CSIT6000F Artificial Intelligence
Fall 2018 Final
12/12/2018
Time Limit: 180 Minutes

Name: ______________________

Stu ID: ______________________

Instructions:

1. This exam contains 11 pages (including this cover page) and 13 questions.
2. This is a closed book exam.
3. Please write only in the exam paper. You can use either pen or pencil.

Grade Table (for teacher use only)

Question	Points	Score
Production Systems	5	
Search Problem Formulation	8	
Probabilistic Transition Relation	2	
A* Search	12	
Alpha-Beta Pruning	6	
Game Theory	8	
Representation in PL	5	
Representation in FOL	10	
Uncertainty	10	
MDP	10	
Fitness Function	3	
Linear Features	3	
Perceptron Learning and GSCA Rule Learning	18	
Total:	100	

Question 1: Production Systems .... 5 points

Recall that our boundary-following robot has eight sensors s_1 - s_8 that detect if the eight surrounding cells are free for it to occupy: clockwisely, s_1 returns 1 iff the surrounding cell in the north-west direction is not free for it to occupy, s_2 returns 1 iff the surrounding cell in the north direction is not free for it to occupy, and so on. The robot has four actions: going *north*, *east*, *south*, and *west*. Now consider the following production system:

$$\begin{aligned}
\overline{s_2} &\rightarrow north, \\
\overline{s_4} &\rightarrow east, \\
\overline{s_6} &\rightarrow south, \\
\overline{s_8} &\rightarrow west, \\
1 &\rightarrow north.
\end{aligned}$$

Give the sequence of moves by a robot controlled by this production system in a 5x5 grid without any obstacles, starting at cell $(1, 5)$ (the top left corner).

Solution: east, east, east, east, south, north, south, north, ...

Question 2: Search Problem Formulation .... 8 points

The sorting problem is to sort a given list of integers into a list of sorted integers in ascending order (i.e. smallest first). The only operators allowed are swaps that exchange two neighbouring numbers.

- Formulate this problem as a search problem by describing states, initial states, actions, and goals.
- Give a non-trivial admissible heuristic function for this search problem.

Solution:

- States: lists of integers;
- Initial state: any given list of integers;
- Goal: sorted lists of integers;
- Actions: $swap(i)$ - swap the ith position of the list with its (i+1)th position. If i is the last position, then this action is not executable.
- Cost (optional, assume unit cost): each action costs one unit.
- An admissible heuristic function: number of elements in the list that are larger than its next neighbour.

Question 3: Probabilistic Transition Relation .... **2 points**

For the above sorting problem, assume now that the swapping operators are not very reliable: it only succeeds with 0.8 probability. The other times, it does something random - it is your job to give a reasonable definition of what it means by "random" by defining a probabilistic transition relation based on the states and actions that you defined for the last question. We will take any reasonable definition. (*This question may be conceptually difficult to you. It's just 2 points, so don't spent too much time on it.*)

Solution: We'll take any reasonable definition. One possibility is that with the $swap(i)$ action above, "random" means that it swaps ith element with an element other than the (i+1)th element with equal probability: $T(l, swap(i), l') = 0.8$ if $i < length(l) - 1$ (assuming the starting index is 0) and l' is the result of swapping the ith element with $(i+1)$th element in l. $T(l, swap(i), l') = 0.2/(length(l) - 1)$ if $i < length(l) - 1$ and l' is the result of swapping the ith element with another element other than the $(i+1)$th element in l. This includes the case that $l' = l$.

Question 4: A* Search .... **12 points**

Consider the following state space with the indicated initial state and the goal state. The number next to an arc is the cost of the corresponding operator, and the number next to a state is its heuristic value.

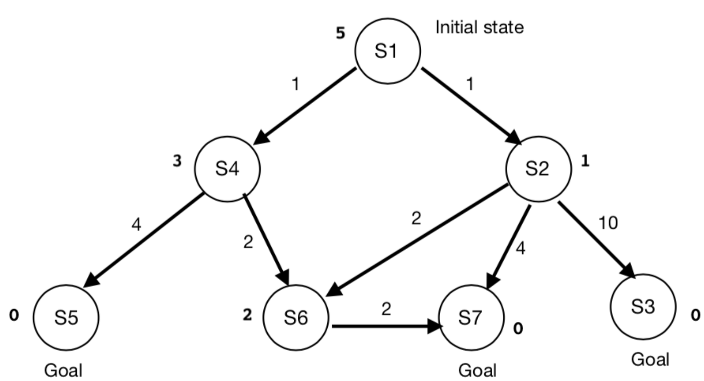


Figure 1: A search problem

- (8 pts) Give the sequence of the nodes expanded by A^* algorithm, starting from the root and terminating at a goal node. Notice that whenever there is a tie, we prefer newly generated nodes (i.e. those at deeper levels), and on the same level, left to right.
- (2 pts) Can you come up with an admissible heuristic function so that using it, your A* search will return the goal state $S3$? If yes, give such an admissible heuristic function. If no, please explain why not.

- (2 pts) Can you come up with an admissible heuristic function so that using it, your A* search will return the goal state $S5$ *without using any tie-breaking rule*? If yes, give such an admissible heuristic function. If no, please explain why not.

Solution:

- Notation: node(state,f-value,parent). You can draw a tree as well. Node sequence: $N1(S1, 5, nil), N2(S2, 2, N1), N3(S4, 4, N1), N5(S5, 5, N3)$. (Technically, just a sequence of states is not enough, for example, state S6 will be generated twice, coming from different branches. However, we'll take your answer in sequence of states as well.)
- Impossible as the path to S3 is not an optimal solution.
- Impossible as there are optimal pathes to both S5 and S7, any nodes for them will have the same f-value, so must use a tie-breaking rule to choose one of them.

Question 5: Alpha-Beta Pruning .... 6 points

Consider the following game tree:

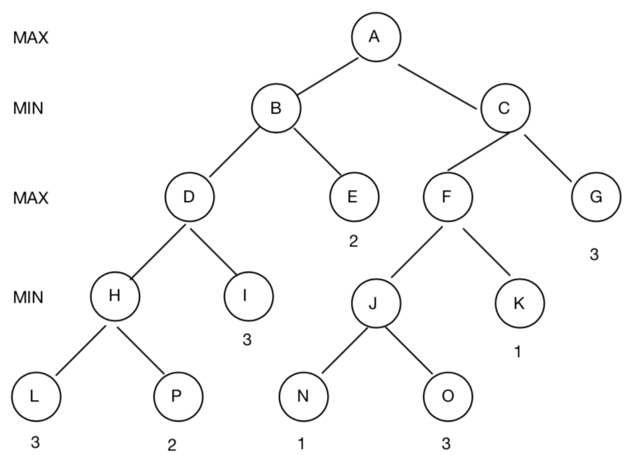


Figure 2: A minimax search tree

- (2 pts) What are the values of nodes A, B and C?
- (4 pts) Perform a left-to-right alpha-beta pruning. Which nodes are pruned? Notice that Left-to-right means that whenever a node is expanded, it's children are considered in the order from left to right. This means the leaf nodes are generated from left to right. So the first leaf node considered is L, followed by P, followed by I and so on.

Solution:

- The values of A,B, and C are 2, 2, 1.
- Nodes O and G are pruned.

Question 6: Game Theory .... **8 points**

There are two bars. Each can choose to set its price for a beer, either \$2, \$4, or \$5. The cost of obtaining and serving the beer can be neglected. It is expected that 6000 beers per month are drunk in a bar by tourists, who choose one of the two bars randomly, and 4000 beers per month are drunk by natives who go to the bar with the lowest price, and split evenly in case both bars offer the same price. What prices would the bars select?

Solve this problem by formalizing the strategic situation as a game in normal form between these two bars and find a solution by computing the pure Nash equilibria.

Solution: The game in normal form (payoffs are average monthly income in thousands):

	\$2	\$4	\$5
\$2	10, 10	14, 12	14, 15
\$4	12, 14	20, 20	28, 15
\$5	15, 14	15, 28	25, 25

For example, if Bar I charges \$2 and Bar II charges \$4, then the monthly income for Bar I is: 3000*2 (from the tourists) + 4000*2 (from the locals) = 14,000. For Bar II, it is 3000*4 = 12,000.

There is a unique Nash equilibrium: (\$4, \$4), so both will charge \$4.

Question 7: Representation in PL .... **5 points**

Suppose we use

- p for "He has a high CGA",
- q for "He took Math3211",
- r for "He will graduate with first-class honor".

Represent the following sentences in propositional logic:

1. He has a high CGA and will graduate with first-class honor.
2. He does not have a high CGA but will still graduate with first-class honor.
3. He has a high CGA because he did not take Math3211.

4. If he has a high CGA, then he did not take Math3211.
5. He either has a high CGA or took Math3211, but not both.

Solution:

1. $p \wedge r$.
2. $\neg p \wedge r$.
3. $p \wedge \neg q$.
4. $p \supset \neg q$.
5. $(p \wedge \neg q) \vee (\neg p \wedge q)$.

Question 8: Representation in FOL . . . . **10 points**

Consider a world with boxes, and the following relations:

- $on(x, y)$: box x is on top of box y.
- $ontable(x)$: box x is on the table.
- $clear(x)$: box x is clear to move.
- $CanMove(x, y, z)$: box x can be moved from y to z.

Represent the following statements in first-order logic:

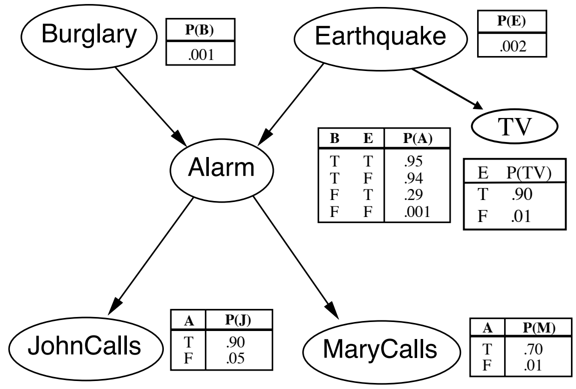
1. (2 pts) A box can have at most one box on top of it.
2. (2 pts) A box is either on top of another box or on the table.
3. (2 pts) A box cannot be on top of two different boxes.
4. (2 pts) A box is clear to move if and only if there is no other box on top of it.
5. (2 pts) A box x can be moved from y to z if and only if x is clear to move, x is on y, and z is clear to move.

Solution:

1. $\forall x, y, z. on(y, x) \wedge on(z, x) \supset y = z$.
2. $\forall x. ontable(x) \vee \exists y. on(x, y)$.
3. $\forall x, y, z. on(x, y) \wedge on(x, z) \supset y = z$.
4. $\forall x. clear(x) \equiv \neg \exists y on(y, x)$.
5. $\forall x, y, z. CanMove(x, y, z) \equiv clear(x) \wedge on(x, y) \wedge clear(z)$.

Question 9: Uncertainty . . . . **10 points**

Consider the following Bayesian network which adds one more node, TV (whether there is a TV report on earthquake), to Pearl's example. There is also a new arc from *Earthquake* to *TV* and the associated conditional probability table:



1. (2 pts) Are $Burglary$ and TV independent given $JohnCalls$? Explain your answer using D-separation.
2. (4 pts) Compute the probability of $Earthquake$ given $Alarm$ is true: $P(E|A)$. There is no need to perform numerical calculations. Your answer can be an expression of numbers like $P(E)P(B)/P(TV|E) = .001 \times .002/.90$.
3. (4 pts) Compute the probability of $Earthquake$ given $Alarm$ is true and TV is not true: $P(E|A, \neg TV)$. Again, there is no need to perform numerical calculations.

Solution:

1. B and TV are not independent given J: there is just one path between them: $B \rightarrow A \leftarrow E \rightarrow TV$, which cannot be separated by A because the given evidence J is a descendant of A, nor by E as it is not in the evidence set.

2.

$$
\begin{aligned}
P(E|A) &= P(E, A)/P(A). \\
P(E, A) &= P(E, A, B) + P(E, A, \neg B) \\
&= P(E)P(B)P(A|E, B) + P(E)P(\neg B)P(A|E, \neg B) \\
&= .002 \times .001 \times .95 + .002 \times (1 - .001) \times .29 \\
P(A) &= P(A, B, E) + P(A, \neg B, E) + P(A, B, \neg E) + P(A, \neg B, \neg E) \\
&= ...
\end{aligned}
$$

3.

$$\begin{aligned}
P(E|A,\neg TV) &= P(E,A,\neg TV)/P(A,\neg TV). \\
P(E,A,\neg TV) &= P(E,A,\neg TV,B) + P(E,A,\neg TV,\neg B) \\
&= P(E)P(B)P(A|E,B)P(\neg TV|E) + \\
&\quad P(E)P(\neg B)P(A|E,\neg B)P(\neg TV|E) \\
&= ... \\
P(A,\neg TV) &= P(A,B,E,\neg TV) + P(A,\neg B,E,\neg TV) + \\
&\quad P(A,B,\neg E,\neg TV) + P(A,\neg B,\neg E,\neg TV) \\
&= ...
\end{aligned}$$

Question 10: MDP .... **10 points**

Consider a 4×3 stochastic grid world laid out in the figure below (the crossed-out cell is an obstacle). The agent starts in state (1,1), and has four available actions: *North*, *South*, *West*, *East*. For each action, the agent goes forward with 0.8 probability, goes left and right with 0.1 probability respectively. If there is a wall, the agent stays at current location. For example, if the agent move *East* in cell (1,3), then she'll end up with 0.8 probability in cell (2,3), 0.1 probability in (1,2) (goes right instead), and 0.1 probability in the same cell (1,3) (goes left, which is a wall). At the terminating states (4,2) and (4,3), the only action is *Exit*. The reward function is defined as follows:

$$R(s,a,s') = R(s') = \begin{cases} -1, & s' = (4,2) \\ +1, & s' = (4,3) \\ 0, & \text{otherwise} \end{cases}$$

Assume that the discount factor $\gamma = 0.9$.

Now consider the initial policy π given in the left grid in the following figure:

3	→	→	→	+1
2	↑		↑	-1
1	→	→	↑	←
	1	2	3	4

3	0.6	0.7	0.9	1.00
2	0.6		0.6	-1.00
1	0.4	0.4	0.5	0.3
	1	2	3	4

Figure 3: A 4x3 grid world and a policy: left is the policy, right its values

We have calculated its value $V_\pi(s)$ in every state s, as shown in the right grid of the above figure. Now for $s_0 = (1,1)$, do the following:

1. Compute $T(s_0, North, s)$ for all s.
2. Compute $Q_\pi(s_0, North)$.

Give the result rounded up to one significant point. Recall the following formula for $Q_\pi(s, a)$:

$$Q_\pi(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_\pi(s')]$$

Solution:

1.

$$T(s_0, North, s') = \begin{cases} 0.8, & s' = (1,2) \\ 0.1, & s' = (1,1) \\ 0.1, & s' = (2,1) \\ 0.0, & \text{otherwise} \end{cases}$$

2.

$$Q_\pi(s_0, North) = \sum_{s'} T(s_0, North, s')[R(s') + \gamma V_\pi(s')]$$

$$=0.8 * 0.9 * 0.6 + 0.1 * 0.9 * 0.4 + 0.1 * 0.9 * 0.4$$

$$=0.5$$

Question 11: Fitness Function .... 3 points

In genetic programming, a fitness function is a mapping from programs to numbers. What are the considerations when designing a good fitness function?

Solution: A good fitness function needs to be accurate in the sense that good programs will have large fitness values. It also needs to be efficient as it needs to be applied to many programs.

Question 12: Linear Features .... 3 points

We know that exclusive or $x_1 \bigoplus x_2$ given by the following truth table is not linear:

x_1	x_2	$x_1 \bigoplus x_2$
1	1	0
1	0	1
0	1	1
0	0	0

Invent some features $f_1, ..., f_k$ so that each feature f_i can be defined linearly from the inputs x_1 and x_2, and the output $x_1 \bigoplus x_2$ can be defined linearly from these features $f_1, ..., f_k$.

Solution: There are many possible solutions. One of them is to use the equivalence:

$$x_1 \bigoplus x_2 \equiv (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$$

to introduce two features $f_1 \equiv x_1 \vee x_2$ and $f_2 \equiv (\neg x_1 \vee \neg x_2)$.

Question 13: Perceptron Learning and GSCA Rule Learning . . . . 18 points

Consider the following data set:

ID	x_1	x_2	x_3	OK
1	0	0	0	Yes
2	0	0	1	No
3	1	0	0	Yes
4	1	1	0	No

where x_1, x_2, and x_3 are some features that should not concern us here.

1. (8 pts) Use these four instances to train a single perceptron using the error-correction procedure. Use the learning rate = 1, and the initial weights all equal to 0. Recall that the threshold is considered to be a new input that always have value "1". Please give your answer by filling in the following table, where weight vector (w_1, w_2, w_3, t) means that w_i is the weight of input x_i, and t is the weight for the new input corresponding to the threshold. Stop when the weight vector converges. If it doesn't converge, explain why not.

ID	Weight vector (w_1, w_2, w_3, t)
Initial	(0, 0, 0, 0)
1	
2	
3	
4	
1	
2	
3	
4	
1	
2	
3	
4	

2. (4 pts) What is the Boolean function corresponding to your perceptron?

3. (6 pts) From the same training set, apply the GSCA algorithm to try to learn a set of rules. Give the set of rules if it succeeds. If it fails to learn a set of rules, explain why it failed.

Solution:

1.

ID	Extended input	Weight vector	Sum	Actual	Desired
Initial		0, 0, 0, 0			
1	0, 0, 0, 1	0, 0, 0, 0	0	1	1
2	0, 0, 1, 1	0, 0, -1, -1	0	1	0
3	1, 0, 0, 1	1, 0 , -1, 0	-1	0	1
4	1, 1, 0, 1	0, -1, -1, -1	1	1	0
1		0, -1, -1, 0	-1	0	1
2		0, -1, -1, 0	-1	0	0
3		0, -1, -1, 0	0	1	1
4		0, -1, -1, 0	-1	0	0

It converges to give the weight vector $(0, -1, -1)$ with the threshold 0.

2. Boolean function: $\neg x_2 \wedge \neg x_3$, i.e. $\overline{x_2} \cdot \overline{x_3}$.

3.
We begin with : $true \supset OK$.
Choose the feature that yields the largest value of r_α:
$r_{x_1} = 0.5, r_{x_2} = 0, r_{x_3} = 0$
So we choose x_1, this will generate : $x_1 \supset OK$
Now consider adding x_2 and x_3: again $r_{x_2} = 0, r_{x_3} = 0$. We can add either of them, say x_2: $x_1 \wedge x_2 \supset OK$. But there are no positive examples covered by this rule. So we can backtrack to x_3: $x_1 \wedge x_3 \supset OK$. But now there are no examples, positive or negative, covered by this rule. So we backtrack to $x_1 \supset OK$ and replace it by either $x_2 \supset OK$ or $x_3 \supset OK$, which give us exactly the same problem. So it backtrack to the beginning and abort (or loop for ever). So GSCA does not work, for the reason that this training set cannot be learned without making use of negative literals like $\neg x_2$ in the rules.