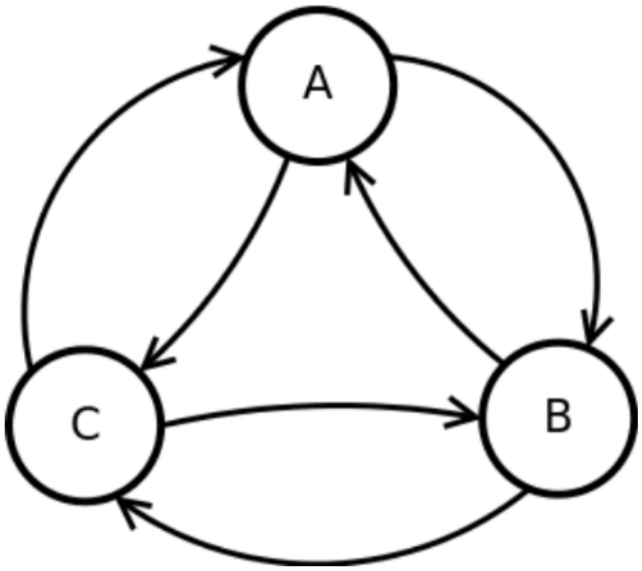
Q1. MDPs - Value Iteration (30 Points)

Part 1 - Cycle. (15 points) Consider the following transition diagram, transition function, and reward function for an MDP.



Discount Factor, $\gamma = 0.5$

s	a	s'	T(s,a,s')	R(s,a,s')
A	Clockwise	B	1.0	0.0
A	Counterclockwise	C	1.0	-2.0
B	Clockwise	A	0.4	-1.0
B	Clockwise	C	0.6	2.0
B	Counterclockwise	A	0.6	2.0
B	Counterclockwise	C	0.4	-1.0
C	Clockwise	A	0.6	2.0
C	Clockwise	B	0.4	2.0
C	Counterclockwise	A	0.4	2.0
C	Counterclockwise	B	0.6	0.0

P1.1. Suppose that after iteration k of value iteration, we obtain the following values for V_k:

$V_k(A)$	$V_k(B)$	$V_k(C)$
0.400	1.400	2.160

Provide the values of $V_{k+1}(A), V_{k+1}(B)$, and $V_{k+1}(C)$.

Solution. Using the Bellman equation $V_{k+1}(s) = \max_a \Sigma_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$:

$$\begin{aligned} V_{k+1}(A) &= \max[1.0 \cdot (0.0 + 0.5 \cdot 1.400), 1.0 \cdot (-2.0 + 0.5 \cdot 2.160)] \\ &= \max(0.7, -0.92) \\ &= \boxed{0.7} \end{aligned}$$

$$\begin{aligned} V_{k+1}(B) &= \max[0.4 \cdot (-1.0 + 0.5 \cdot 0.400) + 0.6 \cdot (2.0 + 0.5 \cdot 2.160), \\ &\quad 0.6 \cdot (2.0 + 0.5 \cdot 0.400) + 0.4 \cdot (-1.0 + 0.5 \cdot 2.160)] \\ &= \max[1.528, 1.352] \\ &= \boxed{1.528} \end{aligned}$$

$$\begin{aligned} V_{k+1}(C) &= \max[0.6 \cdot (2.0 + 0.5 \cdot 0.400) + 0.4 \cdot (2.0 + 0.5 \cdot 1.400), \\ &\quad 0.4 \cdot (2.0 + 0.5 \cdot 0.400) + 0.6 \cdot (0.0 + 0.5 \cdot 1.400)] \\ &= \max[2.4, 1.3] \\ &= \boxed{2.4} \end{aligned}$$

P1.2. Suppose that we ran value iteration to completion and found the following value function, V^*. What are the optimal actions from states A, B, and C, respectively?

$V^*(A)$	$V^*(B)$	$V^*(C)$
0.881	1.761	2.616

Solution. Using $\pi^*(s) = \max_a \Sigma_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$:

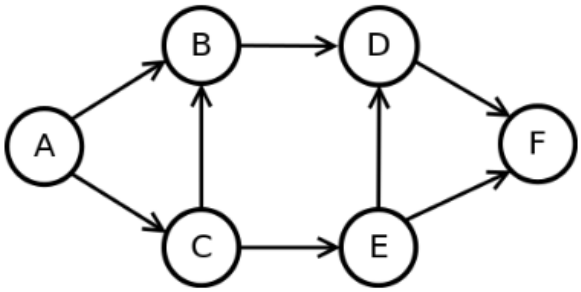
$$\begin{aligned} \pi^*(A) &= \max[1.0 \cdot (0.0 + 0.5 \cdot 1.761), 1.0 \cdot (-2.0 + 0.5 \cdot 2.616)] \\ &= \max(0.8805, -0.692) \\ &= \boxed{0.8805} \end{aligned}$$

$$\begin{aligned} \pi^*(B) &= \max[0.4 \cdot (-1.0 + 0.5 \cdot 0.881) + 0.6 \cdot (2.0 + 0.5 \cdot 2.616), \\ &\quad 0.6 \cdot (2.0 + 0.5 \cdot 0.881) + 0.4 \cdot (-1.0 + 0.5 \cdot 2.616)] \\ &= \max[1.761, 1.5875] \\ &= \boxed{1.761} \end{aligned}$$

$$\begin{aligned}\pi^*(C) &= \max[0.6 \cdot (2.0 + 0.5 \cdot 0.881) + 0.4 \cdot (2.0 + 0.5 \cdot 1.761),\\ &\quad 0.4 \cdot (2.0 + 0.5 \cdot 0.881) + 0.6 \cdot (0.0 + 0.5 \cdot 1.761)]\\ &= \max[2.6165, 1.5045]\\ &= \boxed{2.6165}\end{aligned}$$

The values 0.8805, 1.761, and 2.6165 correspond to the clockwise action for each state. Therefore the optimal actions for A, B, and C are all $\boxed{clockwise}$.

Part 2 - Convergence. (15 Points) We will consider a simple MDP that has six states, A, B, C, D, E, and F. Each state has a single action, go. An arrow from a state x to a state y indicates that it is possible to transition from state x to next state y when go is taken. If there are multiple arrows leaving a state x, transitioning to each of the next states is equally likely. The state F has no outgoing arrows: once you arrive in F, you stay in F for all future times. The reward is one for all transitions, with one exception: staying in F gets a reward of zero. Assume a discount factor $\gamma = 0.5$. We assume that we initialize the value of each state to 0. (Note: you should not need to explicitly run value iteration to solve this problem.)



P2.1. After how many iterations of value iteration will the value for state E have become exactly equal to the true optimum? (Enter inf if the values will never become equal to the true optimal but only converge to the true optimal.)

Solution. When we run the value iteration on E we have two possible paths with equal probability: from E $\rightarrow$ D and E $\rightarrow$ F. F will always stay the same with a value of 0 since staying in F has an immediate reward of 0 (we say that F is an absorbing state). D $\rightarrow$ F will also stay the same since F is unchanging. D's value will always be 1. It will then take one iteration for E to converge to 1.25 (since D and F are not changing).

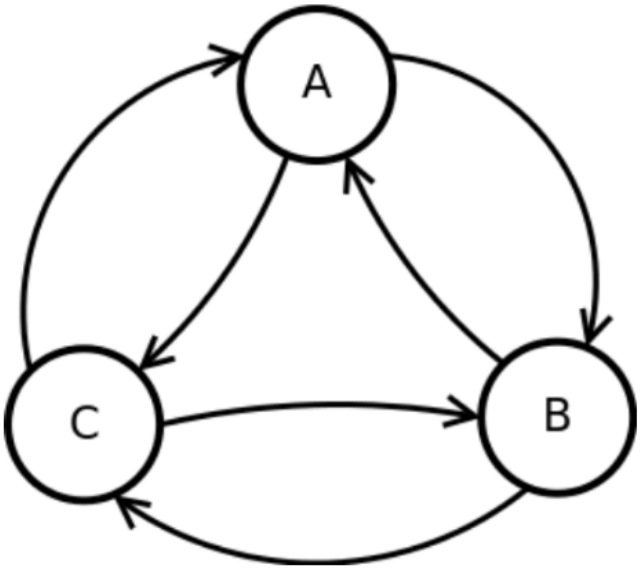
P2.2. How many iterations of value iteration will it take for the values of all states to converge to the true optimal values? (Enter inf if the values will never become equal to the true optimal but only converge to the true optimal.)

Solution. Following the logic of the previous question, E converges after one iteration, therefore B does as well (being dependent on D only). By that reasoning, since E or B aren't changing after 1 iteration, neither will C. Finally, since C and B aren't changing, A will also converge after the first iteration. Thus the entire set of states converges to true optimal after just one iteration.

A note: I'm not sure how to decide on what converges at which step. I think if you're considering starting with a terminating node (given no loops) that you should see when that terminates then it propagates outward. One could look at two versions of this. First, as I described above, D converges on first iteration then the first iteration of E and B start and they all converge after one step. The other route I considered was that D converges on the first iteration, but E and B had to 'wait' for it. So E and B converge on step 2, then C on step 3 and A on step 4. I wanted to speak on this in case it's the correct solution to the second part.

Q2. MDPs - Policy Iteration (20 Points)

Consider the following transition diagram, transition function and reward function for an MDP.



Discount Factor, $\gamma = 0.5$

s	a	s'	T(s,a,s')	R(s,a,s')
A	Clockwise	B	1.0	0.0
A	Counterclockwise	C	1.0	-2.0
B	Clockwise	A	0.4	-1.0
B	Clockwise	C	0.6	2.0
B	Counterclockwise	A	0.6	2.0
B	Counterclockwise	C	0.4	-1.0
C	Clockwise	A	0.6	2.0
C	Clockwise	B	0.4	2.0
C	Counterclockwise	A	0.4	2.0
C	Counterclockwise	B	0.6	0.0

Q1.1. (10 points) Suppose we are doing policy evaluation, by following the policy given by the left-hand side table below. Our current estimates (at the end of some iteration of policy evaluation) of the value of states when following the current policy is given in the right-hand side table.

Provide the value of $V^{\pi}_{k+1}(A), V^{\pi}_{k+1}(B)$, and $V^{\pi}_{k+1}(C)$.

A	B	C
Counterclockwise	Counterclockwise	Counterclockwise

$V^{\pi}_{k}(A)$	$V^{\pi}_{k}(B)$	$V^{\pi}_{k}(C)$
0.000	-0.840	-1.080

Solution. Now with the modified Bellman equation for policies,

$$V^{\pi} = \Sigma_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^{\pi}(s')],$$

we get the following values:

$$\begin{aligned} V^{\pi}_{k+1}(A) &= 1.0[-2.0 + 0.5(-1.080)] \\ &= \boxed{-2.54} \end{aligned}$$

$$V^{\pi}_{k+1}(B) = 0.6[2.0 + 0.5(0.000)] + 0.4[-1.0 + 0.5(-1.080)]$$
$$= \boxed{0.584}$$

$$V^{\pi}_{k+1}(C) = 0.4[2.0 + 0.5(0.000)] + 0.6[0.0 + 0.5(-0.840)]$$
$$= \boxed{0.548}$$

Q1.2. (10 points) Suppose that policy evaluation converges to the following value function, V^{π}_{∞}. Provide the values of $Q^{\pi}_{\infty}(A, clockwise)$ and $Q^{\pi}_{\infty}(A, counterclockwise)$. What is the updated action for A?

$V^{\pi}_{\infty}(A)$	$V^{\pi}_{\infty}(B)$	$V^{\pi}_{\infty}(C)$
-0.203	-1.114	-1.266

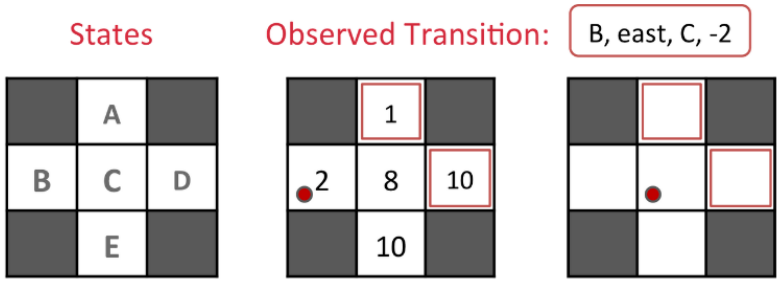
Solution. Using the same equation but with the converged values for $V^{\pi}_{\infty}(s)$:

$$Q^{\pi}_{\infty}(A, clockwise) = 1.0[0.0 + 0.5(-1.114)] = \boxed{-0.557}$$
$$Q^{\pi}_{\infty}(A, counterclockwise) = 1.0[-2.0 + 0.5(-1.266)] = \boxed{-2.633}$$

Therefore the updated action for A will be the higher of these values which corresponds to the clockwise action.

Q3. Temporal Difference Learning (10 Points)

Consider the gridworld shown below. The left panel shows the name of each state A through E. The middle panel shows the current estimate of the value function V^π for each state. A transition is observed, that takes the agent from state B through taking action east into state C, and the agent receives a reward of -2. Assuming $\gamma = 1, \alpha = 0.5$, what are the value estimates for $\hat{V}^\pi(A), \hat{V}^\pi(B), \hat{V}^\pi(C), \hat{V}^\pi(D)$, and $\hat{V}^\pi(E)$ after the TD learning update? (note: the value will change for one of the states only)



Solution. Using the formula above to compute the value estimates after the TD learning update, we know ahead of time that states A, C, D, and E won't change since we're not moving from that state into another. Given the information, the only state that will change is state B since we're moving from $B \to C$. Therefore the value estimates are as follows:

$$\hat{V}^\pi(A) = 1$$
$$\hat{V}^\pi(B) = 2\left(1 - \frac{1}{2}\right) + \frac{1}{2}[-2 + 1(8)] = \boxed{4}$$
$$\hat{V}^\pi(C) = 8$$
$$\hat{V}^\pi(D) = 10$$
$$\hat{V}^\pi(E) = 10$$

Q4. Active Reinforcement Learning (40 Points)

Q4.1. (20 points) Pacman is in an unknown MDP where there are four states [A, B, C, D] and two actions [Left, Right]. We are given the following samples generated from taking actions in the unknown MDP. For the following problems, assume $\gamma = 0.8$ and $\alpha = 0.75$. We run Q-learning on the following samples:

s	a	s'	r
A	Left	B	2.0
B	Right	D	-1.0
D	Left	C	3.0
C	Left	A	-2.0
A	Right	D	1.0

What are the estimates for the Q-values $Q(C, Left)$ and $Q(A, Right)$ as obtained by Q-learning? All Q-values are initialized to 0.

Solution. Since all Q-values are initialized to 0, this means that in the beginning, all states are only really concerned with the immediate reward to calculate the new Q-value. We then use the following to calculate the new Q-values:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r + \gamma \text{max}_{a'} Q(s',a')]$$

Running down the list and updating Q-values we get

$$\begin{aligned} Q(A,left) &= (1-0.75)\cdot 0 + 0.75[2.0 + 0.8\text{max}(Q(B,right))] \\ &= 0.75[2.0 + 0.8\text{max}(0)] \\ &= 1.5 \end{aligned}$$

$$\begin{aligned} Q(B,right) &= (1-0.75)\cdot 0 + 0.75[-1.0 + 0.8\text{max}(Q(D,left))] \\ &= 0.75[-1.0 + 0.8\text{max}(0)] \\ &= -0.75 \end{aligned}$$

$$\begin{aligned} Q(D,left) &= (1-0.75)\cdot 0 + 0.75[3.0 + 0.8\text{max}(Q(C,left))] \\ &= 0.75[3.0 + 0.8\text{max}(0)] \\ &= 2.25 \end{aligned}$$

Now noting that Q(A, left) = 1.5 and Q(A, right) = 0:

$$\begin{aligned} Q(C,left) &= (1-0.75)\cdot 0 + 0.75[-2.0 + 0.8\text{max}(Q(A,left), Q(A,right))] \\ &= 0.75[-2.0 + 0.8\text{max}(1.5, 0)] \\ &= \boxed{-0.6} \end{aligned}$$

and

$$\begin{aligned} Q(A, right) &= (1 - 0.75) \cdot 0 + 0.75[1.0 + 0.8\text{max}(Q(D, left))] \\ &= 0.75[-2.0 + 0.8\text{max}(2.25)] \\ &= \boxed{2.1} \end{aligned}$$

Q4.2. Approximate Q-Learning (14 points) For this part we will switch to a feature based representation. We will use the two features

- $f_1(s, a) = 1.$
- $f_2(s, a) = \begin{cases} -1 & \text{if } a = \text{Left} \\ 1 & \text{if } a = \text{Right} \end{cases}$

Starting from initial weights of 0, we are going to use the first two samples in the above table to update the weights:

1. What are the weights after the first update? (using the first sample)
2. What are the weights after the second update? (using the second sample)

Solution. I'm going to assume that we're using the Q-values as calculated in the previous step. This gives a table that looks like this:

s	a	s'	r	Q(s,a)
A	left	B	2.0	1.5
B	right	D	-1.0	-0.75
D	left	C	3.0	2.25
C	left	A	-2.0	-0.6
A	right	D	1.0	2.1

Therefore calculating the first round of weights using

$$w_i = w_i + \alpha[r + \gamma \text{max}(Q(s', a')) - Q(s, a)]f_i(s, a)$$

we get

$$\begin{aligned} w_1 &= w_1 + 0.75[2.0 + 0.8\text{max}(Q(B, right)) - Q(A, left)] \cdot 1 \\ &= 0 + 0.75[2.0 + 0.8(-0.75) - 1.5)] \\ &= \boxed{-0.075} \end{aligned}$$

$$\begin{aligned} w_2 &= w_2 + 0.75[2.0 + 0.8\text{max}(Q(B, right)) - Q(A, left)] \cdot (-1) \\ &= 0 - 0.75[2.0 + 0.8(-0.75) - 1.5)] \\ &= \boxed{0.075} \end{aligned}$$

On the second update, using the second sample we have

$$\begin{aligned} w_1 &= w_1 + 0.75[-1.0 + 0.8\text{max}(Q(D, left)) - Q(B, right)] \cdot 1 \\ &= -0.075 + 0.75[-1.0 + 0.8(2.25) - (-0.75)] \\ &= \boxed{1.0875} \end{aligned}$$

$$\begin{aligned} w_2 &= w_2 + 0.75[-1.0 + 0.8\text{max}(Q(D, left)) - Q(B, right)] \cdot 1 \\ &= 0.075 + 0.75[-1.0 + 0.8(2.25) - (-0.75)] \\ &= \boxed{1.2375} \end{aligned}$$

Q4.3. Exploration. (6 points) In Q-learning, we can modify the original reward function $R(s, a, s')$ to visit more states and choose new actions. $N(s, a)$ refers to the number of times that you have visited state s and taken action a in your samples.

1. Which of the following rewards would encourage the agent to visit unseeen states and actions **(Yes/No)**?:

- $R(s, a, s') + \frac{1}{1+N(s,a)}$
- $R(s, a, s') + N^2(s, a)$
- $-\exp(N(s, a) + 1)$

Solution.

- $R(s, a, s') + \frac{1}{1+N(s,a)}$ - Yes. As $N(s, a)$ grows, the reward for visiting that state drops. The initial reward would be $R(s, a, s') + 1$ but visiting again would make it $R(s, a, s') + 1/2$ therefore the reward is shrinking and the reward for visiting an unseen or less visited state would be greater.
- $R(s, a, s') + N^2(s, a)$ - No. Visiting the same state will always increase the reward. An agent would most like bounce between two states as each visit has an increased reward. Unless of course they get a reward for staying in the same state.
- $-\exp(N(s, a) + 1)$ - Yes. This is a decaying exponential based on number of times the agent has visited a state. If the reward is decaying exponentially as one visits a state repeatedly, this would strongly encourage the agent to seek states that it has not seen yet.

2. Which of the following modified rewards will eventually converge to the optimal policy with respect to the original reward function $R(s, a, s')$ **(Yes/No)**?:

- $R(s, a, s') + \frac{1}{1+N(s,a)}$
- $R(s, a, s') + N^2(s, a)$
- $-\exp(N(s, a) + 1)$

Solution.

- $R(s, a, s') + \frac{1}{1+N(s,a)}$ - Yes. Will converge to $R(s, a, s')$
- $R(s, a, s') + N^2(s, a)$ - No. As the state is visited multiple times, the reward function will continue to grow infinitely.
- $-\exp(N(s, a) + 1)$ - No. As a state is visited multiple times, this will tend to $-\infty$.

Q5. Properties of MDPs (Grads Only) (10 Points)

Consider an MDP ($\mathbf{S}$, $\mathbf{A}$, T, R) with a finite state space $\mathbf{S}$, finite action space $\mathbf{A}$, the transition function $T(s, a, s')$, a reward function $R(s, a, s')$, and a discount factor $\gamma \in (0, 1)$. The reward $R(s, a, s') \geq 1$ for all (s, a, s'). Denote by $V_k(s)$ the value of state s after k iterations regarding the value iteration method and $V^*(s)$ the optimal value of state s.

Initially, $V_0(s) = 1$ for all s. Prove that $V^*(s) \geq V_k(s)$, for all k.

Proof. Assume that our MDP has finite state space and action space $\mathbf{S}$ and $\mathbf{A}$ respectively. Also assume a transition function $T(s, a, s')$ and a reward function $R(s, a, s')$ such that $R(s, a, s') \geq 1 \ \forall (s, a, s')$. Denote the value function $V_k(s)$ for a state s after k iterations and note that $\lim_{k\to\infty} V_k(s) = V^*(s)$. Lastly, note that $V_0(s) = 1 \ \forall s$.

To prove that $V^*(s) \geq V_k(s) \ \forall s$, we proceed by induction.

-Base case ($k = 0$): Suppose that $k = 0$. Using the Bellman equation:

$$\begin{aligned} V_{k+1}(s) &= \max_a \Sigma_{s'} T(s, a, s')[R(s, a, s') + \gamma V_k(s')] \\ \Rightarrow V_1(s) &= \max_a \Sigma_{s'} T(s, a, s')[R(s, a, s') + \gamma V_0(s')] \end{aligned}$$

Since $V_0(s) = 1 \ \forall s$ and $R(s, a, s') \geq 1 \ \forall (s, a, s')$ we have that

$$\begin{aligned} V_1(s) &= \max_a \Sigma_{s'} T(s, a, s')[R(s, a, s') + \gamma V_0(s')] \\ &\geq \max_a \Sigma_{s'} T(s, a, s')[1 + \gamma] \end{aligned}$$

Now $\Sigma_{s'} T(s, a, s') = 1$ and since $\gamma \in (0, 1)$ we can conclude that $[1 + \gamma] > 1$. Therefore

$$\begin{aligned} \max_a \Sigma_{s'} T(s, a, s')[1 + \gamma] &\geq \max_a \Sigma_{s'} T(s, a, s') = 1 \\ &\Rightarrow V_1(s) \geq 1 = V_0(s) \\ &\Rightarrow V_1(s) \geq V_0(s). \end{aligned}$$

- Induction case: Assume that $V_k(s) \geq V_{k-1}(s) \ \forall s$. We aim to show that this holds for $V_{k+1}(s) \geq V_k(s)$. Again, using the Bellman equation and replacing V_k with V_{k-1} and converting to an inequality, we have that, based on assumption:

$$\begin{aligned} V_{k+1}(s) &= \max_a \Sigma_{s'} T(s, a, s')[R(s, a, s') + \gamma V_k(s')] \\ &\geq \max_a \Sigma_{s'} T(s, a, s')[R(s, a, s') + \gamma V_{k-1}(s')] \end{aligned}$$

Since $V_k(s) = \max_a \Sigma_{s'} T(s, a, s')[R(s, a, s') + \gamma V_{k-1}(s')]$ we can infer that $V_{k+1} \geq V_k$ by substitution, also on the predicate that $R(s, a, s') \geq 1$ and $\gamma \in (0, 1)$. Therefore we can conclude that since $\lim_{k\to\infty} V_k(s) = V^*(s)$, $V^*(s) \geq V_k(s) \ \forall s$. ■