ID3 Decision Tree Algorithm

7 November 2012

Supervised learning

We will explore how a computer program can learn how to solve *classification problems* by examples. A classification problem is one where the attributes of some thing are presented, and a human expert (or computer program) must assign that thing to one of several *classes.* The attributes are sometimes called *independent variables,* and the classification as a *dependent variable.*

For example, when a doctor diagnoses a disease, that is a classification problem. The doctor considers various attributes of the patient (blood pressure, blood sugar, cholesterol, heart rate, pupil dilation, etc.) and then classifies the patient's problem (healthy, diabetes, heart attack, drug overdose, lung cancer, etc.)

Let's take another example. An auto insurance company considers a number of attributes about you and your car in order to place you in a risk category (high, medium, low) that then determines your insurance premium. Attributes about the driver might include age, sex, and number of accidents in the past 5 years. Attributes about your car might include its color, its make, whether it is a convertible, whether it has an alarm, etc. Let's suppose we have sample data about auto insurance risk:

#	RISK	AGE	SEX	ACCIDENTS	COLOR	ALARM
1	high	<25	m	0-2	blue	no
2	high	<25	f	0-2	blue	yes
3	high	25+	f	3+	red	no
4	high	<25	f	3+	blue	yes
5	high	<25	m	3+	red	no
6	high	25+	m	3+	green	no
7	high	<25	m	0-2	red	no
8	low	25+	m	0-2	blue	yes
9	medium	25+	m	0-2	red	no
10	medium	25+	f	0-2	red	yes
11	high	<25	f	3+	blue	no
12	high	<25	m	3+	red	yes
13	high	25+	f	3+	blue	yes
14	high	25+	m	3+	green	no
15	medium	25+	f	0-2	green	no
16	low	25+	m	0-2	green	yes
17	high	<25	f	0-2	green	no
18	high	<25	m	0-2	green	yes

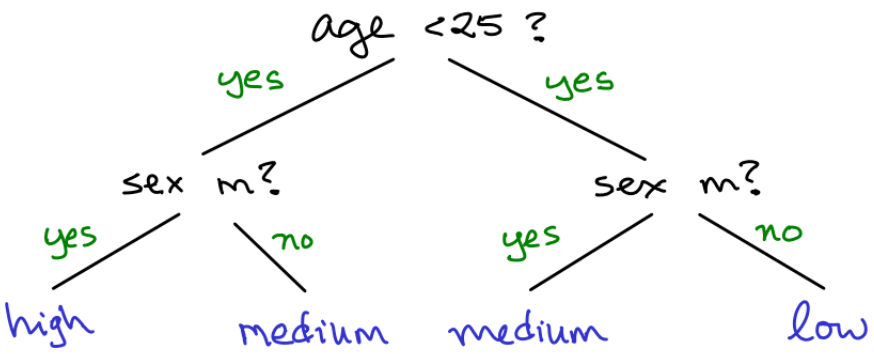
The job of a learner, whether it be human or computer, is to study these examples and determine the pattern that leads to the risk classification. Once you know the pattern, you can apply it to examples that you have never seen before:

```
 #     RISK     AGE  SEX  ACCIDENTS  COLOR  ALARM
19  (  ______   <25  f    3+         green  yes   )
20  (  ______   25+  f    3+         red    yes   )
21  (  ______   25+  f    0-2        blue   yes   )
22  (  ______   <25  m    3+         blue   no    )
23  (  ______   25+  m    0-2        blue   no    )
24  (  ______   <25  f    0-2        red    yes   )
25  (  ______   25+  m    3+         blue   yes   )
```

But how do we go about determining and representing the pattern? That's the subject of the next section.

Decision trees

A decision tree is just a tree structure with a single question (about the attributes) at each branch, and a classification at each leaf. We can easily represent decision trees using `if` statements in a programming language.



We interpret this by checking the condition at each branch, and then choosing the *left* path if the condition is true, or *right* if it is false.

The tree shown above is just an example, and is *not* the decision tree used to classify the data on the previous page. How many of the 18 examples (on the previous page) does the tree above *misclassify?* In other words, how many does it get wrong? _____

Information theory

The tree we illustrated above does not do a very good job classifying the sample data. But to quantify what we mean by doing a good job, we will look to a field called *information theory.* It defines a concept called *entropy,* which is essentially a measure of the *disorder* present in a system. For example, let's suppose we vote on whether to have pizza (P) or spaghetti (S) for dinner, and that 7 people vote P, while 1 person votes S. We are nearly all in agreement about what to eat, so that represents relatively little disorder.

Next, suppose that the votes are 4 and 4; in this case, we are very conflicted, and there is more disorder. If I give more than two choices, and everyone chooses something different, then there is even more disorder.

But how *much* more? To compute it, we need the base-2 logarithm function, which we will write as $\log_2$. If your calculator does not have $\log_2$, you can compute it using any

other logarithm function like this: $\log_2(x) = \log(x) \div \log(2)$

So, let's compute entropy in the first case, where there were 7 votes for P ($v_P = 7$) and 1 for S ($v_S = 1$) out of a total of 8 ($n = v_P + v_S = 8$). We substitute those proportions into this formula:

$$E = \frac{-v_P}{n} \times \log_2\left(\frac{v_P}{n}\right) + \frac{-v_S}{n} \times \log_2\left(\frac{v_S}{n}\right)$$

which simplifies to

$$\begin{aligned} E &= \frac{-7}{8} \times \log_2\left(\frac{7}{8}\right) + \frac{-1}{8} \times \log_2\left(\frac{1}{8}\right) \\ &= -.875 \times \log_2(.875) + -.125 \times \log_2(.125) \end{aligned}$$

Now you'll need a calculator or computer, but you'll find this becomes

$$\begin{aligned} E &= (-.875 \times -.1926) + (-.125 \times -3) \\ &= .1685 + .375 \\ &= .5435 \end{aligned}$$

Now, what about the second case, where the vote was split four-to-four? We expect the entropy to be higher.

$$\begin{aligned} E &= \frac{-4}{8} \times \log_2\left(\frac{4}{8}\right) + \frac{-4}{8} \times \log_2\left(\frac{4}{8}\right) \\ &= -.5 \times \log_2(.5) + -.5 \times \log_2(.5) \\ &= (-.5 \times -1) + (-.5 \times -1) \\ &= .5 + .5 \\ &= 1 \end{aligned}$$

This is the maximum entropy possible, **if** there are only two choices. If, however, everyone can vote for something different then the entropy will exceed 1.

So, turning back to our test data about auto insurance, let's compute the entropy of the classification as given. There are 13 high-risk customers, 3 medium-risk, and 2 low-risk, for a total of 18. To compute entropy with more than two classes, we just expand to as many terms as we need.

$$\begin{aligned} E &= \frac{-v_H}{n} \times \log_2\left(\frac{v_H}{n}\right) + \frac{-v_M}{n} \times \log_2\left(\frac{v_M}{n}\right) + \frac{-v_L}{n} \times \log_2\left(\frac{v_L}{n}\right) \\ &= \frac{-13}{18} \times \log_2\left(\frac{13}{18}\right) + \frac{-3}{18} \times \log_2\left(\frac{3}{18}\right) + \frac{-2}{18} \times \log_2\left(\frac{2}{18}\right) \\ &= (-.7222 \times -.4695) + (-.1667 \times -2.585) + (-.1111 \times -3.1699) \\ &= .3391 + .4308 + .3522 \\ &= 1.122 \end{aligned}$$

The entropy of a set where everyone is in agreement is zero. Unfortunately, this is tricky to compute from the formula because the logarithm of zero is undefined, and this can cause an error. But you can see it like this: suppose we have 8 votes for Pizza and 0 for Spaghetti:

$$\begin{aligned} E &= \frac{-v_P}{n} \times \log_2\left(\frac{v_P}{n}\right) + \frac{-v_S}{n} \times \log_2\left(\frac{v_S}{n}\right) \\ &= \frac{-8}{8} \times \log_2\left(\frac{8}{8}\right) + \frac{-0}{8} \times \log_2\left(\frac{0}{8}\right) \\ &= -1 \times \log_2(1) + 0 \times \log_2(0) \\ &= (-1 \times 0) + (0 \times ???) \end{aligned}$$

The log of zero will cause problems, but we don't really have to worry because anyway we're multiplying it by zero, which will produce zero.

Quinlan's ID3 tree builder

Okay, now we're ready to study the ID3 algorithm for automatically building a decision tree from a set of examples. It was invented by Ross Quinlan in 1979.

The idea is to enumerate all the possible tests that could apply to your data set, and then choose the one that would produce the lowest entropy. That is, we are looking for the test that makes the classification better organized, i.e., more in agreement. This should intuitively make sense, but let's apply it to the auto insurance example.

We already computed the entropy of the raw data: 1.122. Now let's consider two different ways to split it: by number of accidents, and by sex. We'll see which is the better choice. We just fill out a chart like the one below by counting how many of our examples that match the given criteria are classified as high/medium/low.

	accidents `0-2`?		**sex `m`?**	
	true	false	true	false
`high`	5	8	7	6
`medium`	3	0	1	2
`low`	2	0	2	0
Count	10	8	10	8
Entropy	1.485	0	1.157	.8113

Then, to see which is the better test we just need a *weighted average* of the entropies of the true/false columns. The average is weighted by the proportion of examples that appear in that column. For the test **accidents `0-2`**, we compute

$$E_{acc} = \frac{10}{18} \times 1.485 + \frac{8}{18} \times 0 = .825$$

and for `sex`, it is

$$E_{sex} = \frac{10}{18} \times 1.157 + \frac{8}{18} \times .8113 = 1.003$$

So, both of these reduce the entropy (compared to 1.122), but the number of accidents reduces it lower, so that is the better test.

Using the same technique, compute the weighted average entropy for these tests:

- **age <25?** _____
- **alarm yes?** _____

Out of all four of these tests, which reduces the entropy the most? _____

After choosing the best test for the root of the decision tree, we repeat the process on each branch, using the data as filtered by that test. For example, let's suppose that **color blue?** proves to be the best test – it isn't, but I don't want to give away the answer to the question above!

So, we put **color blue?** at the root of our decision tree, and then split the data into two disjoint subsets: the blue cars and the not-blue cars. If the entropy of one of these sets is zero (which indicates that they all 'agree' on the classification), then we can place a leaf there, labeled with that class. Otherwise, we repeat the algorithm on each subset.

Suppose we have already classified customers under the age of 25 *or* with 3 or more accidents, so we're left with just the 5 customers who are 25+ years old *and* with 0–2 accidents – they are numbers 8, 9, 10, 15, and 16. Considering just these customers, compute the entropy of each of the following tests.

- **sex m?** _____
- **color green?** _____
- **alarm yes?** _____

Which of those tests is best at this point? _____