36-350: Data Mining

Handout 22
November 10, 2003

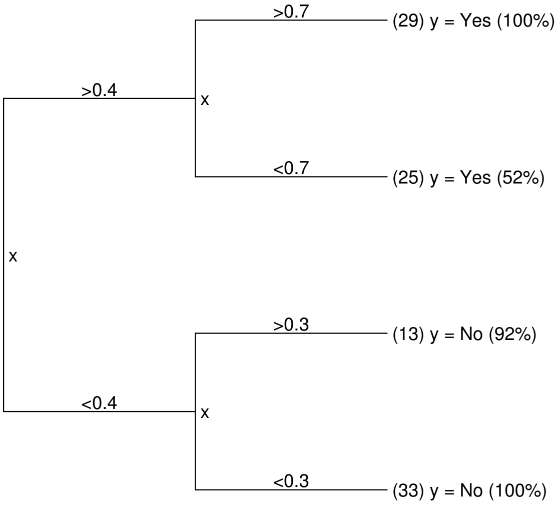
Classification trees

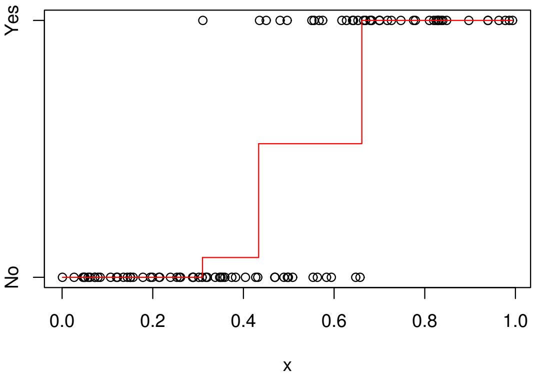
Consider predictive modeling where the response is categorical. These are called **classification problems**. A numeric response can always be turned into a categorical one (by quantization), but not vice versa. Thus classification is a more general problem than regression. We already exploited this fact when we used color-coding to visualize numeric responses (handout 12).

Previously we considered two methods for automatic classification: nearest-neighbor and nearest-prototype. It turns out that we can also apply more advanced modeling techniques like trees and linear regression.

A **classification tree** makes predictions by asking a series of questions, just like a regression tree (handout 14). Compared to nearest-neighbor, you get a very concise prediction rule which involves as few attributes as possible. It also subdivides the data into meaningful groups that you can further visualize.

How to build the tree—A classification tree asks the most informative question at each step. Let x be the outcome of the question, and c the response. Both are categorical. The information that x gives about c can be computed from the contingency table of (x, c) (see handout 4). A question is good if it changes the probability distribution of the response.

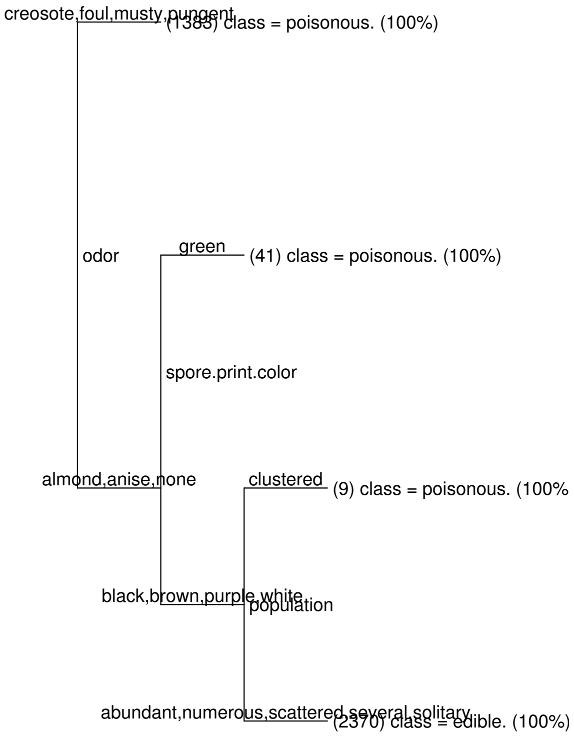




Mushroom classification—5416 varieties from a field guide, listing 22 categorical attributes as well as the "class" (edible or poisonous). The first row:

```
cap.shape cap.surface cap.color bruises odor gill.attachment gill.spacing
   convex     fibrous       red bruises none            free        close
gill.size gill.color stalk.shape stalk.root stalk.surface.above.ring
    broad     purple    tapering    bulbous                   smooth
stalk.surface.below.ring stalk.color.above.ring stalk.color.below.ring
                  smooth                   gray                   pink
veil.type veil.color ring.number ring.type spore.print.color population
  partial      white         one   pendant             brown    several
habitat   class
  woods edible.
```

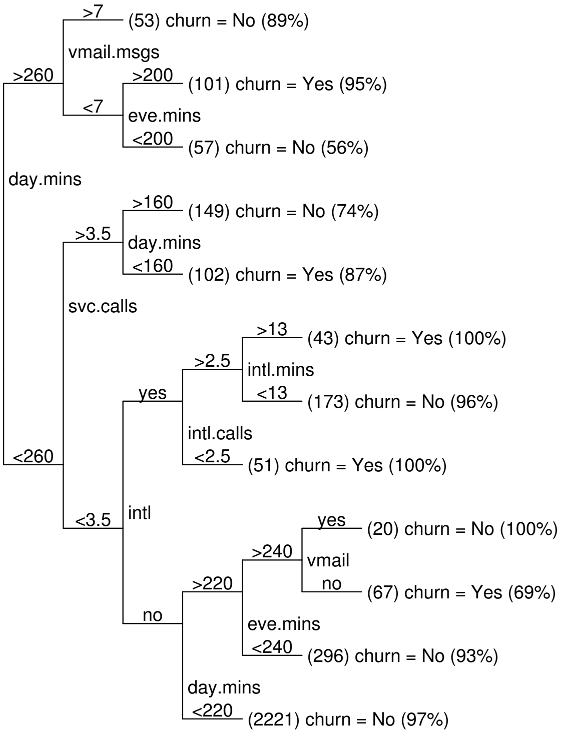
This is a lot of data, but the classification tree is simple:

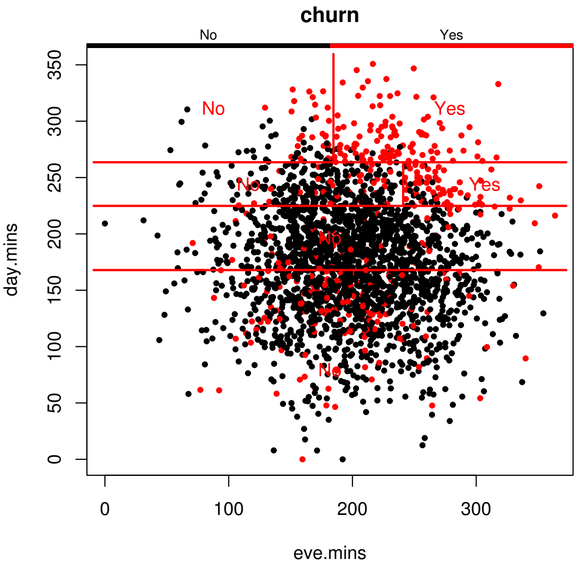

creosote,foul,musty,pungent
(1383) class = poisonous. (100%)
odor
green
(41) class = poisonous. (100%)
spore.print.color
almond,anise,none
clustered
(9) class = poisonous. (100%
black,brown,purple,white
population
abundant,numerous,scattered,several,solitary
(2370) class = edible. (100%)

The tree has achieved perfect separation between edible and poisonous. It has the form of an "OR" rule: if a mushroom has a foul odor OR it has green spores OR it appears in clusters, then it is poisonous.

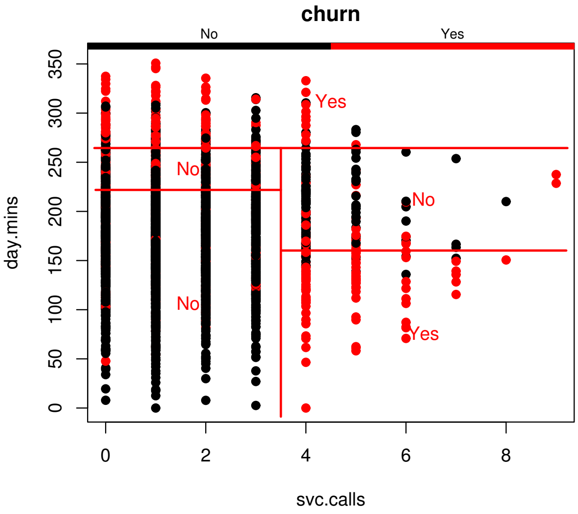
This data describes 3333 long-distance phone customers on 18 monthly statistics. The problem is to predict if a customer will switch to another company ("churn"). (The data was synthetically generated from an existing model.) Here is the first row:

```
churn intl vmail vmail.msgs day.mins day.calls day.charge
   No   no   yes         25    265.1       110      45.07
eve.mins eve.calls eve.charge night.mins night.calls
   197.4        99      16.78      244.7          91
night.charge intl.mins intl.calls intl.charge svc.calls
       11.01        10   1.732051         2.7         1
```

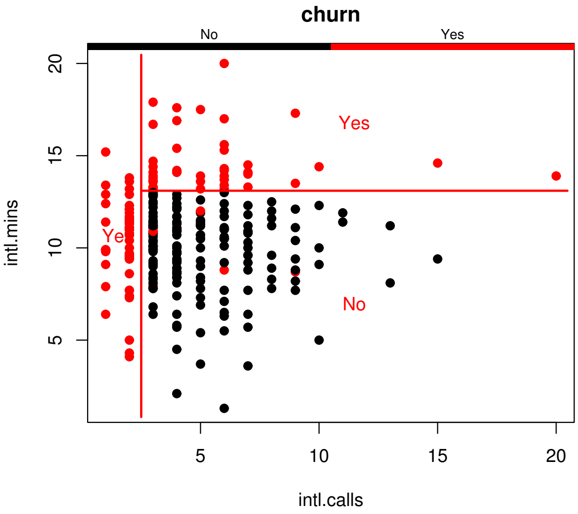




People with a large number of daytime and evening minutes tend to churn, but only if they have a small number of voicemail messages. This is bad since these are the company's best customers. The above plot is for customers with `vmail.msgs < 6.5`.



People who make many service calls yet have a small number of day minutes are likely to churn. These may be people who are just setting up their accounts. There is clearly a change in churn above 3 service calls. The customers with the most minutes do not make many service calls.



There is an interaction between the number of international minutes, international calls, and churn. People who make many short calls do not churn. Perhaps something in the pricing scheme?

Classification costs

A classification tree is designed to give accurate class probabilities—not just to predict the most likely class.

Class probabilities are important when different decisions have different costs. For example, if a customer will churn but we predict otherwise, the cost is 10. If the customer will not churn but we predict they will, the cost is probably lower, say 1. That means we should predict “churn” even if the probability of churn is less than 0.5.

The best decision minimizes **expected cost**. Let $Cost(C_k|C_j)$ be the cost of predicting C_k when the truth is C_j. For observation x, the classifier gives class probabilities $p(C_j|x)$. Then the expected cost of predicting C_k is:

$$Cost(C_k|x) = \sum_j Cost(C_k|C_j)p(C_j|x)$$

Suppose the cost matrix is biased against C_2:

	predict C_1	predict C_2
truth C_1	0	10
truth C_2	1	0

We run an observation through the tree and wind up with class probabilities $(0.4, 0.6)$. The most likely class is C_2, but it is not the most cost-effective decision. The expected cost of predicting C_1 is $0.4 * 0 + 0.6 * 1 = 0.6$, while the expected cost of predicting C_2 is $0.4 * 10 + 0.6 * 0 = 4$. The probability of C_2 must be 10 times higher than C_1 before C_2 is a cost-effective prediction.

References

[1] Mushroom data, from *The Audubon Society Field Guide to North American Mushrooms* (1981). `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mushroom/`

[2] Churn data. `http://www.sgi.com/tech/mlc/db/`