ICS 171, Rina Dechter

Fall 2006

Homework 7, due Tuesday November 21st

Consider the following table of observations:

No.	Outlook	Temperature	Humidity	Windy	Play Golf?
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	Y
4	rain	mild	high	false	Y
5	rain	cool	normal	false	Y
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	Y
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	Y
10	rain	mild	normal	false	Y
11	sunny	mild	normal	true	Y
12	overcast	mild	high	true	Y
13	overcast	hot	normal	false	Y
14	rain	mild	high	true	N

1. From the classified examples in the above table, construct two decision trees (by hand) for the classification "Play Golf."
   A. For the first tree, use Temperature as the root node. (This is a really bad choice.) Continue, using your best judgement for selecting other attributes. Remember that different attributes can be used in different branches on a given level of the tree.
   B. For the second tree, follow the Decision-Tree-Learning algorithm described on page 658. As your Choose-Attribute function, choose the attribute with the highest information gain. Work out the computations of information gain by hand (you can of course use a calculator to compute $\log_2$), and turn in enough of your intermediate steps to convince the readers you did the work yourself.

   Double check that each tree correctly classifies all examples.

2. Do problem 18.4.
3. The formula for *Remainder* near the top of page 660 makes the assumption that the data falls into two classes, "positive" and "negative." However, this is not always the case. For instance, we might want to categorize credit applications as

"reject," "accept," and "borderline - refer to manager." Rewrite the formula for *Remainder* to allow for any number of classes. Use *m* to indicate the number of classes, and define any other variables you use.

4. In this question we will focus on perceptrons with a threshold activation function. Thus instead of (20.12) on page 742, we'll use

$$W_j = W_j + (alpha * Err * x_j)$$

where *Err* is defined as the correct output (either 0 or 1) minus the perceptron's actual output (also 0 or 1).

Consider a three input, threshold = 0 function perceptron, beginning with all weights (including W_0, the threshold) set to 0.1. The learning rate is 0.1. The values of inputs 1, 2, and 3 are real numbers between -1 and 1. The function the perceptron is learning is: "1 if input-1 plus input-2 is greater than input-3; 0 otherwise." The three example sets of values for input-1, input-2, and input-3 are (0.3, 0.1, 0.8), (0.1, 0.5, 0.2), and (0.4, 0.6, 0.5). Show the values of the weights after each input and the associated learning. Show your intermediate steps. Do the computations by hand, don't use a spreadsheet. Just do computations for one epoch.