Assignment → 2

Q→1 (a).

As given $\|\hat{V}_{k+1} - \hat{V}_k\|_\infty < \epsilon$. for $\epsilon > 0$.

& $\|\cdot\|$ denotes max norm.

$$\|V_k - V_*\|_\infty \leq \|V_k - V_{k+1}\|_\infty + \|V_{k+1} - V_*\|_\infty \quad \{\text{from Triangle inequality}\}$$

$$\leq \|V_k - V_{k+1}\|_\infty + \|BV_k - BV_*\|_\infty$$

$$\leq \|V_k - V_{k+1}\|_\infty + \gamma \|V_k - V_*\|_\infty$$

where γ = discount factor

$$\Rightarrow \|V_k - V_*\|_\infty \leq \epsilon + \gamma \|V_k - V_*\|_\infty.$$

$$\Rightarrow (1-\gamma)\|V_k - V_*\|_\infty \leq \epsilon$$

$$\Rightarrow \|V_k - V_*\|_\infty \leq \frac{\epsilon}{1-\gamma}$$

Now as asked to find = $\|V_{k+1} - V_*\|_\infty$

$$\|V_{k+1} - V_*\|_\infty \leq \gamma \|V_k - V_*\|$$

$$\boxed{\|V_{k+1} - V_*\|_\infty \leq \frac{\gamma\epsilon}{1-\gamma}}$$

distance b/w

Thus The estimated value function after k+1 iterations & the true value function is bounded by $\frac{\gamma\epsilon}{1-\gamma}$.

Q.1 (b).

We know that Bellman evaluation operator L^π for any given policy π is defined as

$$L^\pi(V)(s) = E\left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s')\right].$$

where $a = \pi(s)$ = action chosen in state s.
γ = discount factor

To prove:- $U(s) \leq V(s)$ for all $s \in S$. $\Rightarrow L^\pi(U)(s) \leq L^\pi(V)$ for all $s \in S$.

Let U & V be two value functions s.t $U(s) \leq V(s)$ for all states $s \in S$.

We apply Bellman evaluation operator L^π to both U & V.

$$L^\pi[U] = E\left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) U(s')\right]$$

Similarly Bellman operator applied to V is

$$L^\pi(V) = E\left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s')\right]$$

Substracting

$$L^\pi(V) - L^\pi(U) = \gamma \sum_{s'} P(s'|s,a)\left(V(s') - U(s')\right)$$

Since $V(s') \leq U(s')$ for all $s' \in S$ Hence we know that $V(s') - U(s') \geq 0$ for all s'.

Therefore

$$L^n(V) - L(V) \geq 0$$

$$\Rightarrow L^n(V)(s) \geq L^n(U) \quad \text{for all } s \in S.$$

Hence this proves that Bellman evaluation operation L^n is monotonic

Q→1(c).

∴ The initial weight for n=1

$$w_1 = (1-\lambda).$$

weight after n steps

$$w_n = (1-\lambda)\lambda^{n-1}.$$

Given $w_{n(\lambda)} = \frac{1}{2} w_1$

Substituting values for $w_{n(\lambda)}$ & w_1

$$(1-\lambda)\lambda^{n n(\lambda)-1} = \frac{1}{2}(1-\lambda).$$

$$\Rightarrow \lambda^{n(\lambda)-1} = \frac{1}{2}$$

$$(n(\lambda)-1)\log\lambda = \log(1/2)$$

$$\Rightarrow (n(\lambda)-1) = \frac{\log(1/2)}{\log\lambda}$$

$$\Rightarrow n(\lambda) = 1 + \frac{\log(1/2)}{\log\lambda}$$

$$\Rightarrow n(\lambda) = 1 - \frac{\log 2}{\log\lambda}.$$

Now as asked

Value of λ such that weights decay to half of the initial value after 3 steps.

i.e $n(t) = 3$.

$$3 = \frac{i - \log 2}{\log \lambda}$$

$$\Rightarrow \frac{\log 2}{\log \lambda} = -2$$

$$\Rightarrow \log \lambda = -\frac{\log 2}{2}$$

$$\Rightarrow \lambda = 2^{-1/2} = \frac{1}{\sqrt{2}} = 0.707$$

Q) 1(d).

We know the Q learning update formula as

$$Q(s,a) = Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Transitions.

1. $s = C$, $a = jump$, $r = 4$, $s' = E$.
2. $s = E$, $a = right$, $r = 1$, $s' = F$.
3. $s = f$, $a = left$, $r = -2$, $s' = E$.
4. $s = E$, $a = right$, $r = +1$, $s' = f$.

Applying Q learning formula to each transition.

Transition 1

$s = C$, $a = \text{jump}$, $r = 4$, $s' = E$.

$$Q(C, \text{'jump'}) = -10 + 0.5 * (4 + 1 * (-10) - (-10))$$

$$= -10 + 0.5 * 4 = -10 + 2 = -8.$$

Transition 2

$s = E$, $a = \text{right}$, $r = 1$, $s' = f$.

$$Q(E, \text{'right'}) = -10 + 0.5 * (1 + 1 * (-10) - (-10))$$

$$= -10 + 0.5 * 1$$

$$= -10 + 0.5 = -9.5.$$

Transition 3

$$Q(f, \text{'left'}) = -10 + 0.5(-2 + 1 * (-9.5) - (-10))$$

$$= -10 + 0.5 * (-1.5)$$

$$= -10 - 0.75 = -10.75.$$

Transition 4

$$Q(E, \text{'right'}) = -9.5 + 0.5(1 + 1 * (-10) - (-9.5))$$

$$= -9.5 + 0.5 * 0.5$$

$$= -9.25.$$

Final Q values.

Initial	Q(C, 'left')	Q(C, 'jump')	Q(E, 'left')	Q(E, 'right')	Q(E, 'left')	Q(f, 'right')
Initial	-10	-10	-10	-10	-10	-10
1		-8				
2				-9.5		
3					-10.75	
4				-9.25.		

Greedy Policy

To choose greedy policy we choose action that has the highest Q-value in each state.

for C max Q(C, 'a') = Q(C, 'jump') = -8.

E highest Q value is Q(E, 'right') = -9.25

f highest Q value is Q(f, 'right') = -10

Hence greedy policy is C → jump
E → right
f → right.