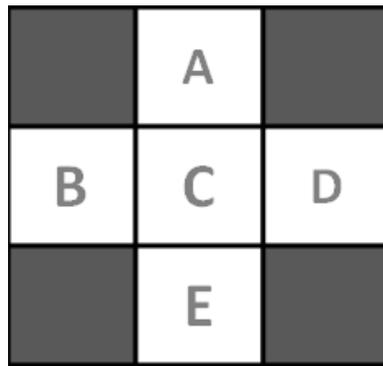


1 Learning in Gridworld

Consider the example gridworld that we looked at in lecture. We would like to use TD learning and q-learning to find the values of these states.



Suppose that we have the following observed transitions:
(B, East, C, 2), (C, South, E, 4), (C, East, A, 6), (B, East, C, 2)

The initial value of each state is 0. Assume that $\gamma = 1$ and $\alpha = 0.5$.

(a) What are the learned values from TD learning after all four observations?

$$V(B) = 3.5$$

$$V(C) = 4$$

All other states have a value of 0.

(b) What are the learned Q-values from Q-learning after all four observations?

$$Q(B, \text{East}) = 3$$

$$Q(C, \text{South}) = 2$$

$$Q(C, \text{East}) = 3$$

All other q-states have a value of 0.

2 Odds and Ends

1. Can all MDPs be solved using expectimax search? Justify your answer.

No, MDPs with self loops lead to infinite expectimax trees. Unlike search problems, this issue cannot be addressed with a graph-search variant.

2. Why might Q-learning be superior to TD learning of values?

- (a) If you use temporal difference learning on the values, it is hard to extract a policy from the learned values. Specifically, you would need to know the transition model T and reward function R . For Q-learning, the policy can be extracted directly by taking $\pi(s) = \arg \max_a Q(s, a)$.
- (b) While TD learning learns values for a particular policy (online policy evaluation), Q-learning is *off-policy*: it will converge to Q-values that give you the optimal policy, even if the actions you used to explore were sub-optimal. The caveats to this are that you need to explore enough, and you need conditions on the learning rate α (has to eventually be small enough, but not decrease too fast)

3. When performing Q-learning with ϵ -greedy action selection, is it a good idea to decrease ϵ to 0 with time? Why or why not? Remember that ϵ is the (small) probability that you choose a *random* action, and $1 - \epsilon$ is the (large) probability you act on your current policy.

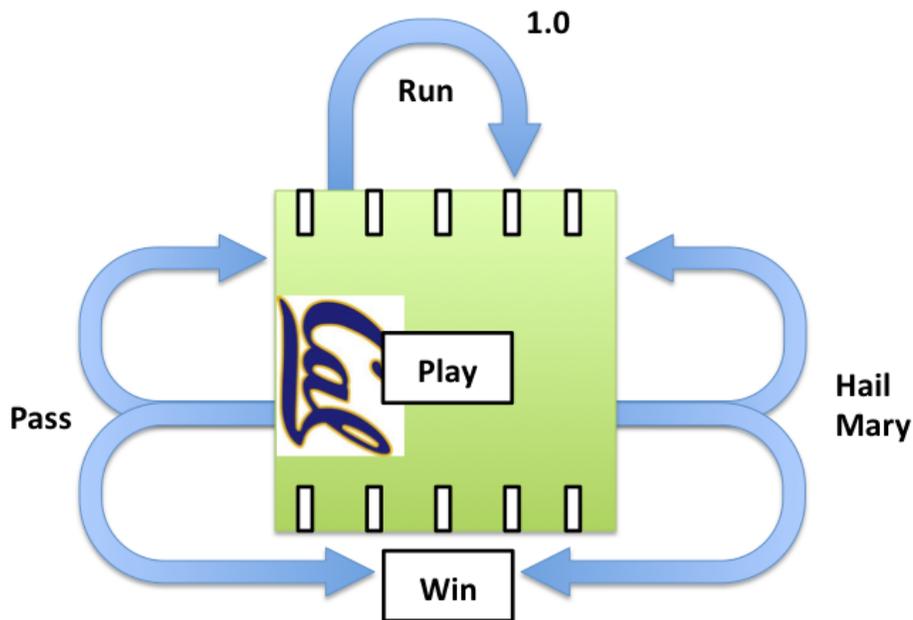
Yes, especially when using on-policy learning methods. The reason is that as the agent learns the actual optimal policy for the world, it should switch from a mix of exploration and exploitation to mostly exploitation (unless the world is changing, in which case it should always keep exploring).

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

$$V(\text{play}) = 2 + 0.5 * V(\text{play}) = 2 + 0.5 * 0 = 2$$

$$V(\text{play}) \leftarrow (1 - \alpha) * 0 + \alpha * 2 = 1$$

then

$$V(\text{play}) = 0 + 0.5 * 1 = 0.5$$

$$V(\text{play}) \leftarrow (1 - \alpha) * 1 + \alpha * 0.5 = 0.75$$

$$V(\text{play}) = 0.75$$

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

$$\pi(s) = \text{Run}$$

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

$$V(\text{play}) = 2 + 0.5 * (2 + 0.5 * (2 + 0.5 * (2 + 0.5 * \dots)))$$

$$V(\text{play}) = 2 + 1 + 0.5 + 0.25 + 0.125 \dots = 4$$

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

(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.

$$Q(\text{play}, \text{run}) = 2 + 0.5 * 0 = 2$$

$$Q(\text{play}, \text{hail}) = 0 + 0.5 * 1 = 0.5$$

$$Q(\text{play}, \text{pass}) = 10 + 0.5 * 0 = 10$$

Remember you need to update $V(\text{play})$ 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?

$$Q(\text{play}, \text{hail}) = 0.1 * (100) + 0.9 * (0) = 10$$

$$\pi_{k=1}(\text{play}) = \text{hail} \quad \text{mary}$$

$$V_{k=1}(\text{play}) = 10$$

(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.

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

$$\pi_{k=2}(\text{play}) = \text{hail} \quad \text{mary}$$

$$V_{k=2}(\text{play}) = 14.5$$