Name:
Student ID:

CSE P 573 Winter 2022: HW 2

Due 3/6/2022
Total: 100 points

Instructions:

1) The homework should be done individually. Don't forget to write your name.
2) We highly recommend typing your homework, but writing and scanning, or annotating a PDF are also acceptable.
3) Keep your answers brief but provide enough explanations and details to let us know that you have understood the topic.
4) The assignment is due on March 6.
5) You should upload your assignments through gradescope

Topics:	Points
Short Answers	10
Value Iteration	20
Q-Learning	15
Bayes Nets	20
Probability	12
Machine Learning	23

Problem 1. Short Answer[10 pts]

1.1. [3 pts] Briefly explain what “epsilon-greedy” means in Q-learning and discuss what aspects of off-policy learning it aims to improve.

1.2. [4 pts] In practice, we often use approximate Q-learning by learning the following function, $Q(s, a) = g(f_1(s, a), f_2(s, a) \dots , f_n(s, a))$. What is the advantage of using this approach compared to tabular (or exact) Q-learning?

1.3 [3 pts] Short Answer – Briefly describe a sign of overfitting in Naive Bayes learning, and how it can be avoided.

Problem 2. Value Iteration (MDPs) [20 pts]

Consider the 101x3 world below. In the start state, the agent has a choice of two deterministic actions, Up or Down, but in the other states the agent always takes the deterministic action, Right. Grayed out parts are not accessible to the agent.

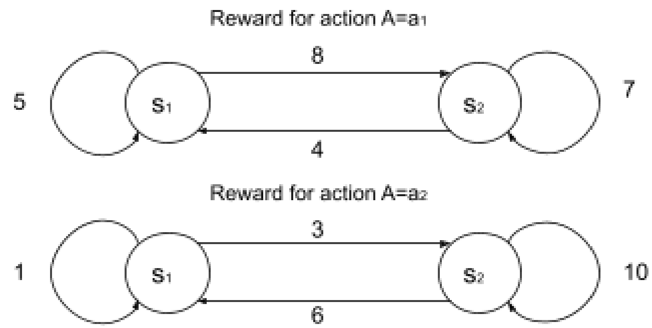
+50	-1	-1	-1	-1	…	-1	-1	-1	-1(**TERMINAL**)
START									
-50	+1	+1	+1	+1	…	+1	+1	+1	+1(**TERMINAL**)

2.1. [10 pts] Compute the utility of each action as a function of γ.

2.2. [10pts] Assuming a discounted reward function, for what values of the discount factor γ should the agent choose Up as the initial action?

Problem 3. Q-Learning (Reinforcement learning/MDPs) [15 pts]

Let's run Q-learning on the following problem with the state space $S = \{s_1, s_2\}$ and the action space $A = \{a_1, a_2\}$. The following state machines show rewards for each state transition and action:



The following table shows the initial Q-values each state and action:

Q-Values			
		Action	
		a_1	a_2
State	s_1	0	0
	s_2	1	1

Table 1. Initial Q-Values

Then, we observe the two following episodes:

<u>*Episode* 1:</u>

$(previous = s_1,\ action = a_1,\ next = s_2)$

$s_1,\ a_1,\ s_2,\ 8$

<u>*Episode* 2:</u>

$(previous = s_2,\ action = a_2,\ next = s_2)$

$s_2,\ a_2,\ s_2,\ 10$

We will conduct Q-learning with discount factor $\gamma = 0.5$ and learning rate $\alpha = 0.1$.

3.1. [7 pts] From the initial Q-value table in Table 1, update Q-values given the first episode. What is the updated Q-value table?

		Action	
		a_1	a_2
State	s_1		
	s_2		

3.2. [8 pts] **From the Q-value table updated in 3.1**, update Q-values with the second episode. What is the updated Q-value table?

		Action	
		a_1	a_2
State	s_1		
	s_2		

Problem 4. Bayes Nets [20 pts]

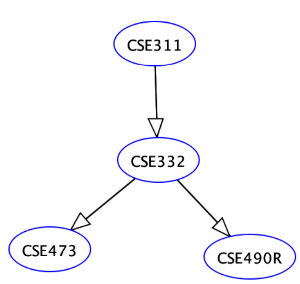
As part of a comprehensive study of the role of 10-601 on people's happiness we have been collecting important data from graduating students. In an entirely optional survey that all students are required to complete, where they were asked about whether they partied, whether they did well on their homeworks and projects, whether they used a Mac, and whether they thought of themselves as smart, creative, happy and successful. After consulting a behavioral psychologist, the following complete set of conditional relationships between the eight binary variables *Party, Smart, Creative, HW, Mac, Project, Success,* and *Happy*:

- Students' homework performance depends only on their partying habits and how smart they are.
- Students' Mac use is related to how smart and creative they are.
- Students' project outcomes depend only on how smart and creative they are.
- A student's success depends on their performance on homeworks and projects alone.
- Finally, students are happy depending on their degree of success, their partying habits, and if they use a Mac!

4.1 [6 pts] Draw a reasonable Bayes Network for the relationships between these variables.

4.2 [6 pts] Write joint distribution as a product of conditional probabilities for the above Bayesian network.

4.3 [8 pts] Consider the following Bayesian Network about class prerequisites in UW.



Give reasonable probability tables for all nodes in this subnetwork.

Problem 5. Probability [12 pts]

From patients admitted to the emergency room, let's assume 6% of patients were admitted due to the infectious disease A. According to the clinical test for the infectious disease A, 85% of infected patients are test-positive while 3% of non-infected patients are test-positive.

5.1. [4 pts] What is the **joint** probability of the clinical test being positive and a patient was infected?

5.2. [4 pts] What is the conditional probability that the patient was infected if the clinical test was positive?

5.3. [4 pts] What is the conditional probability that the clinical test was not positive but the patient was infected?

Problem 6. Machine Learning (Perceptrons/Naive Bayes) [23 pts]

6.1 [12pts] Naive Bayes

In this problem, you will build a Naive Bayes classifier for a Professor's email spam filter at UW - a machine learning algorithm to determine whether they should read a mail or not. To train this classifier model, the following data set of binary-valued features about each email is available, including whether they know the author or not, whether the email is long or short, and whether it has any of several key words, along with a final decision about whether to read it (y = +1 for "read", y = −1 for "discard").

x1 know author?	x2 Is long?	x3 has "research"?	x4 has "grade"?	x5 has "lottery"?	y read?
0	0	1	1	0	-1
1	1	0	1	0	-1
0	1	1	1	1	-1
1	1	1	1	0	-1
0	1	0	0	0	-1
1	0	1	1	1	1
0	0	1	0	0	1
1	0	0	0	0	1
1	0	1	1	0	1
1	1	1	1	1	-1

Note: In case of ties, the email is generally read (+1).

i. [8pt] Compute all the probabilities necessary for a naïve Bayes classifier, i.e., the class probability $p(y)$ and all the individual feature probabilities $p(x_i|y)$, for each class y and feature x_i.

ii. [4pt] Which class would be predicted for x = (0 0 0 0 0)? What about for x = (1 1 0 1 0)?

6.2 [11 pts] Consider using the perceptron algorithm to learn the logical OR function with the training set:

(x_1, x_2, b)	y^*
(-1,-1,1)	-1
(-1,1,1)	1
(1,-1,1)	1
(1,1,1)	1

Assume the following perceptron definition: $f(x) = sign(w \cdot x)$ where

$$\text{sign}(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

Updating if we guess incorrectly, i.e., $f(x) \neq y^*$, using the rule:

$$w \leftarrow w + y^* \cdot x$$

(i) [6 pts] Fill out the below table:

Iteration	x	w	$f(x)$	y^*
1	(-1,-1,1)	(0,0,0)	1	-1
2	(-1,1,1)			1
3	(1,-1,1)			1
4	(1,1,1)			1
5	(-1,-1,1)			-1
6	(-1,1,1)			1
7	(1,-1,1)			1

8	(1,1,1)			1

(ii) [3 pts] Has training converged? Why or why not?

(iii) [2 pts] If we initialized w to a non-zero weight vector would we necessarily converge to the same final weight vector? Why or why not?