

Decision Trees

CS 221

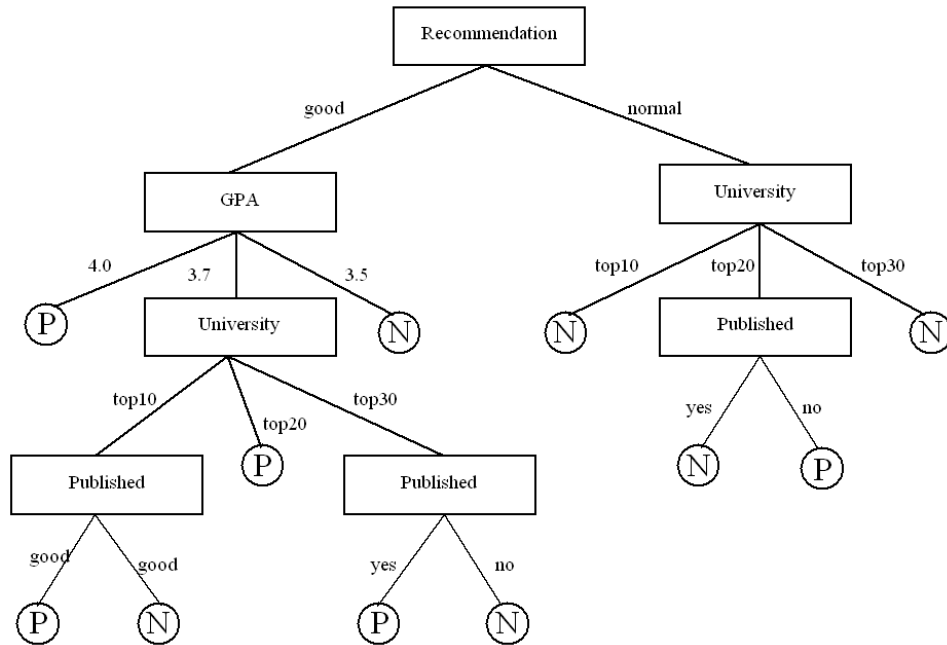
Section 5

October 23, 2009

Today we will work through an example problem of creating a decision tree. Time permitting, we will also work an example problem about AdaBoost.

1 Creating a decision tree

Consider the criteria for accepting candidates to the Ph.D. program at the mythical University of St. Nordaf. Each candidate is evaluated according to four attributes: The grade point average (GPA), the quality of the undergraduate university attended, the publication record, and the strength of the recommendation letters. To simplify our example, let us discretize and limit the possible value of each attribute: Possible GPA scores are 4.0, 3.7, and 3.5; universities are categorized as top10, top20, and top30 (by top20 we mean places 11-20, and by top 30 we mean 21-30); publication record is a binary attribute - either the applicant has published previously or not; and recommendation letters are similarly binary, they are either good or normal. Finally, the candidates are classified into two classes: accepted, or P (for 'positive'), and rejected, or N (for 'negative'). Following is an example of one possible decision tree determining acceptance.



Applicant Pat doesn't know this decision tree, but she does have the following data regarding twelve of last year's applicants:

No.	Attributes				Class
	GPA	University	Published	Recommendation	
1	4.0	top10	yes	good	P
2	4.0	top10	no	good	P
3	4.0	top20	no	normal	P
4	3.7	top10	yes	good	P
5	3.7	top20	no	good	P
6	3.7	top30	yes	good	P
7	3.7	top30	no	good	N
8	3.7	top10	no	good	N
9	3.5	top20	yes	normal	N
10	3.5	top10	no	normal	N
11	3.5	top30	yes	normal	N
12	3.5	top30	no	good	N

1. Verify that the tree given above correctly categorizes these examples.

Answer: Yes the tree provided does classify the examples correctly. We can verify this by tracing a path from the root to a leaf for every example and confirming that the leaf value matches the example value.

2. Pat uses the decision tree algorithm shown in class (with the information gain computations for splitting variables) to induce the decision tree em-

ployed by St. Nordaf's officials. What tree will the algorithm come up with? Show **all** the computations involved, in addition to the tree itself.

Answer: Recall that we choose to split on the feature which gives the greatest increase in the log likelihood. This corresponded to choosing the feature that gave the greatest reduction in entropy, also known as the highest information gain.

Given a leaf l , its contribution to the log likelihood is $-m_l \mathcal{H}(p_l)$, where m_l is the number of samples associated with l , and p_l is the number associated with the leaf (we found that the optimal value for p_l is $\frac{m_{l+}}{m_{l-}}$). The entropy function is defined as

$$\mathcal{H}(p) = -p \log_2 p - (1-p) \log_2 (1-p)$$

Suppose we split a leaf l in a tree T into $l_1 \dots l_k$ in tree T' . Then l 's contribution to T 's log likelihood, $-m_l \mathcal{H}(p_l)$, is now replaced by l_1, \dots, l_k 's contribution to T' 's log likelihood, $-\sum_{i=1}^k m_{l_i} \mathcal{H}(p_{l_i})$. Hence the increase in log likelihood / entropy reduction / information gain is

$$-\sum_{i=1}^k m_{l_i} \mathcal{H}(p_{l_i}) - (-m_l \mathcal{H}(p_l))$$

or

$$m_l \mathcal{H}(p_l) - \sum_{i=1}^k m_{l_i} \mathcal{H}(p_{l_i})$$

We will use this equation throughout our decision tree computation.

Initially, we have a single root leaf that contains all of the 12 samples, of which 6 are positive and 6 are negative:

$$\begin{aligned} m_l \mathcal{H}(p_l) &= 12 \mathcal{H}\left(\frac{1}{2}\right) \\ &= 12 \end{aligned}$$

The information gains for splitting on each of the features is therefore as

follows:

$$\begin{aligned}\text{Gain(GPA)} &= 12 - \left[3\mathcal{H}\left(\frac{3}{3}\right) + 5\mathcal{H}\left(\frac{3}{5}\right) + 4\mathcal{H}\left(\frac{0}{4}\right) \right] \\ &= 12 - [3(0) + 5(0.97095) + 4(0)] \\ &= 7.145\end{aligned}$$

$$\begin{aligned}\text{Gain(University)} &= 12 - \left[5\mathcal{H}\left(\frac{3}{5}\right) + 3\mathcal{H}\left(\frac{2}{3}\right) + 4\mathcal{H}\left(\frac{1}{4}\right) \right] \\ &= 12 - [5(0.97095) + 3(0.91830) + 4(0.81128)] \\ &= 1.145\end{aligned}$$

$$\begin{aligned}\text{Gain(Published)} &= 12 - \left[5\mathcal{H}\left(\frac{3}{5}\right) + 7\mathcal{H}\left(\frac{3}{7}\right) \right] \\ &= 12 - [5(0.97095) + 7(0.98523)] \\ &= 0.249\end{aligned}$$

$$\begin{aligned}\text{Gain(Recommendation)} &= 12 - \left[4\mathcal{H}\left(\frac{1}{4}\right) + 8\mathcal{H}\left(\frac{5}{8}\right) \right] \\ &= 12 - [4(0.81128) + 8(0.95443)] \\ &= 1.119\end{aligned}$$

Since GPA has the highest information gain, we split first on GPA . This divides our evidence E into three classes: $GPA = 4.0$, $GPA = 3.7$, and $GPA = 3.5$. For $E_{4.0}$, all instances are positive, so that branch is done. For $E_{3.5}$, all instances are negative, so again we're done. For $E_{3.7}$ there are some positive and some negative instances, so we must repeat this procedure again.

We again first compute the current contribution of this leaf to the log likelihood. The leaf contains 5 samples, of which 3 are positive and 2 are negative:

$$\begin{aligned}m_l \mathcal{H}(p_l) &= 5\mathcal{H}\left(\frac{3}{5}\right) \\ &= 4.854\end{aligned}$$

Now we can compute the information gain for splitting on each of the three

remaining attributes:

$$\begin{aligned}\text{Gain(University)} &= 4.854 - \left[2\mathcal{H}\left(\frac{1}{2}\right) + 1\mathcal{H}\left(\frac{1}{1}\right) + 2\mathcal{H}\left(\frac{1}{2}\right) \right] \\ &= 4.854 - [2(1) + 1(0) + 2(1)] \\ &= 0.854\end{aligned}$$

$$\begin{aligned}\text{Gain(Published)} &= 4.854 - \left[2\mathcal{H}\left(\frac{2}{2}\right) + 3\mathcal{H}\left(\frac{1}{3}\right) \right] \\ &= 4.854 - [2(0) + 3(0.91830)] \\ &= 2.099\end{aligned}$$

$$\begin{aligned}\text{Gain(Recommendation)} &= 4.854 - \left[5\mathcal{H}\left(\frac{3}{5}\right) \right] \\ &= 4.854 - [5(0.97095)] \\ &= 0\end{aligned}$$

Of these, *Published* has the highest gain, so we choose to split on it next. This divides these 5 cases into two categories of evidence, E_{yes} (yes, the student did publish) and E_{no} . For E_{yes} , all cases are P , so we are done.

We still need to split on E_{no} , a set containing 3 samples (1 positive and 2 negative). First we compute the amount of information in this subtree:

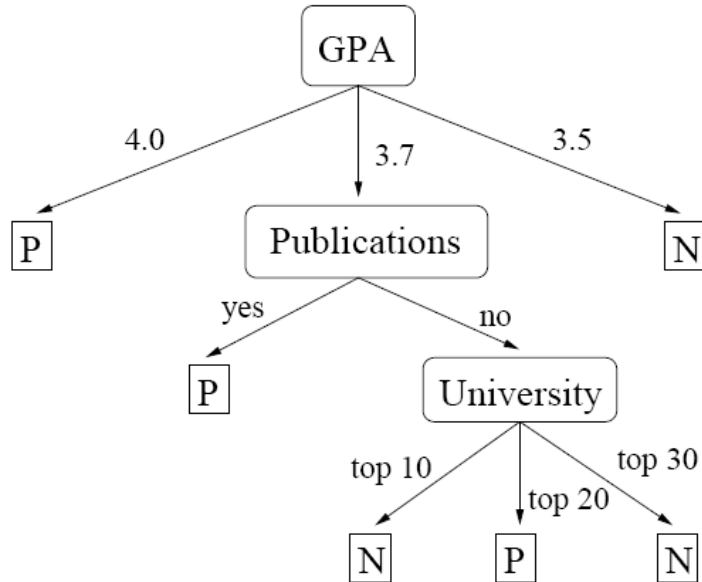
$$\begin{aligned}m_i\mathcal{H}(p_i) &= 3\mathcal{H}\left(\frac{1}{3}\right) \\ &= 2.755\end{aligned}$$

And then the information gain of the remaining attributes:

$$\begin{aligned}\text{Gain(University)} &= 2.755 - \left[1\mathcal{H}\left(\frac{0}{1}\right) + 1\mathcal{H}\left(\frac{1}{1}\right) + 1\mathcal{H}\left(\frac{0}{1}\right) \right] \\ &= 2.755 - [1(0) + 1(0) + 1(0)] \\ &= 2.755\end{aligned}$$

$$\begin{aligned}\text{Gain(Recommendation)} &= 2.755 - \left[3\mathcal{H}\left(\frac{1}{3}\right) \right] \\ &= 2.755 - [3(0.91830)] \\ &= 0\end{aligned}$$

So we split on *University*. It breaks the evidence down into E_{top10} , E_{top20} , and E_{top30} , each of which is completely classified, so we're done:



3. Is the tree you got in question 2 equivalent to the tree given above (i.e., do the two trees classify every application in the same way)? If the answer is yes, explain whether or not this is a coincidence. If the answer is no, give an example of a data case that will be classified differently by the two trees.

Answer: No, this tree is not equivalent to the one used by St. Nordaf's officials. for example, the case { GPA = 4.0, University = top10, Published = yes, Recommendation = normal } is classified *N* by St. Nordaf's, and classified *P* by this tree.

2 Some AdaBoost properties

In the AdaBoost algorithm, we train a classifier, h_b , on training examples sampled from the distribution D_b . This classifier is a "weak learner" that should achieve an error rate of less than 50%. We then build a new distribution D_{b+1} . Show that this distribution assigns 50% of the probability mass to the examples that the previous classifier labeled incorrectly.

Specifically, show:

$$\sum_{i=1}^m D_{b+1}(i) \mathbf{1} \{ h_b(x^{(i)}) \neq y^{(i)} \} = \frac{1}{2}$$

Answer:

First, recall that ϵ_b is the error rate of the classifier h_b trained with examples sampled using the distribution D_b :

$$\epsilon_b = \sum_{i: h_b(x^{(i)}) \neq y^{(i)}} D_b(i)$$

Also, recall that AdaBoost defines:

$$\alpha_b = \frac{1}{2} \log \frac{1 - \epsilon_b}{\epsilon_b}.$$

Finally, the AdaBoost update rule for computing $D_{b+1}(i)$ is:

$$\begin{aligned} D_{b+1}(i) &= \frac{D_b(i)}{z_b} \begin{cases} e^{-\alpha_b} & \text{if } h_b(x^{(i)}) = y^{(i)} \\ e^{\alpha_b} & \text{if } h_b(x^{(i)}) \neq y^{(i)} \end{cases} \\ &= \frac{D_b(i)}{z_b} \begin{cases} \left(\frac{\epsilon_b}{1 - \epsilon_b}\right)^{1/2} & \text{if } h_b(x^{(i)}) = y^{(i)} \\ \left(\frac{1 - \epsilon_b}{\epsilon_b}\right)^{1/2} & \text{if } h_b(x^{(i)}) \neq y^{(i)} \end{cases} \end{aligned}$$

The z_b term is a normalizing constant so that D_{b+1} will be a probability distribution summing to 1. To find this, we just compute the sum of all of the unnormalized terms. If we separate the examples classified correctly from the examples classified incorrectly by h_b , we get:

$$\begin{aligned} z_b &= \sum_{i: h_b(x^{(i)}) = y^{(i)}} D_b(i) \left(\frac{\epsilon_b}{1 - \epsilon_b}\right)^{1/2} + \sum_{i: h_b(x^{(i)}) \neq y^{(i)}} D_b(i) \left(\frac{1 - \epsilon_b}{\epsilon_b}\right)^{1/2} \\ &= \left(\frac{\epsilon_b}{1 - \epsilon_b}\right)^{1/2} \sum_{i: h_b(x^{(i)}) = y^{(i)}} D_b(i) + \left(\frac{1 - \epsilon_b}{\epsilon_b}\right)^{1/2} \sum_{i: h_b(x^{(i)}) \neq y^{(i)}} D_b(i) \\ &= \left(\frac{\epsilon_b}{1 - \epsilon_b}\right)^{1/2} (1 - \epsilon_b) + \left(\frac{1 - \epsilon_b}{\epsilon_b}\right)^{1/2} \epsilon_b \\ &= (\epsilon_b(1 - \epsilon_b))^{1/2} + ((1 - \epsilon_b)\epsilon_b)^{1/2} \\ &= 2((1 - \epsilon_b)\epsilon_b)^{1/2} \end{aligned}$$

Using this value of z_b , we can simplify D_{b+1} :

$$D_{b+1}(i) = D_b(i) \begin{cases} \frac{1}{2(1 - \epsilon_b)} & \text{if } h_b(x^{(i)}) = y^{(i)} \\ \frac{1}{2\epsilon_b} & \text{if } h_b(x^{(i)}) \neq y^{(i)} \end{cases}$$

Note that if ϵ_b is close to 0.5 (that is, h_b only slightly better than the required %50 for a weak learner), then the next distribution, D_{b+1} , is almost the same as the previous one. Conversely, if ϵ_b is very small (that is, h_b made very few mistakes), then the misclassified examples are given much larger weight.

From this final equation, we get our result:

$$\begin{aligned}\sum_{i=1}^m D_{b+1}(i) \mathbf{1} \{h_b(x^{(i)}) \neq y^{(i)}\} &= \sum_{h_b(x^{(i)}) \neq y^{(i)}} D_{b+1}(i) \\ &= \frac{1}{2\epsilon_b} \sum_{h_b(x^{(i)}) \neq y^{(i)}} D_b(i) \\ &= \frac{1}{2\epsilon_b} \epsilon_b \\ &= \frac{1}{2}\end{aligned}$$