

University of Wisconsin-Madison
Computer Sciences Department

CS 760 — Machine Learning

Spring 1996

Midterm Exam

(one page of notes allowed)

100 points, 90 minutes

April 29, 1996

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 five (5) problems on eight (8) pages.

Problem	Score	Max Score
1	_____	25
2	_____	25
3	_____	15
4	_____	10
5	_____	25
Total	_____	100

1. Learning from Classified Examples (25 pts)

Assume you are given the following three nominal features with the possible values shown.

$$\begin{aligned} F1 &\in \{v1, v2\} \\ F2 &\in \{v3, v4, v5, v6\} \\ F3 &\in \{v7, v8, v9\} \end{aligned}$$

Use the following training examples for all parts of Question 1.

Be sure to show all your work for each part.

F1 = v1	F2 = v3	F3 = v7	category = +
F1 = v1	F2 = v3	F3 = v8	category = -
F1 = v1	F2 = v4	F3 = v9	category = +
F1 = v2	F2 = v5	F3 = v9	category = -
F1 = v2	F2 = v6	F3 = v8	category = +

Part A. What score would Quinlan's *info gain* formula assign to each of these three features? Which one would be chosen by ID3 as the root node?

Part B. What would Quinlan's *info gain ratio* choose as the root node?

Part C. Consider extending Aha et al.'s distance formula by weighting the distance along each dimension of feature space by the info gain of that feature. Using this scoring function, how would a *one-nearest-neighbors* algorithm classify the following test example?

F1 = v2 F2 = v4 F3 = v9 category = ?

Part D. How would the *Naive Bayes* algorithm classify Part C's test example? (Assume that any feature-value pair not seen in the above training examples occurs once in every hundred positive examples and also once in every hundred negative examples.) What is the main limiting assumption made by the Naive Bayes algorithm?

2. Reinforcement Learning and the Perceptron Algorithm (25 pts)

Assume you have a deterministic, four-state Markovian world, and that you have chosen to represent each state by its binary encoding (e.g., 00, 01, 10, 11). This world's actions are *leftOn*, *leftOff*, *rightOn*, and *rightOff*, where:

<i>leftOn</i>	moves from state xy to state $1y$
<i>leftOff</i>	moves from state xy to state $0y$
<i>rightOn</i>	moves from state xy to state $x1$
<i>rightOff</i>	moves from state xy to state $x0$

The reward received for entering a state is: *left bit - right bit*.

However, assume the learner does *not* know the next state and immediate reward functions; all the learner knows is the actions it can perform, plus it can measure the state it is in and the immediate reward it receives.

Part A. Draw this world as a graph, where states are nodes and actions are arcs. Be sure to label the nodes and arcs according to the problem statement, including the immediate reward received for following an arc.

Part B. How many cells would a Q-table for this environment contain? Instead of using a Q-table, consider using *perceptrons* to represent the Q-function. Sketch below how this would be set up. How many free parameters are needed when using perceptrons? Initialize all these free parameters to zero.

Part C. Assume the learner starts in state 00 and chooses the action *leftOn*. Show how Watkin's *one-step Q-learning* algorithm would change the weights in your perceptrons. Use a discount rate of 0.9 and a perceptron learning rate of 0.5. Be sure to explain your calculations.

Part D. Assume the learner next chooses the *leftOff* action. Show how Q-learning changes your connectionist Q-function after this experience.

Part E. Briefly explain how Sutton's *DYNA* algorithm could be applied to this task. What advantage does it provide?

3. Feedforward Neural Networks (15 pts)

Part A. When using neural networks to learn a Boolean-valued function, a common error measure is the square of the difference between the teacher's and network's outputs. Based on Rumelhart et al.'s Bayesian analysis of what a network should optimize, *briefly* explain the mistake in using squared error for Boolean-valued functions.

Part B. Assume you are training a feedforward neural network that contains i input units, one layer of h hidden units, and o output units, where the inputs fully connect into the hidden units and the hidden units fully connect into the outputs (i.e., the 'standard' architecture you used in HW 3). Assume the hidden and output units use the $output = \tanh(input)$ activation function, whose derivative is $(1 - output^2)$. Note: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

Show that if the weights and biases of such a network are initialized to zero and the squared-difference error function is used, then backpropagation training will not alter the network's weights.

Part C. Explain and briefly justify the *early stopping* technique for deciding how many epochs should be used when training neural networks.

4. Explanation-Based Learning (10 pts)

Consider the following EBL domain theory. Terms beginning with ?'s are implicitly universally-quantified variables.

$$\begin{array}{ll} A(?x1, ?x1) \wedge B(?y1, ?z1, ?y1) & \rightarrow C(?x1, ?y1, ?z1) \\ A(?x2, ?x2) \wedge D(?y2, ?y2) & \rightarrow B(?x2, ?y2, ?z2) \end{array}$$

Assume the following problem-specific facts are asserted:

$$\begin{array}{llll} A(1, 1) & A(3, 1) & D(1, 1) & D(2, 2) \\ A(2, 2) & A(3, 3) & D(2, 3) & D(3, 3) \end{array}$$

Part A. Explain, with a proof tree, that $C(1, 2, 3)$ is true. Draw to the *right* of your proof tree the corresponding *explanation structure*. Clearly indicate the necessary unifications by marking them with three parallel lines.

Part B. Assuming that predicates A and D are operational, what rule would Mooney's *EGGS* algorithm produce? Explain.

5. Short Essays (25 pts)

Part A. Explain why proper experimental methodology requires the use of both a *tuning* set and a *testing* set.

Part B. Based on Holland's *Schema Theory*, explain why features that are believed to strongly interact should be placed near one another when designing the bit string that represents an individual in a genetic algorithm.

Part C. Describe one advantage of Fisher's *COBWEB* algorithm over Rumelhart and Zipser's *Competitive Learning* technique.

Part D. What problem does the *minimal description length* principle address? How does this principle manifest itself in Rumelhart et al.'s Bayesian analysis of what should be optimized by a neural network?

Part E. Briefly describe how one could use a *k-nearest-neighbor* algorithm in reinforcement learning. (Use back of this sheet to answer.)