University of Wisconsin-Madison
Computer Sciences Department

CS 760 — Machine Learning

Spring 1990

Midterm Exam

(five pages of notes allowed)

100 points, 90 minutes

May 1, 1990

Write your answers on these pages and show your work. If you feel that a question is not fully specified, state any assumptions you need to make in order to solve the problem. You may use the backs of these sheets for scratch work. Notice that all questions do not have the same point-value. Divide your time appropriately.

Before starting, write your name on this and all other pages of this exam. Also, make sure your exam contains 4 problems on 8 pages.

Problem	Score	Max Score
1	_______	25
2	_______	15
3	_______	30
4	_______	30
Total	_______	100

1. Similarity-Based Learning (25 points)

ID3 can be easily extended to handle numerically-valued features. The basic idea is as follows. When computing the *max gain*, dynamically convert all of the numerically-valued features into a collection of Boolean-valued features. For example, if the current set of examples have the values 1, 3, 7, and 17 for feature *X*, define the new Boolean features *X<2*, *X<5*, and *X<12*. (That is, plot the existing values on a number line and consider all possible ways to "chop" the line so that there are values on both sides of the cuts.) Compute the gain of each of these Boolean-valued features, as well as that for any existing nominal features, and insert the best feature into the decision tree being built.

Assume the features *weight (W)* and *height (H)*, each of which takes on values in [0..100], are used to describe some examples as follows:

class	***features***
+	*H = 10, W = 10*
+	*H = 50, W = 50*
—	*H = 20, W = 20*
—	*H = 70, W = 30*

part a (20 points)

Compute an ID3 tree using Quinlan's max-gain formula for these examples. ***Show all your work.***

part b (5 points)

How sensitive do you think this technique would be to the presence of *noise*?

2. Learning without a Teacher (15 points)

Consider the UNIMEM system. Show the concept hierarchy that results after *each* of the following instances are processed. (That is, you should draw *eight* concept hierarchies, one to the right of each instance, and make sure to enumerate **both** the concept's features and the associated instances.)

Assume that an instance is considered to be a member of a concept if it matches the concept in *at least* two unexplained feature values. Make the same assumption when deciding if two instances are "close enough" to form a new concept. If a new concept is to be formed, its description must contain at least two features. Do *not* worry about discarding and "freezing" features, nor consider discarding a concept.

The feature definitions are as follows:

FEATURE	VALUES
OP	{plus, minus}
NUM-BND	{low, mid, high}
SOLUTION	{T, F, partial}
TYPE	{join, combine, compare}
STRATEGY	{basic, inter, adv, indeterm}

```
instance 1
   OP         plus
   NUM-BND    mid
   SOLUTION   T
   TYPE       join
   STRATEGY   inter
```

```
instance 2
   OP         plus
   NUM-BND    high
   SOLUTION   F
   TYPE       combine
   STRATEGY   adv
```

```
instance 3
   OP         plus
   NUM-BND    low
   SOLUTION   partial
   TYPE       compare
   STRATEGY   indeterm
```

```
instance 4
   OP         plus
   NUM-BND    mid
   SOLUTION   F
   TYPE       compare
   STRATEGY   adv
```

```
instance 5
   OP         plus
   NUM-BND    mid
   SOLUTION   partial
   TYPE       combine
   STRATEGY   adv
```

```
instance 6
   OP         plus
   NUM-BND    high
   SOLUTION   F
   TYPE       combine
   STRATEGY   indeterm
```

```
instance 7
   OP         plus
   NUM-BND    mid
   SOLUTION   F
   TYPE       combine
   STRATEGY   adv
```

```
instance 8
   OP         minus
   NUM-BND    mid
   SOLUTION   F
   TYPE       compare
   STRATEGY   indeterm
```

3. Explanation-Based Learning (30 points)

part a (10 points)

One technique used in some EBL systems is to replace constants in the explanation by variables. This can result in improper generalization. Consider the problems of *under* and *over* generalization.

Can *under* generalization occur with this technique? Explain why not or give an example.

Can *over* generalization occur with this technique? Explain why not or give an example.

part b (5 points)

The cs760 **EGGS** algorithm places a newly-learned rule at the *front* of the rule base. Assuming a standard PROLOG-like prover, what would be the effect on classification performance if newly-learned rules are placed at the *rear* of the entire rule-base? Consider the effect on both members and non-members of the concept.

part c (15 points)

Some EBL researchers claim that EBL systems should learn *macroperators*, while others claim EBL should learn *search-control* knowledge. Briefly describe these two types of EBL-acquired knowledge and state a strength and weakness of each.

macroperators

definition

strength

weakness

search-control

definition

strength

weakness

4. Artificial Neural Networks (30 points)

I. M. HOT graduates from college and gets a job. I. M. is assigned the task of building a letter recognizer and decides to apply a neural network approach. As a start, I. M. chooses to experiment with a 3x3 pixel array, picking four different letters as test cases: two vowels (**I** and **O**) and two consonants (**H** or **T**). The task is to differentiate between the vowels and the consonants.

part a (15 points)

Consider the simple perceptron drawn on the following page. If the weighted of its inputs equals or exceeds its bias, its output is 1; otherwise it is 0. The desired outputs for the first two examples appear. Assume all the weights are initially set to zero and that the perceptron's bias is initially -0.5. Indicate, on the top network, what the network's settings will be after the first training example (**I**). Use η = 0.1. Assume the filled pixels provide a "1" signal, while the empty units produce an "0." In the second network, show what happens after the second training example (**H**).

part b (5 points)

I. M. gets nervous upon noticing that superimposing **I** on **O** produces the same result as superimposing **H** on **T** and begins thinking about the issue of *linear separability*.

Can the delta rule learn to separate these four patterns? If not, explain why it cannot. If it can, set the weights (by hand) on the *lowest* figure on the next page (the **T**) so that the perceptron separates the **I** and **O** from the **H** and **T**.

part c (10 points)

Consider, now, the following more complicated task. Somewhere in a 10x10 pixel array one of these four letters (of size 3x3) will appear and all other pixels will be blank. Design a neural network approach that will determine if this letter is a vowel (**I** or **O**) or a consonant (**H** or **T**). Briefly describe how you will configure the network and how you would train it.

Networks for Problem 4