-15 more generally shows a "probability estimation tree" model for our simple write-off prediction example, giving not only a prediction of the class but also the estimate of the probability of membership in the class.[6]

Fortunately, the tree induction ideas we have discussed so far can easily produce probability estimation trees instead of simple classification trees.[7] Recall that the tree induction procedure subdivides the instance space into regions of class purity (low entropy). If we are satisfied to assign the same class probability to every member of the segment corresponding to a tree leaf, we can use instance counts at each leaf to compute a class probability estimate. For example, if a leaf contains n positive instances and m negative instances, the probability of any new instance being positive may be estimated as $n/(n+m)$. This is called a *frequency-based* estimate of class membership probability.

At this point you may spot a problem with estimating class membership probabilities this way: we may be overly optimistic about the probability of class membership for segments with very small numbers of instances. At the extreme, if a leaf happens to have only a single instance, should we be willing to say that there is a 100% probability that members of that segment will have the class that this one instance happens to have?

6. We often deal with binary classification problems, such as write-off or not, or churn or not. In these cases it is typical just to report the probability of membership in one chosen class $p(c)$, because the other is just $1 - p(c)$.

7. Often these are still called classification trees, even if the decision maker intends to use the probability estimates rather than the simple classifications.

This phenomenon is one example of a fundamental issue in data science ("overfitting"), to which we devote a chapter later in the book. For completeness, let's quickly discuss one easy way to address this problem of small samples for tree-based class probability estimation. Instead of simply computing the frequency, we would often use a "smoothed" version of the frequency-based estimate, known as the Laplace correction, the purpose of which is to moderate the influence of leaves with only a few instances. The equation for binary class probability estimation becomes:

$$p(c) = \frac{n + 1}{n + m + 2}$$

where n is the number of examples in the leaf belonging to class c, and m is the number of examples not belonging to class c.

Let's walk through an example with and without the Laplace correction. A leaf node with two positive instances and no negative instances would produce the same frequency-based estimate ($p = 1$) as a leaf node with 20 positive instances and no negatives. However, the first leaf node has much less evidence and may be extreme only due to there being so few instances. Its estimate should be tempered by this consideration. The Laplace equation smooths its estimate down to $p = 0.75$ to reflect this uncertainty; the Laplace correction has much less effect on the leaf with 20 instances ($p \approx 0.95$). As the number of instances increases, the Laplace equation converges to the frequency-based estimate. Figure 3-16 shows the effect of Laplace correction on several class ratios as the number of instances increases (2/3, 4/5, and 1/1). For each ratio the solid horizontal line shows the uncorrected (constant) estimate, while the corresponding dashed line shows the estimate with the Laplace correction applied. The uncorrected line is the asymptote of the Laplace correction as the number of instances goes to infinity.

Example: Addressing the Churn Problem with Tree Induction

Now that we have a basic data mining technique for predictive modeling, let's consider the churn problem again. How could we use tree induction to help solve it?

For this example, we have a historical data set of 20,000 customers. At the point of collecting the data, each customer either had stayed with the company or had left (churned). Each customer is described by the variables listed in Table 3-2.

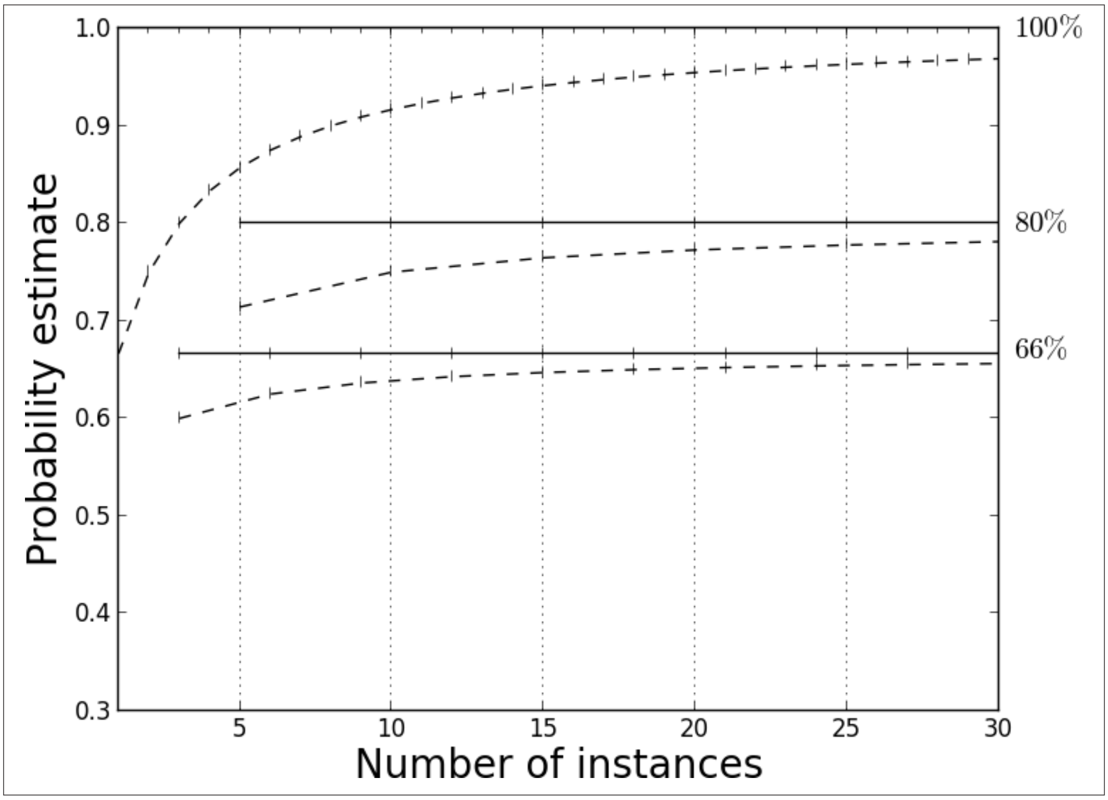

Figure 3-16. The effect of Laplace smoothing on probability estimation for several instance ratios.

Table 3-2. Attributes for the cellular phone churn-prediction problem

Variable	Explanation
COLLEGE	Is the customer college educated?
INCOME	Annual income
OVERAGE	Average overcharges per month
LEFTOVER	Average number of leftover minutes per month
HOUSE	Estimated value of dwelling (from census tract)
HANDSET_PRICE	Cost of phone
LONG_CALLS_PER_MONTH	Average number of long calls (15 mins or over) per month
AVERAGE_CALL_DURATION	Average duration of a call
REPORTED_SATISFACTION	Reported level of satisfaction
REPORTED_USAGE_LEVEL	Self-reported usage level
LEAVE *(Target variable)*	Did the customer stay or leave (churn)?