15-281	Artificial Intelligence	HW8

INSTRUCTIONS

- **Due: Tuesday, 29 October 2019 at 10:00 PM EDT.** Remember that you have NO slip days for Written Homework, but you may turn it in up to 24 hours late with 50% Penalty.
- **Format:** Submit the answer sheet pdf containing your answers. You should solve the questions on this handout (either through a pdf annotator, or by printing, then scanning). Make sure that your answers (typed or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.
- **How to submit:** Submit a pdf with your answers on Gradescope. Log in and click on our class 15-281 and click on the submission titled HW8 and upload your pdf containing your answers.
- **Policy:** See the course website for homework policies and Academic Integrity.

Name	
Andrew ID	
Hours to complete?	

For staff use only

Q1	Q2	Q3	Q4	Q5	Total
/13	/10	/22	/30	/25	/100

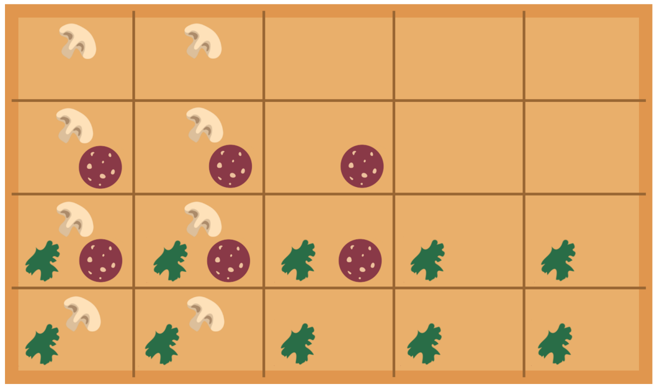
Q1. [13 pts] Probability: Product Rule and Chain Rule

Omega Pizzeria

As we step through the probability concepts of product rule and chain rule, you may find it helpful to check your understanding by referring back to our Omega Pizzeria example.

Let's define the following binary random variables and outcomes:

- X_1: Spinach random variable
  - $-x_1$: no spinach outcome
  - $+x_1$: spinach outcome
- X_2: Mushroom random variable
  - $-x_2$: no mushrooms outcome
  - $+x_2$: mushrooms outcome
- X_3: Pepperoni random variable
  - $-x_3$: no pepperoni outcome
  - $+x_3$: pepperoni outcome



Product Rule

Sometimes you have conditional and marginal distributions, but you actually want to compute the full joint distribution. We know the basic product rule:

$$P(X_1, X_2) = P(X_1 \mid X_2)P(X_2) \tag{1}$$

and

$$P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) \tag{2}$$

We can generalize the product rule a bit more when there are more than two variables. The following are two valid ways to break up the joint distribution of three variables using a more general application of product rule:

$$P(X_1, X_2, X_3) = P(X_1, X_2 \mid X_3)P(X_3) \tag{3}$$

and

$$P(X_1, X_2, X_3) = P(X_1 \mid X_2, X_3)P(X_2, X_3) \tag{4}$$

You can check the math with the pizzeria to make sure that all three of these sentences are equal:

- The probability of getting a slice with spinach, mushrooms, and pepperoni.
- (The probability of getting a slice with spinach and mushrooms, given that we asked for a slice with pepperoni) times (the probability of getting a slice with pepperoni)
- (The probability of getting a slice with spinach, given that we asked for a slice with mushrooms and pepperoni) times (the probability of getting a slice with mushrooms and pepperoni)

Chain Rule

We can further break down equation 4 by applying the product rule again, $P(X_2, X_3) = P(X_2 \mid X_3)P(X_3)$:

$$\begin{aligned} P(X_1, X_2, X_3) &= P(X_1 \mid X_2, X_3)P(X_2, X_3) && (4) \\ &= P(X_1 \mid X_2, X_3)P(X_2 \mid X_3)P(X_3) && (5) \end{aligned}$$

This brings us to the general chain rule for N random variables, $X_1, X_2, \ldots, X_N$:

$$P(X_1, X_2, \ldots, X_N) = \prod_{n=1}^{N} P(X_n \mid X_1, \ldots, X_{n-1}) \tag{6}$$

Mutton, Lettuce, and Tomato

"True love is the greatest thing in the world...except a nice MLT: mutton, lettuce, and tomato sandwich, where the mutton is nice and lean..." –*Miracle Max*

You are given the following probability tables for binary random variables M, L, T:

T	$P(T)$
$+t$	0.4
$-t$	0.6

T	L	$P(L \mid T)$
$+t$	$+l$	0.8
$+t$	$-l$	0.2
$-t$	$+l$	0.25
$-t$	$-l$	0.75

L	T	M	$P(M \mid L, T)$
$+l$	$+t$	$+m$	0.95
$+l$	$+t$	$-m$	0.05
$+l$	$-t$	$+m$	0.75
$+l$	$-t$	$-m$	0.25
$-l$	$+t$	$+m$	0.40
$-l$	$+t$	$-m$	0.60
$-l$	$-t$	$+m$	0.10
$-l$	$-t$	$-m$	0.90

(a) [8 pts] Calculate $P(L, T, M)$ from the tables given.

L	T	M	$P(L, T, M)$
$+l$	$+t$	$+m$	
$+l$	$+t$	$-m$	
$+l$	$-t$	$+m$	
$+l$	$-t$	$-m$	
$-l$	$+t$	$+m$	
$-l$	$+t$	$-m$	
$-l$	$-t$	$+m$	
$-l$	$-t$	$-m$	

(b) [5 pts] Which of the following are valid decompositions of the joint probability distribution of M, L, and T, given no assumptions about the relationship between these random variables? Select all that apply.

- ☐ i) $P(M)P(L)P(T)$
- ☐ ii) $P(M)P(M, L)P(M, L, T)$
- ☐ iii) $P(M, L \mid T)P(T)$
- ☐ iv) $P(M \mid L, T)P(L)$
- ☐ v) $P(M \mid L, T)P(T)$
- ☐ vi) None of the above

Q2. [10 pts] Probability: Chain Rule, Joint Distributions, and Marginalization

Recall *marginalization* from HW7, which means summing out unwanted variables from a joint distribution. Consider three binary random variables A, B, and C with domains $\{+a, -a\}$, $\{+b, -b\}$, and $\{+c, -c\}$, respectively. Remember that $P(A)$ refers to the table of probabilities of all the elements of A's domain.

(a) [2 pts] Express $P(A)$ in terms of the joint distribution $P(A, B, C)$. Your answer should contain summation notation.

$P(A) =$

(b) [2 pts] Express $P(A)$ in terms of $P(C)$, $P(B \mid C)$ and $P(A \mid B, C)$. Your answer should contain summation notation.

$P(A) =$

(c) [6 pts] Expand the sums from part (b) to express the two elements of $P(A)$ ($P(+a)$ and $P(-a)$) in terms of the individual probabilities (e.g. $P(+b \mid +c)$, $P(-c)$ instead of tables $P(B \mid C)$ or $P(C)$). Your answer should *NOT* contain summation notation.

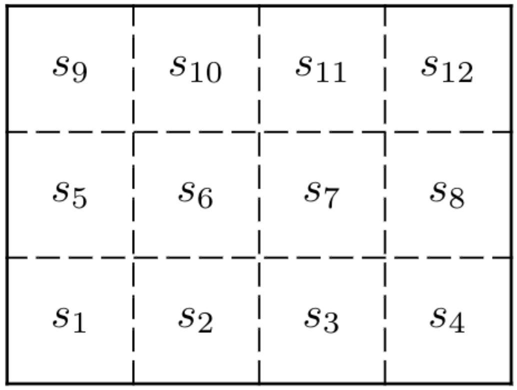
$P(+a) =$

$P(-a) =$

Q3. [22 pts] Dark Room Q-learning

A robot mysteriously finds itself in a dark room with 12 states as shown below. The robot can take four actions at each state (North, East, South, and West). We do not know where the interior walls are, except for the walls surrounding the room (represented by solid lines).

An action against a wall leaves the robot in the same state. Otherwise, the outcome of an action deterministically moves the robot in the direction corresponding to the action.



The robot learns the Q-values below. Note that only the maximum values for each state are shown, as the other values (represented with '-') do not affect the final policy.

	N	E	S	W
s_1	59	59	-	-
s_2	-	66	-	-
s_3	-	73	-	-
s_4	81	-	-	-
s_5	66	-	-	-
s_6	-	59	-	59
s_7	-	-	66	-
s_8	90	-	-	-
s_9	-	73	-	-
s_{10}	-	81	-	-
s_{11}	-	90	-	-
s_{12}	100	100	-	-

(a) [4 pts]

Using the learned Q-values, what are the first six actions the robot takes if it starts in state s_7? If there is more than one best action available, choose one randomly. For now, ignore any potential interior walls.

(b) [9 pts]

We are told the discount factor used during Q-learning was $\gamma = 0.9$.
We are also told there exists a single state, s^* such that $R(s^*, a, s') > 0 \ \forall a, s'$, and for all other states s, $R(s, a, s') = 0 \ \forall a, s'$. Also assume that $R(s^*, a, s')$ are equivalent for all a, s'.

(i) What is s^*, and what is the reward?

s^***:**

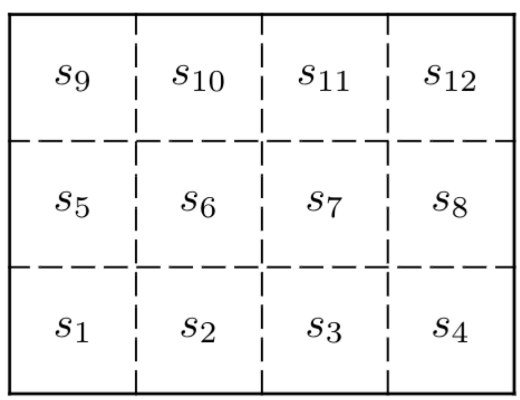
$R(s^*, a, s')$**:**

(ii) Explain how you found your answers.

(c) [9 pts]

Now the robot wants to locate the interior walls within the room.

(i) Where are these walls? Draw them in by filling the corresponding dashed lines below.

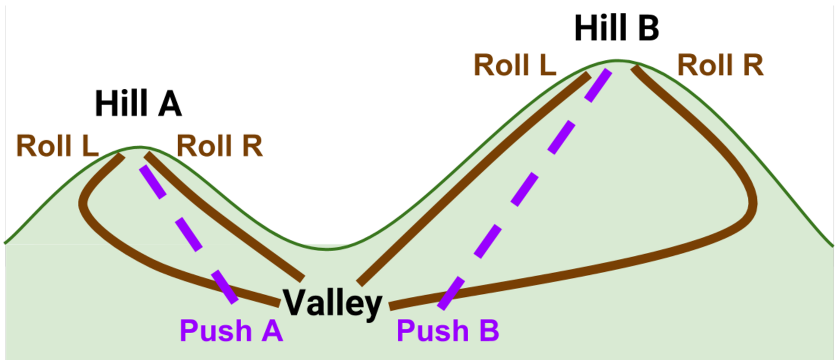


(ii) In 1-2 sentences, explain how you know where the walls are.

Q4. [30 pts] Representation for Buggy Bots

In this problem, we explore how domain representation affects performance and computational time when learning policies for MDPs.

In preparation for Carnival, we are training bots to push our buggy! The buggy bots move between three states: $Valley$, $HillA$, and $HillB$. From the valley, bots can choose to push up to Hill A or push up to Hill B. From the top of either hill, the available actions are to roll left or roll right.



The true transition matrix and reward functions are shown in the table below. At both $HillA$ and $HillB$, rolling right gives positive reward and rolling left gives negative reward, but the positive reward in $HillB$ is slightly higher. In $Valley$, both $PushA$ and $PushB$ give negative reward, but $PushA$ gives a less negative reward.
Note that from $HillB$, both $RollL$ and $RollR$ can result in the buggy ending up on $HillA$.

s	a	s'	$T(s,a,s')$	$R(s,a)$
$HillA$	$RollL$	$Valley$	1.0	-2.0
$HillA$	$RollR$	$Valley$	1.0	0.7
$HillB$	$RollL$	$Valley$	0.8	-2.3
$HillB$	$RollL$	$HillA$	0.2	-2.3
$HillB$	$RollR$	$Valley$	0.9	1.5
$HillB$	$RollR$	$HillA$	0.1	1.5
$Valley$	$PushA$	$HillA$	1.0	-0.2
$Valley$	$PushB$	$HillB$	1.0	-0.9

We consider two representations of the domain by two different robots. Robot One has a sensor that tells it whether it is in state $HillA$, $HillB$, or $Valley$ (allowing it to fully represent the true state). Robot Two has a simpler representation, having only a hill/valley sensor, which allows it to distinguish between $Valley$ and $HillA$ or $HillB$, but it can't distinguish between $HillA$ and $HillB$.

(a) [8 pts] Using a discount factor of $\gamma = 0.9$, Robot One learns the following Q-values:

s	a	$Q(s,a)$
$HillA$	$RollL$	0.302
$HillA$	$RollR$	3.00
$HillB$	$RollL$	0.082
$HillB$	$RollR$	3.84
$Valley$	$PushA$	2.51
$Valley$	$PushB$	2.56

(i) What is the optimal policy for Robot One?

$HillA$: ○ $RollL$ ○ $RollR$

$HillB$: ○ $RollL$ ○ $RollR$

$Valley$: ○ $PushA$ ○ $PushB$

(ii) Using $\gamma = 0.9$, what is the value of this policy? Specify the values with three significant digits. *Note*: You may want to code this up to quickly compute these.

$V^{\pi_1}(HillA)$:	$V^{\pi_1}(HillB)$:	$V^{\pi_1}(Valley)$:

(b) [11 pts] Robot Two uses a uniform exploration strategy while learning, so the observed rewards and transition probabilities in the *Hill* state are the average of the rewards and transitions in *HillA* and *HillB*, as each are visited roughly equally. Under this exploration strategy, with a discount factor of $\gamma = 0.9$, Robot Two learns the following Q-values:

s	a	$Q(s,a)$
Hill	*RollL*	1.79
Hill	*RollR*	5.01
Valley	*PushA*	4.31
Valley	*PushB*	3.61

(i) What is the optimal policy for Robot Two?

Hill: ◯ *RollL* ◯ *RollR*

Valley: ◯ *PushA* ◯ *PushB*

(ii) How is this policy represented in the original domain, i.e. how would Robot One represent this policy?

Answer:

(iii) What is the value of this policy in the original domain, i.e. what would the value of this policy be if it were evaluated by Robot One. Specify the values with three significant digits. *Note*: You may want to code this up to quickly compute these.

$V^{\pi_2}(HillA)$:	$V^{\pi_2}(HillB)$:	$V^{\pi_2}(Valley)$:

(c) [5 pts] Based on the results from Robot One and Robot Two, answer the following questions:

(i) Which of the policies performs better on the original domain?

◯ Robot One ◯ Robot Two

(ii) Can the optimal policy for the original domain be expressed in both domains? Why or why not?

Answer:

(d) [3 pts] Consider a generic MDP in which all actions are deterministic, and all states can be reached from any given state.

Suppose we have two representations of this MDP: R1, which has m states, and R2, which has n states. Assume that one iteration of value iteration using R1 takes t seconds. How long, in terms of t, should we expect one iteration of value iteration to take for R2?

Robot Two time:

(e) [3 pts] What are the trade-offs in choosing the representations for problems with large state spaces?

Answer:

Q5. [25 pts] Approximate Q-learning

A robot is trying to get to its office hours, occurring on floors 3, 4, or 5 in GHC. It is running a bit late and there are a lot of students waiting for it. There are three ways it can travel between floors in Gates: the stairs, the elevator, and the helix.

The **state** of the robot is the floor that it is currently on (either 3, 4, or 5).

The **actions** that the robot can take are *stairs*, *elevator*, or *helix*.

In this problem, we are using a linear, feature based approximation of the Q-values:

$$Q_w(s,a) = \sum_{i=0}^{3} f_i(s,a) w_i$$

We define the feature functions as follows:

Features	**Initial Weights**
$f_0(s,a) = 1$ (this is a bias feature that is always 1)	$w_0 = 1$
$f_1 = f_{\text{speed}}(s,a) = (\lvert s-4 \rvert + 1)t$, where $t = \begin{cases} 50, a = \text{elevator} \\ 20, a = \text{stairs} \\ 10, a = \text{helix} \end{cases}$	$w_1 = 0.5$
$f_2 = f_{\text{accessibility}}(s,a) = \begin{cases} 2, a = \text{stairs} \\ 5, a = \text{helix} \\ 10, a = \text{elevator} \end{cases}$	$w_2 = 2$
$f_3 = f_{\text{emptiness}}(s,a) = \begin{cases} 60, a = \text{elevator}, s = 3 \\ 80, a = \text{elevator}, s = 4 \\ 60, a = \text{elevator}, s = 5 \\ 0, \text{otherwise} \end{cases}$	$w_3 = 0.1$

Furthermore, the weights will be updated as follows:

$$w_i \leftarrow w_i + \alpha[r + \gamma \max_{a'} Q_w(s',a') - Q_w(s,a)] \frac{\delta}{\delta w_i} Q_w(s,a)$$

(a) [9 pts] Calculate the following initial Q values given the initial weights above.

Q_w**(4, elevator):**	Q_w**(4, stairs):**	Q_w**(4, helix):**

(b) [3 pts] For this question, suppose that the initial Q-values for state 3 happen to be equal to the corresponding initial Q-values for state 5.

(i) In this problem, as you update the weights, will these values remain equal? I.e., will $Q_w(3, a) = Q_w(5, a)$ given any action a and vector w?

◯ Yes ◯ No

(ii) Why or why not?

Answer:

(c) [3 pts] Given the Q-values for state 4 calculated in part (a), what are the probabilities that each of the following actions could be chosen when using ϵ-greedy exploration from state 4 (assume random movements are chosen uniformly from all actions)? Write your answers in terms of ϵ.

elevator:	**stairs:**	**helix:**

(d) [8 pts] Given a sample with start state 3, action = stairs, successor state = 4, and reward = -2, update each of the weights using learning rate $\alpha = 0.25$ and discount factor $\gamma = 0.6$.

$w_0 =$

$w_1 =$

$w_2 =$

$w_3 =$

(e) [2 pts] What is an advantage of using approximate Q-learning instead of the standard Q-learning? What is a disadvantage?

Advantage:

Disadvantage: