CS 410/510
Machine Learning
Spring, 2007

Homework 8:
Review for Final

Due Thursday, June 7. We will go over this homework in class on Thursday, so late homework will not be accepted for this assignment.

Hand in neatly written answers to each question, in the spaces provided below. Show your work.

1. Suppose you have run 5-fold cross-validation on two learning methods, L_1 and L_2. The results are:

Test set	**Test error of L_1**	**Test error of L_2**
T_1	.1	.11
T_2	.14	.12
T_3	.2	.17
T_4	.18	.18
T_5	.22	.17

With approximately what confidence can we accept the hypothesis that L_2 has lower true test error than L_1?

2. Consider the equation

$$x_2 = -x_1 + \frac{1}{2}.$$

(a) Give the weights w_1 and w_2 and bias θ that would implement a two-input perceptron whose decision surface is this line. (There is more than one possible solution—you just need to give one solution.)

(b) What Boolean function is implemented by this perceptron?

(c) Sketch the line and show that it correctly separates the two classes defined by this Boolean function.

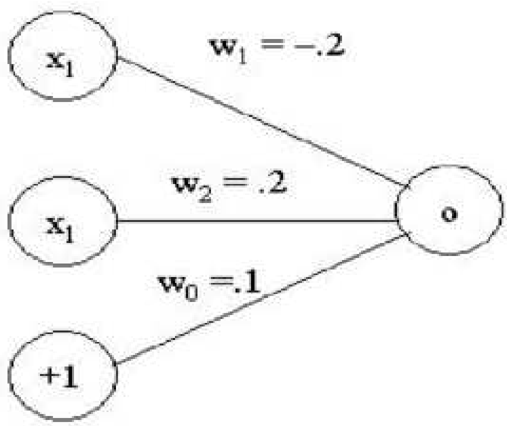
(d) The so-called stochastic "delta" (or "perceptron") learning rule (with learning rate 1) is: $w_i \leftarrow w_i + \Delta w_i$, with

$$\Delta w_i = (y - o))x_i,$$

where $o = sgn(\sum_i x_i w_i)$, y is the correct output for the given training example $\vec{x}$, and $sgn(z)$ is 1 if $z \geq 0$, -1 otherwise. This weight update rule is applied for each training example presented to the perceptron. Given the following training examples:

x_1	x_2	y
0	0	-1
0	1	-1
1	1	1

and the perceptron pictured below (with initial weights as given), show what happens to the weights at each step if the perceptron learning rule is used with these training examples over one epoch.



(e) What function has been learned by the perception in part (d) after the given training examples have been processed? You can give the function as a truth table.

(f) Give a Boolean function of three inputs that is not linearly separable.

(g) Suppose you are using a two-layer neural network (i.e., one layer of hidden units) to learn a classification task. Give one possible advantage of increasing the number of hidden units. Give one possible disadvantage of increasing the number of hidden units.

3 (a) Suppose you have three hypotheses, h_1, h_2, and h_3. Suppose each has a generalization error of 0.45. If the errors of the three hypotheses are independent, what is the probability that a majority vote among the three classifiers on a new instance will be correct?

(b) Repeat part (a), but assume the generalization error of each of the three hypotheses is 0.55.

(c) Suppose you want to use either bagging or boosting to improve the performance of learning method L. Short of trying out bagging and boosting and comparing the results, how might you decide which of these two ensemble methods would be likely to give better performance using L?

4. (a) In a few sentences, describe three machine learning applications for which you think genetic algorithms would be a good method to use.

(b) Choose one of the applications you listed in part (a), and describe in a few sentences:

- How you might represent candidate solutions as individuals in a GA population.
- A possible fitness function for these individuals.

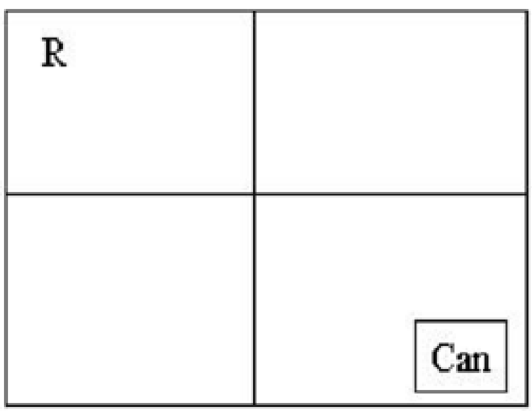
5. (a) Suppose Robby the robot lives in a $2x2$ grid (pictured below), and is applying Q-learning to create a policy to follow. At each time step, he can see the contents ("empty" or "can") in his current site, and the north, south, west, east sites. His possible actions are *move north*, *move south*, *move east*, *move west*.

The reward function is as follows: -1 for each move he makes, 5 each time he lands in a site with a can.

Let the $\hat{Q}$ matrix be initialized to all zeros.

Apply Q-learning for two epochs, with discount factor $\gamma = 0.8$ and learning rate $\eta = 1$. Each epoch consists of four time steps. At each epoch Robby starts in the upper-left corner site, and at each time step Robby moves one step in a clockwise direction. The environment does not change between epochs.

For each time step t in each epoch, give the new value of all $\hat{Q}(s, a)$ entries that change value at that time step.



(b) Given the learned Q-matrix from part (a), describe how would you use it to decide what action to take next given the following configuration:

R	Can	
Can		

(b) There are many possible ways for a Q-learning agent to select an action to perform at each step. Suggest (i) an action-selection method that is biased toward *exploitation* and (ii) an action-selection method that is biased toward *exploration*.

(c) Consider writing a learning algorithm to play the game of checkers. Describe in a couple of paragraphs how you might apply TD-learning to this problem. Explain what you would use as states, actions, rewards, how you would represent the $\hat{Q}$-function, and how you would perform learning.