Machine Learning 15:681

Midterm Exam, Fall 1995

Tom Mitchell, CMU

Instructions:

1. Write your name at the top of each page.
2. Answer questions in the space provided. If you need extra space, use the back of the preceding sheet.
3. Short, precise answers will receive the most credit. Long rambling answers are worth fewer points.

- *Hint:* A calculator is *not* necessary for this test.
- *Hint:* The extra credit question may be time consuming. Don't even think about working on it until you are finished with all the other questions!

1. [10 points] True or false? If true, give a one-sentence explanation. If false, a one-sentence counter-example or explanation.

- The version space algorithm cannot be applied if H contains any disjunctive hypotheses.

- Consider the two-input perceptron $P1$ defined by the weights $w_0 = -0.4, w_1 = 1.0, w_2 = 2.0$, and the two-input perceptron $P2$ defined by weights $w_0 = -0.9, w_1 = 1.0, w_2 = 2.0$. True or false: $P1$ is *more-general-than* $P2$.

- The sample complexity bounds of the PAC learning theorems from chapter 3 apply even if the learner is allowed to actively query the trainer.

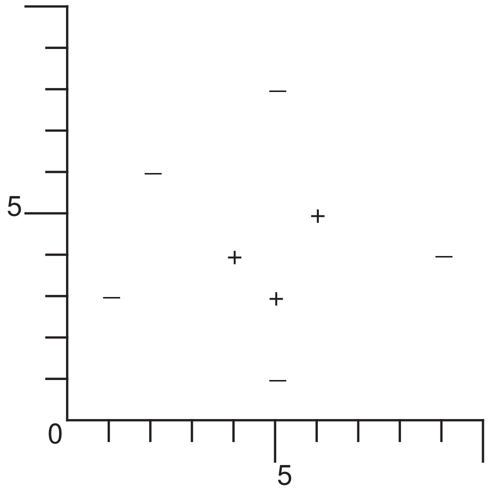
- When training multilayer neural nets, we choose sigmoid units rather than perceptrons so that there won't be local minima that might be found by the gradient descent search.

- ID3 is an example of learner that (with respect to discrete-valued functions over instances having discrete-valued attributes) has no inductive bias.

2. [15 points] Concept learning and version spaces

Consider the instance space consisting of points in the x, y plane, and the set of hypotheses H consisting of rectangles. More precisely, hypotheses may be either of the form $a < x < b, c < x < d$, or of the form $a \leq x \leq b, c \leq y \leq d$ where a, b, c and d can be any real numbers.

- Consider the version space with respect to the set of postive (+) and negative (-) training examples shown below. These examples happen to all have integer values for x and y. What is the S boundary of this version space? Write out the hypotheses in S below, *and* draw them in on the diagram.



- What is the G boundary of this version space? Write out the hypotheses *and* draw them in.

- Suppose the learner may now suggest a new x, y instance and ask the trainer for its classification. Suggest a query guaranteed to reduce the size of the version space, regardless of how the trainer classifies it. Draw it in on the diagram as a "X." Suggest one that will not. Draw it in as an "O."

- (*Extra credit:*) Assume the instances given above are generated at random with uniform probability over the region bounded by (0,0) and (10,10) (with probability zero outside this region). Assume also that the target concept c is contained in H. How ϵ-exhausted is the above version space? That is, what is the minimum ϵ for which you can show that it is ϵ-exhausted?

3. [10 points] Decision trees
Consider the following training examples.

Instance	Classification	A1	A2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

- What is the entropy of this collection of training examples with respect to the target function classification?

- What is the information gain of $A2$ relative to these training examples?

- What is the reason that one typically holds out training data during decision tree learning?

4. [20 points] Perceptron trees.

Consider an instance space X in which each instance is described by a vector of real-valued attributes (e.g., $temperature, humidity$). Suppose we wish to learn boolean concepts defined over X, represented by what we will call *perceptron trees*[1] A perceptron tree is a decision tree where each decision node in the tree is made by a perceptron that considers all attributes of the instance. The perceptron output (0 or 1) is used in the same way that decision trees typically use boolean attributes as tests. (see blackboard)

- The decision surface of a perceptron is a line in hyperspace. Draw a typcial decision surface for a perceptron tree. More specifically, assume instances are described by just two real-valued attributes, and consider a tree of the form shown on the blackboard

- Draw a typcial decision surface of a normal decision tree of the same form, defined over the same two-attribute instance space. A "normal" decision tree, of course, is one that tests only one attribute per decision node. Assume that the tree tests a real-valued attribute by simply thresholding it as discussed in the chapter on decision trees.

- Sketch a greedy top-down algorithm for learning perceptron trees. You may use the subroutine $Train - perceptron(training - examples)$ if you like.

- (*Extra credit:*) Give a lower bound on the VC dimension for depth 2 perceptron trees defined over instances with two real-valued attributes.

[1] See Utgoff, 1988.

5. [15 points] Artificial neural networks

- Describe a sigmoid unit that represents the boolean function $x_1\ OR \neg x_2$. Assume values input to the unit are either 1 (true) or 0 (false). The output will be interpreted as "true" if it is greater than 0.5.

- Derive a gradient descent training rule for a single unit with output o where

$$o = w_0 + w_1 x_1 + w_1 x_1^2 + \ldots + w_n x_n + w_n x_n^2$$

6. [15 points] PAC learning

- What is the VC dimension of circles in the x, y plane?

- Suppose instances are drawn at random from some unknown probability distribution over the x, y plane. Consider a learner using the hypothesis space H consisting of circles in the plane, and assume the target concept is contained in H. How many training examples are sufficient to assure, with 95% probability, that the learned hypothesis will have error less than 10%? (you may leave your answer in the form of an expression if you have no calculator).

- Suppose you are the trainer, attempting to teach this learner a specific target concept (i.e., circle). What is the smallest number of training examples you can present to assure, with probability 1, that the learned hypothesis will have error less than 10%? Answer this question assuming

  1. Instances are generated by a uniform distribution over the region of the plane bounded by (0,0) and (100,100) (with zero probability outside this region).

  2. Instances are drawn according to some unknown distribution.