

# CS325 Artificial Intelligence – Spring 2013

## Midterm *Solution Guide*

Instructor: Cengiz Gunay, Ph.D.

March 19, 2013

### Instructions

Total points: 50 (25% of course grade, subject to change)

1. Use your own empty sheets to write your answers. I left a lot of free space on this handout so you can use it for scribbling.
2. On the top of your sheets, write your name and name of students sitting next to you.
3. You don't need to repeat the questions on your answer sheets.
4. There are some formulas at the end on page 11. They are referenced from questions that need them.

( $n$ ★) shows popularity of question –  $n$  students chose it.

### I. Verbal Questions

(10 points) Pick and answer ANY 5 of the following 12 questions.

1. (9★) How do you get started if you wanted make an intelligent agent to solve a problem? What are the first things you need for building an agent?

Answer: Must define PEAS: Performance, Environment, Actuators, and Sensors.

2. (10★) List some environmental properties that are important for intelligent agents. Describe the properties of the environment for the Mars Rover.

Answer: Fully/partially observable, discrete/continuous, single/multi agent, stochastic/deterministic, adversarial, dynamic/static. Mars rover: partially observable (because sensors are limited), single agent, continuous, stochastic (e.g., weather events), dynamic (slips?), non-adversarial.

3. (18★) Compare breadth-first, depth-first, cheapest-first and A\* algorithm strategies and benefits.

Answer: Breadth-first: siblings first, then go deeper. Complete only if breadth,  $b$ , is finite. Non-optimal if cost not unitary. Time and space complexity are both  $O(b^d)$ . Depth-first: go to deeper nodes first, then go to next branch. Complete only if depth,  $d$ , is finite and there are no loops. Non-optimal as in breadth-first. Space complexity is  $O(bd)$ , which is smaller than breadth-first.

Cheapest-first: Like breadth first, but minimizes cost of path taken so far. Complete and optimal if  $b$  is finite.

A\*: Adds heuristic function in addition to cost. Complete and optimal if heuristic is admissible.

4. (8★)What is the difference between conditional and joint probabilities between two random variables?  
 Answer: Conditional probability,  $P(A|B)$ , is the probability of  $A$  when we know that  $B$  happened. Joint probability,  $P(A, B)$ , is the probability of both events happening. Only if the two are independent,  $A \perp B$ , then  $P(A, B) = P(A)P(B)$ .
5. (10★)What type of questions can you answer with a Bayes Net?  
 Answer: When many factors contribute in probabilistic fashion to an outcome. We can both make predictions about the outcome and diagnose causes.
6. (13★)When can you employ supervised learning to solve a problem?  
 Answer: When you want to find the relationship between some input data and desired outcomes. For instance classification based on multiple parameters.
7. (17★)What is unsupervised learning good for in real life problems?  
 Answer: Finding hidden structure and regularities in data. Practically used for clustering, dimensionality reduction, etc.
8. (4★)What is the advantage of using logic over other methods of representation?  
 Answer: Logic not only provides a concise description of knowledge and situation of an agent, but also allows use of automated reasoning.
9. (14★)Why do we need to alternate between plan and execution?  
 Answer: Making long-term plans may fail due to several reasons. In partially observable, uncertain and stochastic environments we need to plan for the goal, but interleave the execution of the plan with observations to make sure we are on the right track.
10. (8★)Why do we need a belief state for planning?  
 Answer: Again, in partially observable, uncertain and stochastic environments, the agent can be in one of several states, which is why it needs a belief state during planning to consider all options.
11. (2★)What's the difference between Markov Decision Processes (MDPs) and Reinforcement Learning (RL)?  
 Answer: MDPs can find the way to known reward locations in a fully observable environment. In contrast, RL can operate in a partially-observable environment to discover reward locations.
12. (1★)What is the dilemma between exploration and exploitation for an RL agent?  
 Answer: In a partially-observable environment, the RL agent needs to explore to find the rewards, however it should also know when to stop exploring and pursuing the rewards. Otherwise, it may waste its resources during unnecessary exploration.

## II. Simple Knowledge Questions

(20 points) Pick and answer ANY 2 of the following questions. Note that some questions are slightly different than their homework versions.

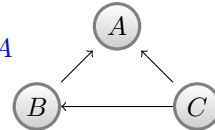
1. (1★)(10 points) Describe a single-state agent architecture for the 3-location vacuum cleaner. Discuss:

- (a) Size of state space  
Answer: Agent's possible location (3)  $\times$  possible dirt configurations ( $2^3$ ) =  $3 \times 2^3 = 24$  states.
- (b) Initial configuration and goal test  
Answer: Choose one of the 24 states as described above as initial state. Goal states are the ones without any dirt in any of the locations, irrespective of where the agent is located.
- (c) Transition function  
Answer: Agent can move left or right, which affects its location. Agent can suck dirt, which affects existence of dirt.
- (d) Adequate tree-search algorithm and its complexity for width  $n$  and depth  $d$   
Answer: Without any costs defined, we can use breadth-first search with time and space complexity of  $O(n^d)$ .

2. (22★)(HW2 – 10 points) Briefly describe what each of the following means:

- (a)  $P(A, B)$   
Answer: Probability of both  $A$  and  $B$  happening.
- (b)  $P(A|B, C)$   
Answer: Probability of  $A$  given that we know  $B$  and  $C$  to be true or false.
- (c)  $A \perp B | C$   
Answer:  $A$  is independent of  $B$  conditional on knowing outcome of  $C$ .
- (d)  $P(A|B, C) = P(A|B)$   
Answer:  $A$  is independent of  $C$  conditional on knowing outcome of  $B$ . We do not know if  $A$  is independent of  $C$  in general.
- (e)  $P(A, B, C) = P(A|B, C)P(B|C)P(C)$   
(explain this and also draw it as a Bayes Net)

Answer:  $C$  is independent;  $B$  depends on  $C$ ; and,  $A$  depends on both  $B$  and  $C$ .



3. (4★)(HW3 – 10 points) Briefly describe what each of the following means. Explain whether it is an unsupervised or supervised learning scheme, its advantages and disadvantages, and indicate whether it employs a hill climbing (or gradient descent) method, which is subject to local minima.

- (a)  $k$ -Means Clustering  
Answer: Unsupervised iterative algorithm for finding cluster centers. Not a gradient-descent method (no derivative), but suffers from local minima.
- (b) Expectation-Maximization using Maximum Likelihood  
Answer: Unsupervised iterative algorithm to maximize probability of a Bayes Net. Can be used for maximizing most likely Gaussian representation of clusters. No gradient, has local minima, but works better than  $k$ -means.
- (c) Multi-layer Perceptron (MLP)  
Answer: Supervised learning algorithm with multiple layers of neural units (perceptrons). They are used for classification and pattern recognition problems. It suffers from local minima because of the gradient descent method following the derivative of the loss function.

(d) Support Vector Machines (SVM)

Answer: Supervised learning algorithm with transformation such that it converges faster. While still suffering from local minima, it performs much better than MLPs.

(e) Genetic/Evolutionary Algorithms

Answer: Supervised learning algorithm with no gradient descent. It can find global minimum, but takes longer to find solution.

4. (20★)(HW4 – 10 points) Select and write one set of true/false values for the variables  $A, B, C, D$  and then answer whether the following logic sentences are true or false. Also indicate if the sentence is valid, satisfiable or unsatisfiable.

Answer: For  $A = t, B = t, C = t, D = t$ ;

(a)  $(A \wedge B) \vee (\neg C \vee \neg D)$

Answer: True, satisfiable.

(b)  $A \Rightarrow B$

Answer: True, satisfiable.

(c)  $A \wedge C \Rightarrow B \vee D$

Answer: True, satisfiable.

(d)  $(\neg B \Rightarrow \neg A) \Leftrightarrow (\neg A \vee B)$

Answer: True, valid.

(e)  $(B \Rightarrow \neg A) \Leftrightarrow (\neg C \wedge D)$

Answer: True, satisfiable.

### III. Problem Solving Questions

(20 points) Pick and choose questions that add up to 20 points. You are allowed +10 bonus points if you do more, which will count only WITHIN the points of this exam. That is, you cannot get more than 50 points from this exam.

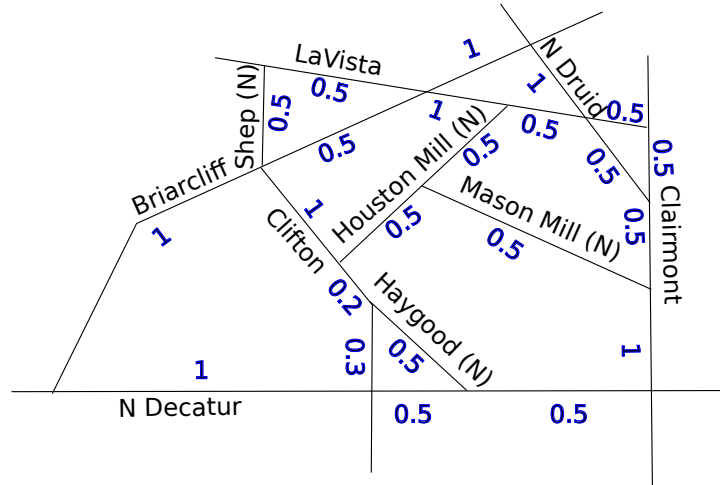


Figure 1: A road map of the Emory vicinity. Numbers show approximate distance in miles. See Question 1 for details.

- (19★)(HW1 – 10 points) Starting from the Clairmont Rd. and North Druid Hills intersection on map in Figure 1, manually simulate the FIRST THREE ITERATIONS of the uniform-cost (cheapest-first) algorithm to target one of the following destinations:
  - Clifton & Houston Mill
  - Clifton & Haygood
  - N Decatur & Clifton

Write down what you add and remove from the frontier at each step and how you select which intersection to choose next.

Answer: Traverse the map and manipulate the frontier:

- Pop (Clairmont & North Druid Hills, 0)  
Add (Clairmont & Mason Mill, 0.5), (North Druid Hills & La Vista, 0.5), (Clairmont & La Vista, 0.5)  
All the same, choose one randomly:
- Pop (Clairmont & Mason Mill, 0.5)  
Add (Houston Mill & Mason Mill, 1), (Clairmont & North Decatur, 1.5)
- Pop (North Druid Hills & La Vista, 0.5)  
Add (La Vista & Houston Mill, 1), (North Druid Hills & Briarcliff, 1.5)

(Solution for all destinations are the same for the first three iterations.)

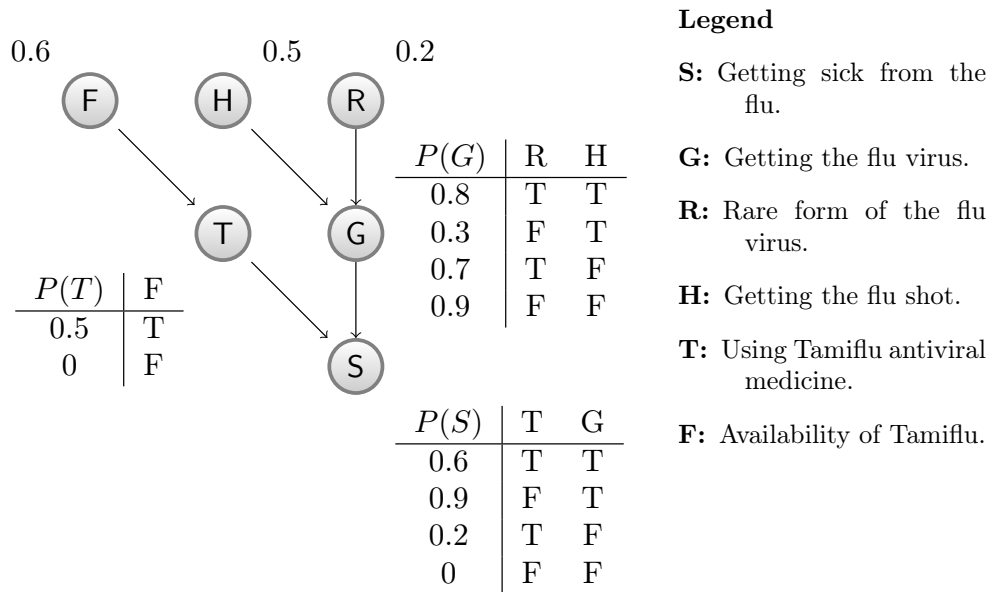


Figure 2: A fake flu Bayes net for catching the flu. See text for details.

2. (16★)(HW2 – 10 points) Based on the Bayes net in Figure 2, solve the probability formula (by explaining each step) described by the following verbal question:

”What’s the probability of getting sick from with the rare form of the flu if you knew that Tamiflu was not available?”

Answer: I also gave points for a second interpretation for the verbal question:

- (a)  $P(S|\neg F, R)$ , which is the probability of getting sick given that we know it’s the rare form and Tamiflu was unavailable.

- i. Eliminate  $F$  completely because  $P(T|\neg F)$  is impossible:

$$P(S|\neg F, R) = P(S|\neg T, R).$$

- ii. Eliminate  $H$  by enumerating  $P(G|R)$ :

$$\begin{aligned} P(G|R) &= P(G|R, H)P(H) + P(G|R, \neg H)P(\neg H) \\ &= 0.5 \times 0.8 + 0.7 \times 0.5 = 0.75. \end{aligned}$$

- iii. Use the chain rule and enumeration over possible states of  $G$ :

$$\begin{aligned} P(S|\neg T, R) &= \sum_g P(S|\neg T, g)P(g|R) \\ &= 0.9 \times 0.75 = 0.675. \end{aligned}$$

- (b)  $P(S, R|\neg F)$  is exactly what I asked.

- i. Again, start by eliminating  $F$ :

$$P(S, R|\neg F) = P(S, R|\neg T).$$

ii. Enumerate the full joint probability:

$$\begin{aligned} P(S, R|\neg T) &= \sum_{g,h} P(S, R, \neg T, h, g) \\ &= \sum_{g,h} P(S|\neg T, g)P(g|R, h)P(h)P(R) \end{aligned}$$

reuse  $P(G|R)$  from above:

$$\begin{aligned} &= \sum_g P(S|\neg T, g)P(G|R)P(R) \\ &= 0.9 \times 0.75 \times 0.2 = 0.134. \end{aligned}$$

because  $P(S|\neg T, \neg G) = 0$ .

Shopper	Milk	Bread	Beer	Soda	Water	Eggs	Price	Uses coupons?
$S_0$	0	0	1	1	0	0	\$10	1
$S_1$	1	0	0	1	1	1	\$42	0
$S_2$	1	1	1	0	1	1	\$12	1
$S_3$	0	1	1	1	1	0	\$35	0
$S_4$	0	1	0	0	1	1	\$5	1

Table 1: Shopping data from shoppers  $S_0, \dots, S_4$ .

3. (5★)(HW3 – 10 points) Simulate the  $k$ -Means clustering algorithm (see page 11) for two iterations to find two clusters ( $k = 2$ ) in the data of Table 1. Do not use the “Shopper,” “Price,” or “Uses coupons?” columns, and start with initial cluster centers of respectively all zeros and all ones.

Answer:

	$C_1 = (1, 1, 1, 1, 1, 1)$	$C_2 = (0, 0, 0, 0, 0, 0)$
(a) Iteration 1:	$S_1$ $S_2$ $S_3$	$S_0$ $S_4$ (either into $C_1$ or $C_2$ )
	$C_1 = (\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 1, \frac{2}{3})$	$C_2 = (0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$
(b) Iteration 2:	$S_1$ (2.3) $S_2$ (2) $S_3$ (2)	$S_0$ (2.5) $S_4$ (2.5)

4. (2★)(HW3 – 10 points) Now consider all columns in Table 1 to predict which shoppers have a larger tendency to use coupons. Use the linear perceptron definition on page 11.

- (a) Start with a weight vector,  $w = [3.0, 6.5, 8.0, 1.0, 5.0, 7.0, -3.0]$  and threshold,  $T = 9.3$ . Evaluate any three rows in table and report the weighted sum of the inputs and the perceptron output, and compare it to the expected output. Look into the weights and discuss which column(s) were most influential in the perceptron’s right and wrong decisions.

Answer:

Example, evaluate:

Multiply by weights	Guess
$S_0 \cdot w = 0 + 0 + 8 + 1 + 0 + 0 - 30 = -21 > 9.3?$	0 Wrong
$S_1 \cdot w = 3 + 0 + 0 + 1 + 5 + 7 - 126 = -110 > 9.3?$	0 Correct
$S_2 \cdot w = 3 + 6.5 + 8 + 0 + 5 + 7 - 36 = -6.5 > 9.3?$	0 Wrong

Price column is the most influential by far. Other weights need to be trained more to overcome it.

- (b) Then, choose one row from the table and use its values to update the perceptron weight vector. Use a training rate parameter of  $\alpha = 0.5$  to update the weights. What changed?

Answer: Use  $S_0$  inputs, only non-zero inputs will cause a change:

Weight update	Change?
$w_2 = 8 + 0.5 \times (1 - 0) \times 1 = 8.5$	increase
$w_3 = 1 + 0.5 \times (1 - 0) \times 1 = 1.5$	increase
$w_6 = -3 + 0.5 \times (1 - 0) \times 10 = 2$	increase

Large increase in price column as expected from 4a.



<p>i. <math>\forall a, b \text{ Loves}(a, b) \wedge \text{Loves}(b, a) \Rightarrow \text{Happy}(a) \wedge \text{Happy}(b)</math></p> <p>ii. <math>\forall a, b, c \text{ Loves}(a, b) \wedge \text{Loves}(b, c) \wedge \neg(a = c) \Rightarrow \text{Unrequited}(a, b) \wedge \text{Sad}(a)</math></p> <p>iii. <math>\exists a \text{ Happy}(a)</math></p> <p>iv. <math>\exists a \text{ Sad}(a)</math></p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 2: A knowledge base.

5. (13★)(HW4 – 10 points) To the knowledge base (KB) given in Table 2, add THREE of the following sentences and then answer if the query “Unrequited(*John*, *Nancy*)?” is true, false, or unknown. Explain your logic. Make sure that the KB is consistent; that is, it includes no contradicting knowledge.
- (a) Loves(*John*, *Mary*)
  - (b) Loves(*Mary*, *John*)
  - (c) Loves(*John*, *Nancy*)
  - (d) Loves(*Nancy*, *Fred*)
  - (e) Loves(*Fred*, *Emma*)

Answer: There are a few solutions to this problem.

- (a) A full solution can be given by selecting: (a), (b), and (c).  
According to i.

$$\text{Loves}(\textit{John}, \textit{Mary}) \wedge \text{Loves}(\textit{Mary}, \textit{John}) \Rightarrow \text{Happy}(\textit{John}) \wedge \text{Happy}(\textit{Mary})$$

According to ii.

$$\text{Loves}(\textit{Mary}, \textit{John}) \wedge \text{Loves}(\textit{John}, \textit{Nancy}) \wedge \neg(\textit{Mary} = \textit{Nancy}) \Rightarrow \text{Unrequited}(\textit{Mary}, \textit{John}) \wedge \text{Sad}(\textit{Mary})$$

iii. and iv. are also satisfied because Sad(*Mary*) and Happy(*Mary*). Although unintuitive, the KB is consistent because it doesn't say same person cannot be both happy and sad.

- (b) Other selections can lead to satisfying iii. or iv. but not both. As long as it was unknown, I accepted it as consistent.

**Game objective:**

Swap the places of left arrow tiles with right arrow tiles.

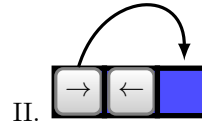
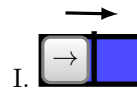
**Board:****Rules:**

Figure 3: A simple tile-moving game. You need to reach the objective by only using the two rules. First rule says a tile can move in the direction of the arrow if its immediate neighbor is a blank. The second rule says a tile can jump in its arrow direction over a tile that points the reverse direction if the next tile is blank. Game credits: Machinarium.

6. (4★)(HW4 – 10 points) Consider the tile-moving game in Figure 3. Write the *action schema* for the game in classical planning (see page 11). Define the game board, the tiles, and then write the two rules. Would a depth-first or breadth-first algorithm work better for this task? Is it more advantageous to start from the initial state or the goal state? Based on your answer to these questions, manually simulate the first three steps of the solution by drawing the states from the initial board condition to match and execute the rules by implementing the graph search.

Answer: Action schemas don't use first-order logic, just simple parameterization of variables in propositional logic. Define some predicates:

- (a)  $\text{Location}(L1) \dots \text{Location}(L7)$ : positions on board
- (b)  $\text{Empty}(L4)$ : position being empty
- (c)  $\text{LeftOf}(L4, L5)$ : ordering
- (d)  $\text{TileLeft}(TL1)$  &  $\text{TileRight}(TR3)$ : tile with left/right arrow
- (e)  $\text{On}(L5, TL1)$ : position of tile

Then, use them to define actions:

- (a) Action:  $\text{MoveR}(t, l, b)$   
 Pre:  $\text{Location}(l) \wedge \text{Location}(b) \wedge \text{TileRight}(t) \wedge \text{On}(l, t) \wedge \text{LeftOf}(l, b) \wedge \text{Empty}(b)$   
 Eff:  $\text{Empty}(l) \wedge \text{On}(b, t) \wedge \neg \text{On}(l, t)$
- (b) ...

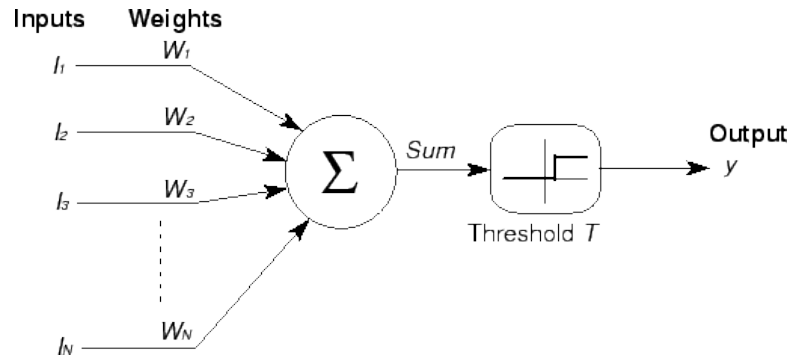
And you're done!

## Appendix

### 1. $k$ -Means clustering algorithm

- (a) Randomly place  $k$  cluster centers
- (b) Assign each point to closest center
- (c) Move each cluster to center of gravity of new set
- (d) Go back to step (b) until no change

### 2. Perceptron equations



$$y = f_{\mathbf{w}}(\mathbf{I})$$

$$\text{sum} = \sum_{i=1}^N I_i W_i$$

$$y = \begin{cases} 1, & \text{if sum} \geq T \\ 0, & \text{if sum} < T \end{cases}$$

The perceptron update rule for sample  $k$  and input  $j$ :

$$w_j \leftarrow w_j + \alpha(y_k - f_{\mathbf{w}}(\mathbf{I}_k)) \times I_{j,k}$$

### 3. Classical planning action schema consists of:

- (a) Action
- (b) Precondition
- (c) Effect

Example:

Action(Fly( $p$ ,  $from$ ,  $to$ ),

Pre:  $At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$

Eff:  $\neg At(p, from) \wedge At(p, to)$