CSE573 - Writing Assignment 2

Joseph David

May 27 2022

Problem 1

1.1 ”Epsilon-greedy” refers to a type of Q-learning in which the agent performs a random action some percent of the time, as opposed to performing the action that is currently considered optimal. The purpose of epsilon-greedy Q-learning is to balance the ”exploration vs exploitation” tradeoff in reinforcement learning, so that the agent explores its environment to some extent while still capitalizing on known rewards.

1.2 The benefit of approximate Q-learning is its generalizability to other environments. If one were to use exact Q-learning, then the information about one environment would be useless in any other environment. By defining Q-values as a function of features of states of an environment, we are able to apply what we learned to other environments in which these features are relevant. Approximate Q-learning is also useful because you only need to learn the weights of the features as opposed to all the state action pairs, which is usually fewer parameters.

1.3 One sign of overfitting is that the model prescribes a probability of zero or 100% to certain events. This may happen if that event is not occurred in the training data, but may still occur in general. One can fix this by performing Laplace smoothing where one assumes that every occurrence was seen once more than it actually was, so that we do not prescribe a zero probability to any occurrence.

Problem 2

2.1 We see that the utility of action "UP" is given by

$$\begin{aligned} 50 + (-1)\gamma + (-1)\gamma^2 + \cdots + (-1)\gamma^{100} &= 50 - (\gamma + \gamma^2 + \cdots + \gamma^{100}) \\ &= 50 - \left(\frac{1-\gamma^{101}}{1-\gamma} - 1 \right) \\ &= 51 - \frac{1-\gamma^{101}}{1-\gamma} \end{aligned}$$

and similarly the utility for "DOWN" is given by

$$-50 + \gamma + \gamma^2 + \cdots + \gamma^{100} = -51 + \frac{1-\gamma^{101}}{1-\gamma}.$$

2.2 From 2.1, we want to find for which values of γ

$$51 - \frac{1-\gamma^{101}}{1-\gamma} > -51 + \frac{1-\gamma^{101}}{1-\gamma}.$$

Plotting this, we see that "UP" is the optimal action when $\gamma \in (0, 0.9844)$ and "DOWN" otherwise.

Problem 3

3.1 Since the episode is $(s_1, a_1, s_2, 8)$, only $Q(s_1, a_1)$ gets updated.

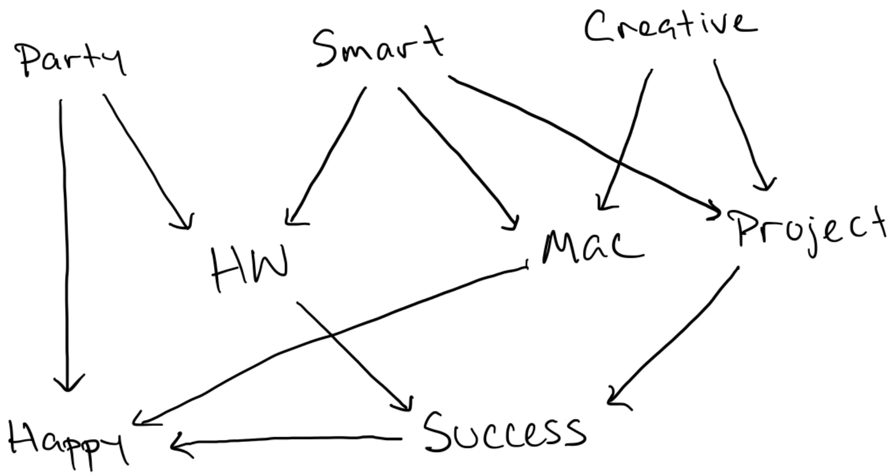
	a_1	a_2
s_1	0.85	0
s_2	1	1

3.2 Since the episode is $(s_2, a_2, s_2, 10)$, only $Q(s_2, a_2)$ gets updated.

	a_1	a_2
s_1	0.85	0
s_2	1	1.95

Problem 4

4.1



4.2

$$\begin{aligned} P(happy, smart, creative, hw, mac, project, party, success) &= P(happy|success, party, mac)P(success|project, hw) \\ &\cdot P(hw|smart, party)P(project|smart, creative) \\ &\cdot P(mac|smart, creative)P(party)P(creative)P(smart) \end{aligned}$$

4.3 I am not a CS major, but I looked up which courses are prerequisites to determine when a probability of taking a class should be zero. For CSE311, since this is probably an important class for the computer science major, I set

$$P(311) = 0.95.$$

Then, since 311 is a prereq for 332 and I assume 332 is a popular course, we have

311	P(332)
T	0.8
F	0

.

Since 332 is a prereq for both 473 and 490R, we set

332	P(473)
T	0.5
F	0

.

332	P(490R)
T	0.2
F	0

.

Problem 5

5.1 $P(positive, infected) = P(positive|infected)P(infected) = 0.85 \cdot 0.06 = 5.1\%$

5.2

$$\begin{aligned} P(infected|positive) &= \frac{P(positive|infected)P(infected)}{P(positive)} \\ &= \frac{0.051}{0.051 + (0.94)(0.03)} \\ &= 64.4\%. \end{aligned}$$

5.3

$$\begin{aligned} P(infected|negative) &= \frac{P(negative|infected) \cdot P(infected)}{P(negative)} \\ &= \frac{(0.15)(0.06)}{(0.15)(0.06) + (0.97)(.94)} \\ &= 0.99\%. \end{aligned}$$

Problem 6

6.1 (i) All of the probabilities shown are of the form $X_i = 1|Y = \pm 1$, since $P(X_i = 0|Y = \pm 1) = 1 - P(X_i = 1|Y = \pm 1)$.

	X1	X2	X3	X4	X5
Y = 1	3/4	0	3/4	1/2	1/4
Y= -1	1/2	5/6	2/3	5/6	1/3

(ii) Using the table from (i) we see that

$$P(X = (0,0,0,0,0), Y = 1) = (1/4)(1)(1/4)(1/2)(1/4) = \frac{1}{128}$$

and similarly we find

$$P(X = (0,0,0,0,0), Y = -1) = (1/2)(1/6)(1/3)(1/6)(2/3) = \frac{1}{324}.$$

Since $P(Y|X) = P(X,Y)/P(X)$, the likelihood values are proportional to $P(X,Y)$ and so our classifier would identify this as ”read” ($Y = 1$). Classifying $x = (1,1,0,1,0)$ we find

$$P(X = (1,1,0,1,0), Y = 1) = (3/4)(0)(1/4)(1/2)(3/4) = 0$$

$$P(X = (1,1,0,1,0), Y = -1) = (1/2)(5/6)(1/3)(1/6)(2/3) = \frac{5}{324},$$

so we classify this as ”discard” ($Y = -1$).

6.2 (i)

Iteration	x	w	f(x)	y*
1	(-1,-1,1)	(0,0,0)	1	-1
2	(-1,1,1)	(1,1,-1)	-1	1
3	(1,-1,1)	(0,2,0)	-1	1
4	(1,1,1)	(1,1,1)	1	1
5	(-1,-1,1)	(1,1,1)	-1	-1
6	(-1,1,1)	(1,1,1)	1	1
7	(1,-1,1)	(1,1,1)	1	1
8	(1,1,1)	(1,1,1)	1	1

(ii) Training has converged since we have cycled through the entire training data set and the perceptron has correctly labelled each data point. Therefore these is no chance that the weight vector will be updated again.

(iii) You might converge to a different weight vector, such as a multiple of $(1,1,1)$.