15-381 Artificial Intelligence: Representation and Problem Solving
Homework 5

Out: 11/13/08
Due: 12/02/08

Instructions

This assignment is due on **Tuesday**, December 2nd, 2008. The written portion must be turned in at the beginning of class at noon on December 2nd. Type or write legibly; illegible submissions will not receive credit. Write your name and Andrew ID clearly at the top of the assignment.

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas in the class in order to help each other answer homework questions. You are also welcome to give each other examples that are not in the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other answers;
- not copy answers;
- not allow your answers to be copied.

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we ask that you specifically record on the assignment the names of the people you were in discussion with (or "none" if you did not talk to anyone else). This will help resolve the situation where a mistake in general discussion led to a replicated error among multiple solutions. This policy has been established in order to be fair to everyone in the class. We have a grading policy of watching for cheating and we will follow up if it is detected.

Refer to the web page for policies regarding collaboration, due dates, and extensions.

1 [20 pts] Naive Bayes

Consider the following set of training examples for the unknown target function $< X1, X2 > \rightarrow Y$. Each row indicates the values observed, and how many times that set of values was observed. For example, (+, T, T) was observed 3 times, while (-, T, T) was never observed.

Y	X_1	X_2	Count
+	T	T	3
+	T	F	4
+	F	T	4
+	F	F	1
-	T	T	0
-	T	F	1
-	F	T	3
-	F	F	5

Construct a Naive Bayes classifier for this data using the method described in the class and state the values of the parameters learnt.

2 [15 pts] Linear Regression

Consider a series of observations of the form $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$. Given these observations("training data"), our aim is to find a function f of the form $f(x) = wx$ such that $f(x)$ is a good estimate of y for this dataset. We saw this problem in class; this is just the problem of linear regression.

In class, we found the w that minimized the square error over the data($\sum_i^n (y_i - wx_i)^2$) and considered that to be a good estimate. We will now see why that might be considered a good estimate.

Suppose the likelihood of the observation y_i is a Gaussian distribution with mean wx_i and variance σ, i.e. $P(Y = y_i|w, x_i) = 1/(2\pi\sqrt{\sigma})e^{-(y_i - wx_i)^2/(2\sigma^2)}$

1. Derive the expression for the conditional likelihood of the data assuming that each observation was independently generated. ie. write the expression for $P(y_1, y_2, \ldots, y_n|w, x_1, x_2, \ldots x_n) = \prod_i^n P(y_i|w, x_i)$

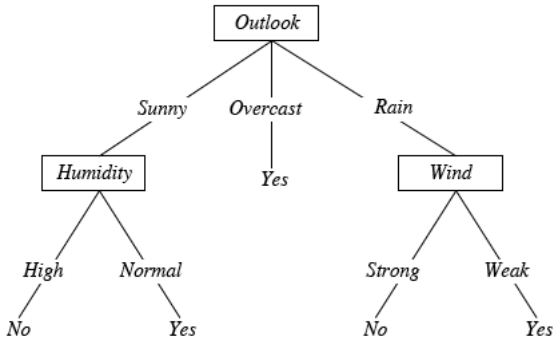
2. Now, using the expression for the conditional likelihood of the data, write down the expression for the conditional *log*-likelihood, $\log(P(y_1, y_2, \ldots, y_n|w, x_1, x_2, \ldots x_n))$

3. Show that computing the Maximum Likelihood Estimate of w is equivalent to finding the w that minimizes the square error. Recall from earlier classes, that a Maximum Likelihood Estimate of a parameter(in this case w) is the value of the parameter that maximizes the likelihood of the data.

3 [15 pts] Learning Decision Trees

Consider the following training data and the following decision tree learned from this data using the algorithm described in class. The decision tree predicts the value of the boolean attribute PlayTennis using the values of the rest of the attributes.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



1. Show that the choice of the Wind attribute at the second level of the tree is correct, by showing that its information gain is superior to the alternative choices.
2. Add one new example to the above data set, so that the learned tree will contain additional nodes.
3. Is it possible to add new examples to the above training set, which are consistent with the above tree, to produce a larger training set such that the algorithm will now learn a tree whose root node is not Outlook?. (We say an example is consistent with the above tree if the tree classifies the example correctly). Justify your answer by explaining informally why this is impossible, or explaining the new data you would add.

4 [15 pts] Decision Trees to Rules

Any decision tree can be re-expressed as a set of rules, with one rule for each leaf. The preconditions of the rule correspond to the sequence of attribute tests along the path from the tree root to the leaf. For example, the leftmost leaf in the tree in Q. 3 corresponds to the rule IF (Outlook = sunny AND Humidity=High) THEN PlayTennis=No

1. Write the rules for the remaining leaves. Note this set of rules produces classifications that are identical to the above tree, over the training data and over any other possible instance.
2. It is possible to translate any tree into a set of rules that represents an equivalent classifier. Is it possible to translate any set of rules into an equivalent tree? Explain how or give a counterexample. You may assume that all your attributes are boolean.

5 [15 pts] Neural Networks

Suppose that you have two types of activation functions at hand:

- Identity function: $g_I(x) = x$
- Step function: $g_s(x) = 1$ if $x \geq 0$, 0 otherwise

So, for example, the output of a neural network with one input x, a single hidden layer with K units having step function activations, and a single output with identity activation can be written as $out(x) = g_I(w_0 + \sum_{i=1} w_i g_s(w_0^{(i)} + w_1^{(i)} x))$, and can be drawn as in Fig. 5

1. Consider the step function: $u(x) = y$ if $x < a$, 0 otherwise.

   Construct a neural network with one input x and one hidden layer whose response is $u(x)$. Draw the structure of the neural network, specify the activation function for each unit (either g_I or g_s), and specify the values for all weights (in terms of a and y).

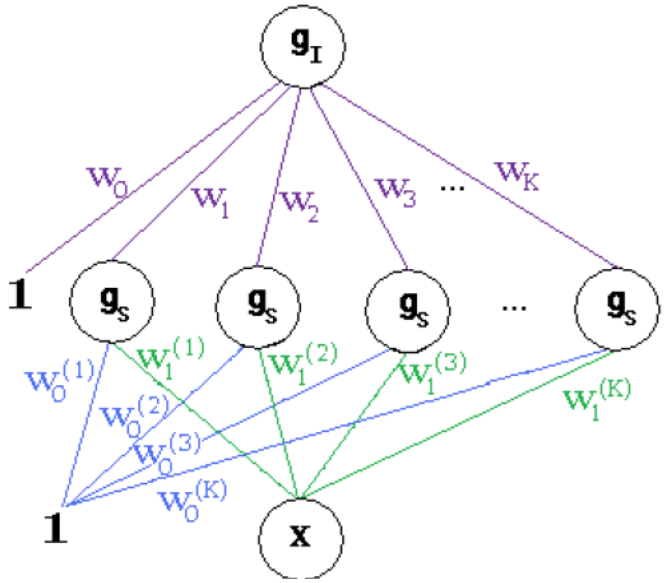


Figure 1: Example Neural Network for Q. 5

2. Now, Construct a neural network with one input x and one hidden layer whose response for given real values y, a, and b is y if $x \in [a, b)$, and 0 otherwise. Draw the structure of the neural network, specify the activation function for each unit (either g_I or g_s), and specify the values for all weights (in terms of a, b, and y).

6 [20 pts] DTrees vs Neural Networks

Consider the problem of learning a function $Y = f(X_1, X_2, \ldots X_n)$ over boolean valued attributes $X_1, X_2, \ldots X_n$. For each of the following cases, state if it is possible for a Decision tree and a Neural Network(NN) to compute this function. If possible, describe/draw the resulting decision tree/neural network. Also, for the decision tree, state the number of levels in the tree; for the NN, state the number of hidden layers that your NN uses.

1. f = Majority function. I.e. $f(X_1, X_2, \ldots X_n) = 1$ if atleast half the X_is are 1; 0 otherwise.
2. f = Parity function. I.e. $f(X_1, X_2, \ldots X_n) = 1$ if the sum of all X_is is odd; 0 otherwise.