CS472

Foundations of Artificial Intelligence

Final Exam CS472 / Q Exam

December 14, 1999

12-2:30pm

Uris Auditorium

Name:

Instructions: You have 2.5 hours to complete this exam. The exam is a closed-book exam. Two pages of notes allowed. [***Notes will not be allowed in the 2000 CS472 final.

#	description	score		max score	Q exam score		Q exam max score
1	search	____	/	15	____	/	15
2	logic	----	/	15	____	/	15
3	decision trees	____	/	20	____	/	20
4	neural net	____	/	25	____	/	25
5	general	____	/	15	____	/	15
6	csp / SAT	____	/	10	____	/	25
7	search (Q only)	xx	/	xx	____	/	15
8	resolution (Q only)	x	/	xx	____	/	10
9	learning (Q only)	xx	/	xx	____	/	10
Total score:		________	/	100	_______	/	150

Problem 1: Search (15 points)

Consider a puzzle where 2N blocks are aligned in a **ruler** with 2N+1 positions. There are N blue (B) and N white (W) blocks and an empty position.

Suppose that the goal is to have all the white blocks positioned to the left of the blue ones AND one blue block on the rightmost position — the empty position is not specified.

Blocks can *hop* to the empty position when the empty position is *at most N cells away.* This means that a block can only move half the length of the ruler; not the whole length. The cost of Hop(i), $i = 1, \cdots, N$ is i. There are no "circular" hops (*i.e.* hops wrapping from one end of the ruler to the other).

For this problem N=2.

Consider the initial configuration:

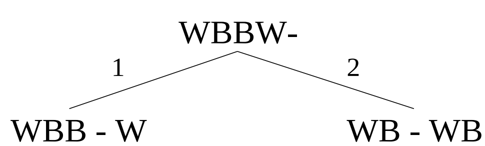
W	B	B	W	-

(Initial configuration)

Use the following rules when answering the questions below:

- Repeated states (anywhere in tree) are detected, and therefore are not repeated in the search tree.
- Successor states are expanded in a fixed order: hops that result in a move of the empty position to the left are done before hops that move the empty position to the right; hops in each direction are done in an increasing order of their cost.

1. (7 pts.) Give the next layer of the search tree below following the rules stated above.



2. (4 pts.) Consider that you are performing **Breadth First Search** using the tree as expanded in part 1. If you are told that the queue contains *five* nodes, what are the five nodes in it? Use labels on the nodes, *e.g.*, label them $n1$, $n2$, etc.

3. (4 pts.) Consider that you are performing **Uniform Cost Search** using the tree as expanded in part 1. If you are told that the priority queue contains *three* nodes, what is one possibility for what those nodes are? (Use same labels on nodes as in previous part.)

Problem 2: Representation and Logic (15 points)

1. (5pts.) State in English what each one of the formulas represents.

   - $President(\text{Clinton})$.

   - $\forall x \; President(x) \Rightarrow (((Republican(x) \vee Democrat(x)) \wedge \neg(Republican(x) \wedge Democrat(x)))$

2. (5 pts.) State in First Order Logic:

   Note: $VotedAgainst(y, x)$ means that x votes against y.

   - Every President voted against someone.

   - Joe hasn't voted against anyone.

3. (5 pts.) From the statements in part 1, show that

   $Republican(Clinton) \vee Democrat(Clinton)$

   Give resolution proof. Use clausal form (conjunctive normal form).

Problem 3: Decision Trees (20 points)

1. (5 pts.) Show the decision tree that would be learned by a decision tree learning procedure using maximum information gain given training examples for EnjoySport shown in the following table. (Hint: you do not need a calculator for this.)

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

2. (5 pts.) Create a new decision tree by extending your previous tree, replacing one of the leaf nodes by a decision with new leaves below it. Create this new tree so that it:

   - classifies more cases positively than the original tree.
   - remains consistent with the four training examples of the table above.

   Note: consider both training and possible test cases.

3. (5 pts.) Again, begin with your tree from the first part of this question. This time extend it to create a new tree that:

   - classifies more cases negatively than the original tree.
   - remains consistent with the four training examples of the table above.

   Note: again, consider both training and possible test cases.

4. (5 pts.) Our decision tree learning method using information gain has an inductive bias that favors short trees. Give one argument against replacing this by a bias that favors maximally general or maximally specific trees. [answer in two sentences or less.]

Problem 4: Neural Networks (25 points)

1. (10 pts.) Consider a perceptron with inputs $x1$ and $x2$, and with weight values $w_0 = 0$, $w_1 = 1$, $w_2 = -3$. (We assume an extra input x_0 fixed at 1 to capture the threshold.) Draw the decision surface of this perceptron; that is, indicate which portion of the x_1, x_2 plane produces outputs of +1 and -1.

2. (5 pts.) True or false? The perceptron defined by $w_0 = 2$, $w_1 = 1$, $w_2 = -3$ classifies more cases positively than the perceptron in the previous part of this question. Explain your answer briefly.

3. (10 pts.) A 2-of-3 majority function is a function of three binary (0/1) inputs, X_1, X_2, and X_3 which outputs 1 when *two or more* of the inputs (X_i's) have a value 1, and outputs 0 otherwise.

(a) Draw a decision tree for the 2-of-3 majority function.

(b) Draw a feedforward neural network (consisting of one of more units) for the 2-of-3 majority function. Use as few units as possible. Make sure you show the weights and threshold values.

Problem 5: General Questions (15 points)

Below is a set of general questions. Provide a short answer for each of them.

1. (5 pts.) Which of the following may converge to a locally optimal solution (i.e., not globally optimal)? (1) Backpropagation with multi-layer networks. (2) The perceptron training rule applied to a single perceptron.

2. (5 pts.) In learning, what do we mean by the generalization error? What do we mean by the training error? In general, does a more expressive hypothesis class lead to a smaller or larger training error? What about the generalization error? What is cross-validation?

3. (5 pts.) In one sentence each, give (a) one advantage of decision tree learning over backpropagation; (b) one advantage of backpropagation over decision tree learning.

Problem 6: CSP / Boolean encodings, Sorting Networks — PART A (10 pts.)

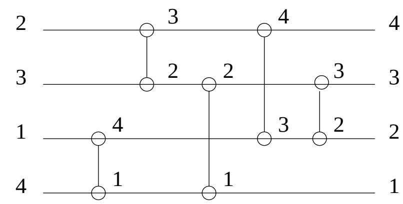


Figure 1: An example sorting network with input $< 4, 1, 3, 2 >$.

Consider the sorting network in Figure 1. A sorting network takes as input N numbers that are fed into the network on the left and outputs the N numbers on the right in numerical (nondecreasing) order (largest value at top). The figure gives a 4 input sorting network ($N = 4$). Each vertical line (a "comparator") represents a comparison of two input values and swaps them if necessary. For example, the left-most comparator in the figure considers the values on the first (bottom) and the second wire, and swaps them if necessary. If the values are already in the right order, the comparator does nothing.

A correct sorting network sorts any arbitrary sequence of N numbers into a nondecreasing order. An interesting research question is to find, for a given number of inputs, the shortest sorting network (i.e., fewest possible comparators) that sorts all possible sequences correctly.

Fact: One can show that a sorting network for sorting N numbers is correct iff it sorts all 2^N binary sequences (zeros and ones) correctly.

1. (10 pts.) Outline how one could write a depth first search routine that searches for a correct sorting network with N inputs and k comparators (N and k fixed). (Ignore efficiency issues.)

PART B — Q EXAM ONLY (15 pts.)

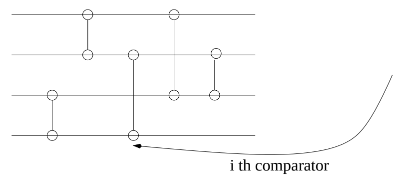


Figure 2: Variables used in Boolean encoding.

2. **(15 pts.)**

We now consider searching for a correct sorting network by encoding the problem as a Boolean satisfiability problem. To simplify our task, we will consider finding a network that correctly sorts a single binary input sequence. Let the number of inputs N and the number of comparators k be fixed.

You're task is to give a Boolean encoding where each satisfying assignment corresponds to a correct sorting network for the given binary input sequence.

We introduce the following variables. See Figure 2.

We use variables to represent the zeros and ones at each step in the network. The input values are given by $x_1^0, \ldots, x_N^0$. So, if the input value on the bottom wire is 1, then x_1^0 should be set to true; if the value is 0, the variable should be set to false.

The values on the wires after passing through the first comparator are represented by $x_1^1, \ldots, x_N^1$. The values after the lth comparator are given by $x_1^l, \ldots, x_N^l$. And, finally, the output values are represented by $x_1^k, \ldots, x_N^k$.

A comparator at position i is defined by two sets of variables: $c_{T,1}^i, \ldots, c_{T,N}^i$ and $c_{B,1}^i, \ldots, c_{B,N}^i$. ("T" for Top and "B" for bottom.)

For example, the left-most comparator in Figure 2 — comparing values on the first and the second wire — would be encoded by setting $c_{T,2}^1$ and $c_{B,1}^1$ to true and all other variables used in representing this comparator to false (i.e., $c_{T,1}^1$, $c_{T,3}^1, \ldots c_{T,N}^1$ and $c_{B,2}^1$, $c_{B,3}^1, \ldots c_{B,N}^1$ are set to false).

(a) (2 pts) Let $N = 4$ and $k = 5$. Consider the input sequence $< 1, 0, 0, 1 >$. What clauses do you need to assert to capture the correct input and output behavior of the sorting network?

(b) (3 pts) Give clauses that encode that there is exactly one comparator at location i ($1 \leq i \leq k$).

(c) (10 pts) Give clauses that encodes the effect of the values on the wires after passing through the ith comparator. (Note you have to give clauses that relate the variables x_j^{i-1} to x_j^i and $c_{T,j}^i$ and $c_{B,j}^i$, for a given i and with $j = 1, \ldots, N$.)

Problem 7: Search — Q EXAM ONLY (15 pts.)

Let S be a finite set of states, O a finite set of operators (each of which is a function $f : S \to S$) and $E : S \to \mathrm{R}^{+}$ an energy function to be minimized. Assume that every state can be reached from every other state via some sequence of operators. Define $\Delta E(s, r) = E(s) - E(r)$ (note that s is "better" than r if this quantity is negative).

Let $T > 0$ be a fixed constant, and consider the following two local search algorithms:

A: initialize s to a state drawn uniformly at random from S
  repeat forever:
    if there is an f in O such that $\Delta E(f(s), s) < 0$
      $s := f_{\min}(s)$, where $f_{\min}$ is an operator yielding the biggest drop in energy;
    else
      choose some g from O according to the distribution $P(g) \propto e^{-\Delta E(g(s),s)/T}$;
      $s := g(s)$;

B: initialize s to a state drawn uniformly at random from S
  repeat forever:
    choose some f from O uniformly at random
    if $\Delta E(f(s), s) < 0$
      $s := f(s)$;
    else
      with probability $e^{-\Delta E(f(s),s)/T}$, set $s := f(s)$;
      /* potentially, nothing happens in this iteration of the loop */

(i) Describe a scenario in which algorithm A converges to a global minimum of E faster than algorithm B. Explain the discrepancy.

(ii) Describe a scenario in which algorithm B converges to a global minimum of E faster than algorithm A. Explain the discrepancy.

(iii) For any S, O, and E satisfying the conditions above, does algorithm A find a global minimum of E with probability converging to 1 as it is run for more and more steps? Explain your answer.

(iv) For any S, O, and E satisfying the conditions above, does algorithm B find a global minimum of E with probability converging to 1 as it is run for more and more steps? Explain your answer.

Problem 8: Resolution — Q EXAM ONLY (10 pts.)

Consider the efficiency (or lack thereof) of resolution theorem proving: We know that repeated applications of the resolution inference rule will find a proof if one exists, but there are no guarantees as to the speed of this process. Describe two heuristics that might be used to improve the efficiency of resolution theorem proving. Explain the intuition behind each heuristic.

Problem 9: Learning — Q EXAM ONLY (10 pts.)

The standard algorithm for top-down induction of decision trees (i.e. ID3 and C4.5) does not handle missing values in the training examples. By *missing values*, we are referring to missing values in the attribute-value pairs (i.e. the features) that describe each training example. We are **not** referring to missing class values (i.e. to missing values for the goal predicate).

Provide **two (2)** clear, precise methods/algorithms for handling such missing values during top-down induction of decision trees (i.e., during the training phase). For each approach, also provide a short, high-level, written description of the intuition behind the algorithm.