# Machine Learning 15:681

# Midterm Exam, Fall 1996

Avrim Blum and 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.

Note 80 minutes, 100 points total
Good luck!

1. [15 points] Short answers.

True or false: If true, explain why in a sentence. If false, give a counterexample.

a. Consider two perceptrons defined by the threshold expression $w_0+w_1x_1+w_2x_2 > 0$. Perceptron $A$ has weight values

$$w_0 = 1,\ w_1 = 2,\ w_2 = 1$$

and perceptron $B$ has the weight values

$$w_0 = 0,\ w_1 = 2,\ w_2 = 1$$

True or false: Perceptron $A$ is *more_general_than* perceptron $B$.

b. The hypothesis space of 3 input perceptrons is an example of a biased hypothesis space.

c. A decision list can compute the XOR function.

d. For sufficiently large sample sizes $n$, the Binomial distribution can be approximated by a Normal distribution.

True or false (continued):

e. Each time `ID3` adds a new decision node to the tree, it creates a new tree that is *more_specific_than* the original. (assume we are learning a boolean–valued target function)

For the two questions below, state whether this is a legitimate use of the "Occam's Razor" PAC model analysis, and if not, give a one-sentence explanation why not.

f. We draw 1000 examples from a distribution, and say that with high probability there is no small hypothesis consistent with the data that has high true error.

g. We repeatedly draw 1000 examples from a distribution until we find a data set that is consistent with some small hypothesis. We then say that with high probability this hypothesis has low true error.

2. [40 points] In this problem, we will examine the hypothesis class $H_{rd2}$ of "regular, depth-2 decision trees" over $n$ Boolean variables. A "regular, depth-2 decision tree" is a depth-2 decision tree (a tree with four leaves, all distance 2 from the root) in which the left and right child of the root are *required to contain the same variable.* For instance, the following tree is in $H_{rd2}$:

```
        x3
       /  \
     x1    x1
    / \   / \
   +   - -   +
```

We will first consider the special case of $n = 2$ (there are just two variables, $x_1$ and $x_2$):

a. How many syntactically distinct trees are there in $H_{rd2}$?

b. How many *semantically* (or *logically*) distinct trees are there in $H_{rd2}$? (two trees are semantically distinct if they represent different functions).

c. Given the following data $D$, write down (one syntactic form of) the most specific and most general boundary sets for the Version Space $VS_{H_{rd2},D}$.

| $x_1$ | $x_2$ | class |
|---|---|---|
| 1 | 0 | $+$ |
| 0 | 0 | $-$ |

2 (continued). Now consider the case of general $n$:

d. As a function of $n$, how many syntactically distinct trees in $H_{rd2}$ are there?

e. Give an upper bound for the number of examples needed in the PAC model to learn $H_{rd2}$ with error $\epsilon$ and confidence $\delta$.

f. Consider the following combination of the Weighted Majority algorithm and Version Spaces, for the class $H_{rd2}$. You begin with all hypotheses in $H_{rd2}$ having weight 1. Every time you see a new example, you predict based on a weighted majority vote over all hypotheses in $H_{rd2}$. Then, instead of throwing out the inconsistent trees, you cut down their weight by a factor of 2.

How many mistakes will this procedure make at most, as a function of $n$ and the number of mistakes of the best tree in $H_{rd2}$? (You may use $O()$ notation).

extra credit: As a function of $n$, exactly how many semantically (logically) distinct trees in $H_{rd2}$ are there?

3. [15 points] "Hill-Climbing" is an algorithm similar to gradient descent for discrete hypothesis spaces. Hill-Climbing works like this: you begin with some hypothesis $h$ in your hypothesis space $H$. You then calculate the error of $h$, and the errors of all "neighbors" of $h$ on your data set (you need to define a notion of "neighborhood" to run the algorithm). If you find that some neighbor of $h$ has lower error than $h$ does, you replace $h$ by its neighbor having the least error and continue. Otherwise, if all neighbors of $h$ have error at least as large as $h$ does, you halt and output $h$ as your solution.

We will consider the particular case where $H$ is the class of monotone conjunctions (no negations), and we define two conjunctions as being "neighbors" if you can remove one variable from one of the conjunctions to make it equal to the other. For instance, if there are 4 boolean variables and h $= x_1 \wedge x_2 \wedge x_3$, then the neighbors of $h$ would be: $x_1 \wedge x_2$, $x_2 \wedge x_3$, $x_1 \wedge x_3$, and $x_1 \wedge x_2 \wedge x_3 \wedge x_4$.

a. Starting with the hypothesis that is the conjunction of zero variables (it says that everything is positive), run hill-climbing on the following data set. List the sequence of hypotheses produced.

| $x_1$ | $x_2$ | $x_3$ | class |
|---|---|---|---|
| 0 | 0 | 1 | $-$ |
| 0 | 1 | 0 | $-$ |
| 1 | 1 | 0 | $-$ |
| 0 | 1 | 1 | $+$ |

3. (continued)

b. Is Hill-Climbing guaranteed to find a consistent monotone conjunction whenever one exists? If so, say why in one paragraph. If not, give a counterexample (i.e, a set of data and a starting hypothesis for which hill-climbing fails).

4. [30 points] Consider a learning task in which instances are described by $n$ boolean variables (e.g., $x_1 \wedge \bar{x_2} \wedge x_3 \ldots \bar{x_n}$), and are drawn according to a fixed but unknown probability distribution $\mathcal{D}$. The target concept is known to be describable by a conjunction of boolean attributes and their negations (e.g., $x_2 \wedge \bar{x_5}$), and the learning algorithm uses this concept class as its hypothesis space $H$. The learner is provided a set of 100 training examples drawn according to $\mathcal{D}$. It outputs a hypothesis $h$ from $H$ that is consistent with all 100 examples (i.e., the error of $h$ over these training examples is zero).

- We are interested in the true error of $h$, that is, the probability that it will misclassify future instances drawn randomly according to $\mathcal{D}$. Based on the above information, can you give an interval into which this true error will fall with at least 95% probability? If so, state it and justify it briefly. You may leave it in the form of an unsimplified expression. If not, explain the difficulty.

- You now draw a new set of 100 instances, drawn independently according to the same distribution $\mathcal{D}$. You find that $h$ misclassifies 30 of these 100 new examples. Can you give an interval into which this true error will fall with approximately 95% probability? (ignore the performance over the earlier training data for this part). If so, state it and justify it briefly. You may leave it in the form of an unsimplified expression. If not, explain the difficulty.

- It may seem a bit odd that $h$ misclassifies 30% of the new examples even though it perfectly classified the training examples. Is this event more likely for large $n$ or small? Justify your answer in a sentence.