CS 188
Fall 2019

Exam Prep 5 Solutions

Q1. RL

Pacman is in an unknown MDP where there are three states [A, B, C] and two actions [Stop, Go]. We are given the following samples generated from taking actions in the unknown MDP. For the following problems, assume $\gamma = 1$ and $\alpha = 0.5$.

(a) We run Q-learning on the following samples:

s	a	s'	r
A	Go	B	2
C	Stop	A	0
B	Stop	A	-2
B	Go	C	-6
C	Go	A	2
A	Go	A	-2

What are the estimates for the following Q-values as obtained by Q-learning? All Q-values are initialized to 0.

(i) $Q(C, Stop) =$ 0.5

(ii) $Q(C, Go) =$ 1.5

For this, we only need to consider the following three samples.

$$Q(A, Go) \leftarrow (1-\alpha)Q(A, Go) + \alpha(r + \gamma \max_a Q(B, a)) = 0.5(0) + 0.5(2) = 1$$
$$Q(C, Stop) \leftarrow (1-\alpha)Q(C, Stop) + \alpha(r + \gamma \max_a Q(A, a)) = 0.5(0) + 0.5(1) = 0.5$$
$$Q(C, Go) \leftarrow (1-\alpha)Q(C, Go) + \alpha(r + \gamma \max_a Q(A, a)) = 0.5(0) + 0.5(3) = 1.5$$

(b) For this next part, we will switch to a feature based representation. We will use two features:

- $f_1(s, a) = 1$
- $f_2(s, a) = \begin{cases} 1 & a = \text{Go} \\ -1 & a = \text{Stop} \end{cases}$

Starting from initial weights of 0, compute the updated weights after observing the following samples:

s	a	s'	r
A	Go	B	4
B	Stop	A	0

What are the weights after the first update? (using the first sample)

(i) $w_1 =$ 2

(ii) $w_2 =$ 2

$$Q(A, Go) = w_1 f_1(A, Go) + w_2 f_2(A, Go) = 0$$
$$difference = [r + max_a Q(B, a)] - Q(A, Go) = 4$$
$$w_1 = w_1 + \alpha(difference) f_1 = 2$$
$$w_2 = w_2 + \alpha(difference) f_2 = 2$$

What are the weights after the second update? (using the second sample)

(iii) $w_1 =$ 4

(iv) $w_2 =$ 0

$$Q(B, Stop) = w_1 f_1(B, Stop) + w_2 f_2(B, Stop) = 2(1) + 2(-1) = 0$$
$$Q(A, Go) = w_1 f_1(A, Go) + w_2 f_2(A, Go) = 2(1) + 2(1) = 4$$
$$difference = [r + max_a Q(A, a)] - Q(B, Stop) = [0 + 4] - 0 = 4$$
$$w_1 = w_1 + \alpha(difference) f_1 = 4$$
$$w_2 = w_2 + \alpha(difference) f_2 = 0$$

Q2. Reinforcement Learning

(a) Each True/False question is worth 1 points. Leaving a question blank is worth 0 points. **Answering incorrectly is worth −1 points.**

(i) [*true* or *false*] Temporal difference learning is an online learning method.
Temporal difference learning is used when we don't have the full MDP model and must collect online samples.

(ii) [*true* or *false*] Q-learning: Using an optimal exploration function leads to no regret while learning the optimal policy.
In order to learn the optimal policy, you must explore, and exploring in general has a non-zero chance of regret.

(iii) [*true* or *false*] In a deterministic MDP (i.e. one in which each state / action leads to a single deterministic next state), the Q-learning update with a learning rate of $\alpha = 1$ will correctly learn the optimal q-values (assume that all state/action pairs are visited sufficiently often). Remember that the learning rate is only there because we are trying to approximate a summation with a single sample. In a deterministic MDP where s' is the single state that always follows when we take action a in state s, we have $Q(s,a) = R(s,a,s') + \max_{a'} Q(s',a')$, which is exactly the update we make.

(iv) [*true* or *false*] A small discount (close to 0) encourages greedy behavior.
A discount close to zero will place extremely small values on rewards more than one step away, leading to greedy behavior that looks for immediate rewards.

(v) [*true* or *false*] A large, negative living reward ($\ll 0$) encourages greedy behavior.
A negative living reward adds a penalty for every step taken. If that penalty is large, the agent will prefer to find an exit as soon as possible despite potential rewards on longer paths.

(vi) [*true* or *false*] A negative living reward can always be expressed using a discount < 1.
While both negative living rewards and discounts can encourage similar behavior, they are mathematically different. A discount has a multiplicative effect at each step, whereas a living reward only has an additive effect.

(vii) [*true* or *false*] A discount < 1 can always be expressed as a negative living reward.
While both negative living rewards and discounts can encourage similar behavior, they are mathematically different. A discount has a multiplicative effect at each step, whereas a living reward only has an additive effect.

(b) Given the following table of Q-values for the state A and the set of actions $\{Forward, Reverse, Stop\}$, what is the probability that we will take each action on our next move when we following an ϵ-greedy exploration policy (assuming any random movements are chosen uniformly from all actions)?

$Q(A, Forward) = 0.75$
$Q(A, Reverse) = 0.25$
$Q(A, Stop) = 0.5$

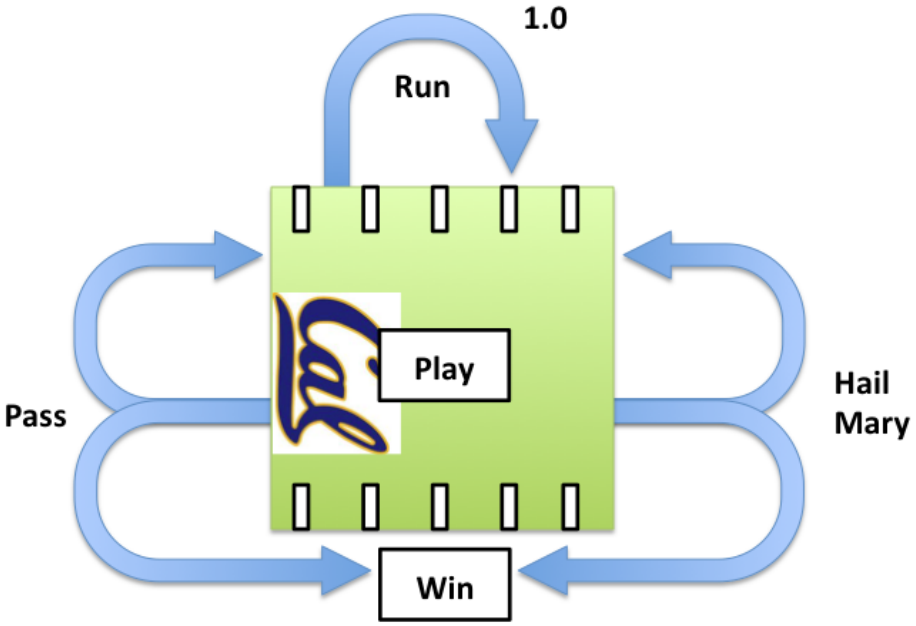
Action	Probability (in terms of ϵ)
Forward	$(1-\epsilon) + \frac{\epsilon}{3} = 1 - \frac{2\epsilon}{3}$
Reverse	$\frac{\epsilon}{3}$
Stop	$\frac{\epsilon}{3}$

Q3. MDPs and RL: Go Bears!

Cal's Football team is playing against UCLA for the big homecoming game Saturday night. With a lot of losses in the season so far, Cal needs to switch up their strategy to get any hope of winning this game.

Luckily, the Quarterback (Joe) is a star student in CS188 and has decided to model the game as a Markov Decision Process. There are only two states – the *Play* state (shown as the field in the diagram) and the *Win* State. Although the connectivity of the states is known, the probabilities for each are not.

There are no actions available from the *Win* state – the game simply ends.



From the *Play* state there are three actions: *Run*, *Pass*, and *HailMary*. The connectivity of each action to the two states is shown above.

Reward Values:

State	Action	State'	R(s,a,s')
Play	Run	Play	2
Play	Pass	Play	4
Play	Pass	Win	10
Play	Hail Mary	Play	0
Play	Hail Mary	Win	100

(a) Learning Values Joe wants to learn the value of the play state so he can estimate the outcome of the game. He uses a discount factor of 0.5 for all questions below.

(i) Joe first uses temporal difference value learning to learn the value of the *play* state. After initializing his beliefs to 0, he sees two episodes while in tape review. With a learning rate α of 0.5 what value of the state *play* does he learn?

State	Action	State'
Play	Run	Play
Play	Hail Mary	Play

$$V(play) = 2 + 0.5 * V(play) = 2 + 0.5 * 0 = 2$$
$$V(play) \leftarrow (1 - \alpha) * 0 + \alpha * 2 = 1$$

then

$$V(play) = 0 + 0.5 * 1 = 0.5$$
$$V(play) \leftarrow (1 - \alpha) * 1 + \alpha * 0.5 = 0.75$$

$$V(play) = 0.75$$

(ii) Coach Tedford decides to give Joe a fixed policy instead:

$$\pi(s) \quad = \quad Run$$

What value for the state *play* would Joe calculate if he ran value iteration until convergence? Keep in mind that $\sum_{n=0}^{\infty}(\frac{1}{2})^n = 2 - (\frac{1}{2})^n = 1 + 0.5 + 0.25 + 0.125 + ...$

$$V(play) = 2 + 0.5 * (2 + 0.5 * (2 + 0.5 * (2 + 0.5 * ....)))$$
$$V(play) = 2 + 1 + 0.5 + 0.25 + 0.125... = 4$$

$$V^{\pi}(play) \quad = 4$$

(b) Game Time Joe watches the next lecture video from class and now wants to use Q-learning to compute his optimal strategy.

(i) First Joe uses temporal difference Q-learning to learn the values of the Q nodes. He sees three episodes during the first quarter:

State	Action	State'
Play	Run	Play
Play	Hail Mary	Play
Play	Pass	Win

Update the Q node values after processing each episode (in order). Use a learning rate of 0.5 and a discount rate of 0.5.

$$Q(play, run) = 2 + 0.5 * 0 = 2$$
$$Q(play, hail) = 0 + 0.5 * 1 = 0.5$$
$$Q(play, pass) = 10 + 0.5 * 0 = 10$$

Remember you need to update V(play) as this process continues. and learning rate of alpha of 0.5 means all the above values get averaged against 0 when stored.

(c) Q learning is going well, but it's taking too much time. Thankfully Oski shows up with some special information – he has watched so many games that he know's the true transition probabilities! Here they are:

State	Action	$Q(s,a)$
Play	Run	1
Play	Hail Mary	0.25
Play	Pass	5

State	Action	State'	R(s,a,s')	T(s,a,s')
Play	Run	Play	2	1.0
Play	Pass	Play	4	0.5
Play	Pass	Win	10	0.5
Play	Hail Mary	Play	0	0.9
Play	Hail Mary	Win	100	0.1

(i) Now with these probabilities, what is the optimal policy when there is one time step left? The value?

$$Q(play, hail) = 0.1 * (100) + 0.9 * (0) = 10$$

$$\pi_{k=1}(play) = hail \qquad mary$$
$$V_{k=1}(play) = 10$$

(ii) For two time steps left, what is the optimal policy with discount factor 0.5? Hint: you can use your value above to aid in this computation.

$$Q(play, hail) = 0.1 * (100 + 0.5 * 0) + 0.9 * (0 + 0.5 * 10) = 14.5$$

$$\pi_{k=2}(play) = hail \qquad mary$$
$$V_{k=2}(play) = 14.5$$