CS 188 Section 10: Decision Trees

1 Decision Trees

In the recursive construction of decision trees, it sometimes happens that a mixed set of positive and negative examples remains at a leaf node, even after all the attributes have been used. Suppose that we have p positive examples and n negative examples.

function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent_examples*) **returns** a tree

if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*)
else if all *examples* have the same classification **then return** the classification
else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*)
else
  $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
  tree $\leftarrow$ a new decision tree with root test A
  for each value v_k of A **do**
    $exs \leftarrow \{e : e \in examples \text{ and } e.A = v_k\}$
    subtree $\leftarrow$ DECISION-TREE-LEARNING(exs, $attributes - A$, *examples*)
    add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
  return *tree*

1. Show that the solution used by the decision tree learning algorithm, which picks the majority classification, minimizes the absolute error over the set of examples at the leaf.
   If α is returned, the absolute error is:

$$\begin{aligned} E &= p(1-\alpha) + n\alpha = \alpha(n-p) + p \\ &= n \text{ when } \alpha = 1. \\ &= p \text{ when } \alpha = 0. \end{aligned}$$

   Thus, the error is minimized by setting $\alpha = 1$ if $p > n$ and 0 otherwise.

2. Show that the class probability, $p/(p+n)$ minimizes the sum of squared errors.

   The sum of squared error is:
   $E = p(1-\alpha)^2 + n\alpha^2$

   Its derivative is:
   $dE/d\alpha = 2\alpha n - 2p(1-\alpha) = 2\alpha(p+n) - 2p.$

   The derivative has a zero at $\alpha = p/(p+n)$.
   Now, we can prove that this is a minimum by evaluating the second derivative, $2(p+n)$, to see that the point of interest is in fact a minimum of E, the sum of squared errors equation.

2 Zero Gain

Suppose that an attribute splits the set of examples E into subsets E_k and that each subset has p_k positive examples and n_k negative examples. Show that the attribute has zero gain when $p_k/(p_k + n_k)$ is the same for all k.

The gain is defined to be:
$B(p/(p+n)) - \Sigma_{k=1}^{d}(p_k + n_k)/(p+n)B(p_k/(p_k + n_k))$.
Since $p = \Sigma p_k$ and $n = \Sigma n_k$, if $p_k/(p_k + n_k)$ is the same for all k, we must have $p_k/(p_k + n_k) = p/(p+n)$ for all k.
Plugging it back into our gain equation, we get:
$B(p/(p+n)) - \Sigma_{k=1}^{d}(p_k + n_k)/(p+n)B(p_k/(p_k + n_k))$.
$= B(p/(p+n)) - (p+n)/(p+n)B(p/(p+n))$.
$= 0$

3 Information Gain

You are a geek who hates sports. Trying to look cool at a party, you join a discussion that you belive to be about football and basketball. You gather information about the two main subjects of discussion, but still cannot figure out what sports they play.

Sport	Position	Name	Height	Weight	Age	College
?	Guard	Charlie Ward	6'02"	185	41	Florida State
?	Defensive End	Julius Peppers	6'07"	283	32	North Carolina

Fortunately, you have brought your CS 188 notes along, and will build some classifiers to determine which sport is being discussed.

You come across a pamphlet from the Atlantic Coast Conference Basketball Hall of Fame, as well as an Oakland Raiders team roster, and create the following table:

Sport	Position	Name	Height	Weight	Age	College
Basketball	Guard	Michael Jordan	6'06"	195	49	North Carolina
Basketball	Guard	Vince Carter	6'06"	215	35	North Carolina
Basketball	Guard	Muggsy Bogues	5'03"	135	47	Wake Forest
Basketball	Center	Tim Duncan	6'11"	260	35	Oklahoma
Football	Center	Vince Carter	6'02"	295	29	Oklahoma
Football	Kicker	Tim Duncan	6'00"	215	33	Oklahoma
Football	Kicker	Sebastian Janikowski	6'02"	250	33	Florida State
Football	Guard	Langston Walker	6'08"	345	33	California

Central to decision trees is the concept of "splitting" on a variable.

1. To review the concept of "information gain", calculate it for a split on the *Sport* variable.

   Since the variable that we want to predict is *Sport*, we want to be caluclating the entropy with respect to the variable *Sport*.

   (a) i. Distribution before: 8 examples with (1/2, 1/2). (here the first number in the tuple is P(basketball), and the second number is P(football)).

   ii. Entropy before: $\frac{8}{8}\left(\frac{log(2)}{2}+\frac{log(2)}{2}\right)$

   (b) i. Distribution after: 4 examples with (1, 0), 4 examples with (0, 1)

   ii. Entropy after: $\frac{4}{8}\left(\frac{log(1)}{1}\right)+\frac{4}{8}\left(\frac{log(1)}{1}\right)=0$

   So, the information gain is $(1-0)=1$, which is the greatest possible.

2. Of course, in our situation this would not make sense, as *Sport* is the very variable we lack at test time. Now calculate the information gain for the decision "stumps" (one-split trees) created by first splitting on *Position*, *Name*, and *College*. Do any of these perfectly classify the training data? Does it make sense to use *Name* as a variable? Why or why not?

   Note that here we will be splitting on different variables but still need to look at the entropy of the distribution of the variable we need to predict which is *sport*. So, the before case remains same as before.

   (a) **Position**

   i. Distribution after: 4 examples with (3/4, 1/4), 2 examples with (1/2, 1/2), 2 examples with (0, 1).

   ii. Entropy after: $\frac{4}{8}\left(\frac{log(4/3)}{4/3}+\frac{log(4)}{4}\right)+\frac{2}{8}\left(\frac{log(2)}{2}+\frac{log(2)}{2}\right)+\frac{2}{8}\left(\frac{log(1)}{1}\right)=0.66$

   (b) **Name**

i. Distribution after: 1 examples with (1, 0), 2 examples with (1/2, 1/2), 1 examples with (0, 1), 2 examples with (1/2, 1/2), 1 example with (0,1), 1 example with (0,1).

ii. Entropy after: 0.5

(c) **College**

i. Distribution after: 2 examples with (1, 0), 1 examples with (1, 0), 3 examples with (1/3, 2/3), 1 examples with (0, 1), 1 example with (0,1).

ii. Entropy after: 0.34

Note that none of these variables completely classifies the data.

Regarding using the **Name** as a feature to use in classifying data: since we expect people's names to be unique, using them as a feature in learning is akin to using the unique ID of each data point. That is to say, it's quite a bad idea—you will overfit to the training data.

3. Decision trees can represent any function of discrete attribute variables. How can we *best* cast continuous variables (*Height*, *Weight*, and *Age*) into discrete variables?

   Use an inequality relation, Attribute $> a$, where a is a split point chosen to give the highest information gain. E.g., an initial split on Age > 34 will perfectly classify the training data.

4 Decision Graphs

A decision graph is a generalization of a decision tree that allows nodes (i.e., attributes used for splits) to have multiple parents, rather than just a single parent. The resulting graph must still be acyclic. Now, consider the XOR function of three binary input attributes, which produces the value 1 if and only if an odd number of the three attributes has value 1.

1. Draw a minimal-sized decision tree for the three-input XOR function.

2. Draw a minimal-sized decision graph for the three-input XOR function.