Date: __/__/____

CSE564 - Reinforcement Learning
HW 2

Name: Arsh Arora
Roll No.: 2019022

① Given arms 1 to 10 with Gaussian distributions with mean 1 to 10 respectively and variance 1.

$$p[a] = \begin{cases} 2/15 & a = 2, 4, 6, 8, 10 \\ 1/15 & a = 1, 3, 5, 7, 9 \end{cases}$$

Expected Reward on each pick

$$= E[q_*(a)]$$
$$= \sum p(a) \times q_*(a) \quad \left(q_*(a) = E[R_t | A_t = a]\right)$$
$$= \frac{2}{15} \times (2+4+6+8+10) + \frac{1}{15} \times (1+3+5+7+9)$$
$$= \frac{2}{15} \times (30) + \frac{1}{15}(25)$$
$$= 4 + \frac{5}{3} = \frac{17}{3}$$

Total reward after 10 picks $= 10 \times E[q_*(a)]$

$$= 170/3 = 56.66$$

② $q_*(a) = 0 \times 0.5 + 1 \times 0.5$
$= 0.5 \quad \forall a \in \{1, 2, 4, 5, 7, 9, 10\}$

Ily, $q_*(a) = 0 \times 0.3 + 0.2 \times 0.3 + 1 \times 0.4$
$= 0.46 \quad \forall a \in \{3, 6, 8\}$

Date: __/__/____

Thus, for stochastic optimal policies, we need to make use of arms with $q_*(a) = 0.5$.
Any combination of these arms would be optimal
Six such -

Arm #

1) Select 9 and 10 with 0.5 each
2) Select 7, 9 and 10 with 0.33 each
3) Select 5, 7, 9 and 10 with 0.25 each
4) Select 4, 5, 7, 9 and 10 with 0.2 each
5) Select 1 with 0.3 and 2 with 0.7
6) Select 1 with 0.1, 2 with 0.1 and 4 with 0.8.

(3) Let at time $t=1$,
$Q_1(1) = 0.2$, $Q_1(2) = 0.3$ $Q_1(3) = 0.5$
In the given ε-greedy -
1) We explore at $t = 1, 3, 5$
2) We exploit at $t = 2, 4, 6$
3) We explore over non-greedy actions only.

At $t=1$, explore among arms 1, 2 (non-greedy)
Suppose we select Arm #1.
Let it return value 1.
Then, after $t=1$, $A_1 = 1$, $R_1 = 1$
$Q_2(1) = \frac{0.2+1}{2} = 0.6$, $Q_2(2) = 0.3$, $Q_2(3) = 0.5$

At $t=2$, choose greedy -
We select arm #1.
Let it return value 0.
Then, after $t=2$, $A_2 = 1$, $R_2 = 0$
$Q_3(1) = \frac{0.6+0}{3} = 0.2$, $Q_3(2) = 0.3$, $Q_3(3) = 0.5$

Date: __/__/____

At $t=3$, explore from arms 1, 2
We select arm #2.
Let it return value 0.
Then, after $t=3$, $A_3 = 2$, $R_3 = 0$
$Q_4(2) = \frac{0.3+0}{2} = 0.15$, $Q_4(1) = 0.2$ $Q_4(3) = 0.5$

At $t=4$, choose greedy -
We select arm #3.
Let it return value 1.
Then after $t=4$, $A_4 = 3$, $R_4 = 1$
$Q_5(3) = \frac{0.5+1}{2} = 0.75$, $Q_5(1) = 0.2$, $Q_5(2) = 0.15$

At $t=5$, choose from arms (1, 2) (explore)
We select arm #1.
Let it return value 1.
After $t=5$, $A_5 = 1$, $R_5 = 1$
$Q_6(1) = \frac{0.2+1}{4} = 0.3$, $Q_6(2) = 0.15$, $Q_6(3) = 0.75$

At $t=6$, greedy.
We select arm #3
It returns value 0.
After $t=6$, $A_6 = 3$, $R_6 = 0$.

Date: __/__/____

(4)

s	a	s'	r	$p(s', r \mid s, a)$
high	search	high	r_{search}	α
high	search	low	r_{search}	$1-\alpha$
high	wait	high	r_{wait}	1
low	search	high	-3	$1-\beta$
low	search	low	r_{search}	β
low	wait	low	r_{wait}	1
low	recharge	high	0	1

This table was derived from the MDP table and using the transition graph given for the example 3.3.
Here, in the case where s is low and s' is high, we assume that the robot got discharged and to get recharged again took a -ve reward of -3.

(5) Prove using (3.8) that adding a constant c to all rewards only adds a constant v_c to values of all states, and thus doesn't affect relative values of any states under any policies. Write v_c in terms of c and γ.

For a given state at time t,

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots$$

$$G_{t,new} = (c + R_{t+1}) + (\gamma(c + R_{t+2})) + (\gamma^2(c + R_{t+3})) + \dots$$

$$= (R_{t+1} + \gamma R_{t+2} + \dots) + (c + \gamma c + \gamma^2 c + \dots)$$

$$\Rightarrow G_{t,new} = G_t + \underbrace{\frac{c}{1-\gamma}}_{v_c}$$

∴ Relative values remain unchanged.

Date: __/__/____

Add a constant c to an episodic task. What effect does it have?

For an episodic task,

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-(t+1)} R_T$$ where T is final time step.

Now, progressing like before

$$G_{t,new} = (c + R_{t+1}) + \gamma(c + R_{t+2}) + \dots + \gamma^{T-(t+1)}(c + R_T)$$

$$= (c + \gamma c + \dots + \gamma^{T-(t+1)} c) + G_t$$

$$= c\,\frac{1-\gamma^{T-(t+1)}}{1-\gamma} + G_t$$

$$\underbrace{\quad}_{v_c}$$

∴ If the value of T increases, the value of v_c would also increase.
So, the added constant would promote the agent to take more steps, thus having an effect.

Date: __/__/____

(8) Equation for v_* in terms of q_*.

$$v_*(s) = \max_{a \in A(s)} q_{\pi^*}(s,a)$$

$$= \max_{a \in A(s)} \sum_{s',r} p(s',r \mid s,a) \left[r + \gamma \max_{a' \in A(s')} q_*(s',a') \right]$$

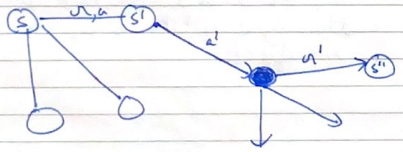
(11) We need to figure out if random variable R_{t+2} is dependent on S_t and A_t, and if yes then how.

Let $r' = R_{t+2}$ and $r = R_{t+1}$.

Then,

$$P_\pi(r = R_{t+2} \mid S_t = s, A_t = a)$$

$$= \sum_{\substack{s' \in S \\ r \in R}} p(s',r \mid s,a) \sum_{\substack{a' \in A(s') \\ s'' \in S}} \pi(a' \mid s') \cdot p(s'',r \mid s',a')$$



Date: __/__/____

(12) $E_\pi[R_{t+2} \mid S_t = s, A_t = a]$

$$= \sum_{r'} r' \cdot P_\pi(r = R_{t+2} \mid S_t = s, A_t = a)$$

$$= \sum_{r'} r' \sum_{s',a} p(s',r \mid s,a) \sum_{a'',s''} \pi(a' \mid s') p(s'',r' \mid s',a')$$

(13) $v_\pi(s) = E[G_t \mid S_t = s]$

$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$

$= E_\pi[R_{t+1} \mid S_t = s] + \gamma E_\pi[G_{t+1} \mid S_t = s]$

$= E_\pi[R_{t+1} \mid S_t = s] + \gamma E_\pi[E_\pi[G_{t+1} \mid S_{t+1} = s'] \mid S_t = s]$

$= E_\pi[R_{t+1} \mid S_t = s] + \gamma E_\pi[v_\pi(s') \mid S_t = s]$ (s' = S_{t+1})

$= E_\pi[R_{t+1} + \gamma v_\pi(s') \mid S_t = s]$

$= \sum_{a \in A(s)} \pi(a|s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$

Bellman eq^n for $v_\pi(s)$

(14) We start from the back. $\gamma = 0.5$

$G_3 = R_4 + \gamma G_4$
$= -3 + 0.5 \times 0$
$= -3$

$G_2 = R_3 + \gamma G_3$
$= 10 + 0.5 \times (-3)$
$= 8.5$

$G_1 = R_2 + \gamma G_2$
$= -1 + 0.5 \times (8.5)$
$= 3.25$

$G_0 = R_1 + \gamma R_1$
$= 2 + 0.5 \times 3.25$
$= 2 + 1.625$
$= 3.625$

If an agent receives a constant reward of c,

$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots$

$= c + \gamma c + \gamma^2 c + \dots$
$= c(1 + \gamma + \gamma^2 + \dots)$
$= c\left(\frac{1}{1-\gamma}\right)$
$= \frac{c}{1-\gamma}$

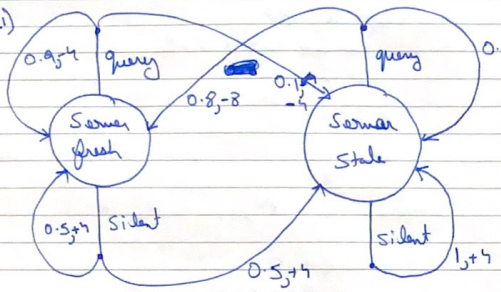
(15) $\pi^*(s) = \underset{a \in A(s)}{\text{argmax}}\ q_*(s, a)$

$\Rightarrow \pi_*(s) = \underset{a \in A(s)}{\text{argmax}} \sum_r \sum_{s'} p(s', r \mid s, a)[r + \gamma \cdot v_*(s')]$

Thus, what we do is simply go over all actions that can be taken at a given state s and find the best action that returns the highest value.

Date: __/__/____

(16) a)



∴ The MDP-

s	a	s'	r	p(s',r|s,a)
fresh	query	fresh	-4	0.9
fresh	query	stale	-4	0.1
fresh	silent	fresh	+4	0.5
fresh	silent	stale	+4	0.5
stale	query	fresh	-8	0.8
stale	query	stale	-8	0.2
stale	silent	stale	+4	1

b) Value iteration- $\gamma = 0.5$

Initialization $V(\text{fresh}) = 0$, $V(\text{stale}) = 0$

$$V_{k+1}(s) = \max_a \sum_{s',r} p(s',r|s,a)\,(r + \gamma V_k(s'))$$

1st iteration-

$$V(\text{fresh}) = \max\{0.9\times(-4+0.5\times0)+0.1(-4+0.5\times0),\ 0.5\times(4+1/2\times0)+0.5(4+1/2\times0)\}$$

$$= \max\{-4, 4\} = 4$$

$$V(\text{stale}) = \max\{0.8\times(-8+0.5\times0)+0.2(-8+0.5\times0),\ 1\times(4+0.5\times0)\}$$

$$= \max\{-8, 4\} = 4$$

$\therefore V(\text{fresh}) = 4$, $V(\text{stale}) = 4$.

2nd iteration-

$$V(\text{fresh}) = \max\{0.9\times(-4+0.5\times4)+0.1\times(-4+0.5\times4),\ 0.5\times(4+0.5\times4)+0.5\times(4+0.5\times4)\}$$

$$= \max\{-2, 6\} = \boxed{6}$$

$$V(\text{stale}) = \max\{0.8\times(-8+0.5\times4)+0.2(-8+0.5\times4),\ 1\times(4+0.5\times4)\}$$

$$= \{-6, 6\} = \boxed{6}$$

3rd iteration-

$$V(\text{fresh}) = \max\{0.9\times(-4+0.5\times6)+0.1(-4+0.5\times6),\ 0.5\times(4+0.5\times6)+0.5(4+0.5\times6)\}$$

$$= \max\{-1, 7\} = 7$$

$$V(\text{stale}) = \max\{0.8\times(-8+0.5\times6)+0.2(-8+0.5\times6),\ 1\times(4+0.5\times6)\}$$

$$= \max\{-5, 7\} = 7$$

4th iteration-

$$V(\text{fresh}) = \max\{0.9\times(-4+0.5\times7)+0.1\times(-4+0.5\times7),\ 0.5\times(4+0.5\times7)+0.5\times(4+0.5\times7)\}$$

$$= \max\{-0.5, 7.5\} = 7.5$$

Date: __/__/____

#learnthesmarterway

Date: __/__/____

$v(\text{stale}) = \max\left\{ -8 + \tfrac{1}{2}(0.8\times 7 + 0.2\times 7),\ 4 + \tfrac{1}{2}\times 1\times 7 \right\}$

$= \max\{-4.5, 7.5\} = 7.5$

∴ Final after 4 iterations,

$v(\text{fresh}) = 7.5 \quad v(\text{stale}) = 7.5$

Policy Iteration –

$$v_{k+1}(s) = \sum_{a\in A(s)} \pi_k(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_{\pi_k}(s')]$$

and then,

$$\pi_{k+1}(s) = \operatorname*{argmax}_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_{\pi_k}(s')]$$

$v(\text{fresh}) = \pi(\text{query}|\text{fresh}) \times [0.9(-4 + 0.5 v_k(\text{fresh})) + 0.1\times(-4 + 0.5 v_k(\text{stale}))]$

$+ \pi(\text{quiet}|\text{fresh}) \times [0.5\times(4 + 0.5 v_k(\text{fresh})) + 0.5\times(4 + 0.5\times v_k(\text{stale}))]$

Ildy, $= \pi(\text{query}|\text{fresh})[-4 + 0.5(\overset{0.9\times}{v_k(\text{fresh})} + \overset{0.1\times}{v_k(\text{stable})})]$

$+ \pi(\text{quiet}|\text{fresh})[4 + 0.5(v_k(\text{fresh}) + v_k(\text{stale}))]$

$v(\text{stale}) = \pi(\text{query}|\text{stale})[0.8\times(-8 + 0.5 v_k(\text{fresh})) + 0.2\times(-8 + 0.5 v_k(\text{stale}))]$

$+ \pi(\text{quiet}|\text{stale})[4 + 0.5 v_k(\text{stale})]$

$= \pi(\text{query}|\text{stale})[-8 + 0.5(\overset{0.8\times}{v_k(\text{fresh})} + \overset{0.2}{v_k(\text{stale})})]$

$+ \pi(\text{quiet}|\text{stale})[4 + 0.5 v_k(\text{stale})]$

#learnthesmarterway

1st Iteration

Policy evaluation –

Let π_0 be –

$\pi_0(\text{query}|\text{stale}) = 0$

$\pi_0(\text{query}|\text{fresh}) = 0$ $\quad v_{\pi_0}(\text{stale}) = 0 \to$ fg.

$\pi_0(\text{quiet}|\text{stale}) = 1$ $\quad v_{\pi_0}(\text{fresh}) = 0 \to$ fg.

$\pi_0(\text{quiet}|\text{fresh}) = 1$

$v_{\pi_0}(\text{fresh}) = x = 4 + 0.25x + 0.25y$

Ildy $\quad y = 4 + 0.5y.$

$\Rightarrow y = 8, \quad x = 8$

$v_{\pi_0}(\text{stale}) \qquad v_{\pi_0}(\text{fresh})$

Policy Improvement –

$\pi_1(\text{fresh}) = \operatorname*{argmax}_{\text{query, quiet}} \left\{ -4 + 0.5(0.9\times 8 + 0.1\times 8),\ 4 + 0.5(8+8) \right\}$

$= \operatorname{argmax}\{0, 8\}.$

$\therefore \pi_1(\text{fresh}) = \text{quiet} \Rightarrow \pi_1(\text{quiet}|\text{fresh}) = 1$

$\pi_1(\text{stale}) = \operatorname*{argmax}_{\text{query, quiet}} \left\{ -8 + 0.5(0.8\times 8 + 0.2\times 8),\ 4 + 0.5(8) \right\}$

$= \operatorname{argmax}\{-4, 8\} = \text{quiet}$

$\therefore \pi_1(\text{stale}) = \text{quiet} \Rightarrow \pi_1(\text{quiet}|\text{stale}) = 1$

$\therefore \pi_0 = \pi_1$, by policy improvement theorem we have achieved the optimal policy.

#learnthesmarterway

Date: __/__/____

(17) Proving the correctness of policy improvement step -

$v_\pi(s) \le q_\pi(s, \pi'(s))$ where $\pi'(s)$ is policy derived from DP.

$= E[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)]$

$= E_{\pi'}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$

$\le E_{\pi'}[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s]$

$= E_{\pi'}[R_{t+1} + \gamma E_{\pi'}[R_{t+2} + \gamma v_\pi(S_{t+2}) \mid S_{t+1}, A_{t+1} = \pi'(S_{t+1})] \mid S_t = s]$

$= E_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) \mid S_t = s]$

⋮

On further expanding

⋮

$\le E_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} \dots \mid S_t = s]$

$= E_{\pi'}[G_t \mid S_t = s]$

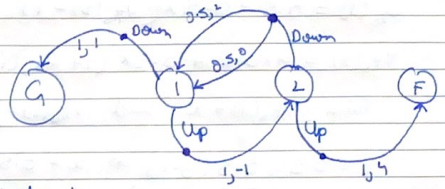
$= v_{\pi'}(s)$

$\therefore v_\pi(s) \le v_{\pi'}(s)$ if $\pi(s) \le \pi'(s)$

#learnthesmarterway

Date: __/__/____

(18) Modelling taking a stair well of 2 steps b/w G and F.



Initialization -

$\pi(\text{Up} \mid 1) = 0.5 \quad \pi(\text{Up} \mid 2) = 0.5$

$\pi(\text{Down} \mid 1) = 0.5 \quad \pi(\text{Down} \mid 2) = 0.5$

$v_{\pi_0}(s) = 0 \quad \forall s \in \{G, 1, 2, F\}$.

Up = ↑ Down = ↓

General forms -

$$v_\pi(s) = \sum_{a \in A(s)} \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$

($\gamma = 1$)

$v_\pi(1) = \pi(\uparrow \mid 1)(1 \times (-1 + v_\pi(2))) + \pi(\downarrow \mid 1)(1 \times (1 + v_\pi(G)))$

$= \pi(\downarrow \mid 1) + \pi(\uparrow \mid 1)(v_\pi(2) - 1)$

$v_\pi(2) = \pi(\uparrow \mid 2)(1 \times (4 + v_\pi(F))) + \pi(\downarrow \mid 2)(0.5 \times (2 + v_\pi(1)) + 0.5 \times (0 + v_\pi(1)))$

$= 4\pi(\uparrow \mid 2) + (1 + v_\pi(1)) \times \pi(\downarrow \mid 2)$

$= 4\pi(\uparrow \mid 2) + \pi(\downarrow \mid 2)(1 + v_\pi(1))$

#learnthesmarterway

Date __/__/____

1st iteration

$V_{\pi_0}(1) = 0.5 + 0.5 \times (V_{\pi_0}(2) - 1)$

$V_{\pi_0}(2) = 4 \times 0.5 + 0.5 \times (1 + V_{\pi_0}(1))$

Taking $V_{\pi_0}(1) = x$ and $V_{\pi_0}(2) = y$

$x = 0.5 + 0.5(y-1) \Rightarrow x = 0.5y$

$y = 2 + 0.5(1+x) \Rightarrow y = 2.5 + 0.5x$

$y = 2.5 + 0.25y$

$0.75y = 2.5$

$y = \frac{2.5}{0.75} = \frac{10}{3}$

$x = 0.5y = \frac{5}{3}$

$\therefore V_{\pi_0}(1) = \frac{5}{3}$

$V_{\pi_0}(2) = \frac{10}{3}$

Policy improvement-

$\pi_1(1) = \underset{\uparrow,\downarrow}{\text{argmax}} \{1 \times (10/3 - 1), 1\}$

$= \text{argmax} \{7/3, 1\} = \uparrow$

$\pi_1(2) = \underset{\uparrow,\downarrow}{\text{argmax}} \{4, 1 + 5/3\}$

$= \text{argmax} \{4, 8/3\} = \uparrow$

$\therefore \pi_1(\uparrow|1) = 1 \quad \pi_1(\downarrow|1) = 0$

$\pi_1(\uparrow|2) = 1 \quad \pi_1(\downarrow|2) = 0$

Date __/__/____

2nd iteration

$V_{\pi_1}(1) = 0 + 1 \times (V_{\pi_1}(2) - 1)$

$V_{\pi_1}(2) = 1 \times 4 + 0 = 4$

$\therefore V_{\pi_1}(1) = 3, \quad V_{\pi_1}(2) = 4$

Policy improvement-

$\pi_2(1) = \underset{\uparrow,\downarrow}{\text{argmax}} \{1 \times (4-1), 1\}$

$= \text{argmax} \{3, 1\} = \uparrow$

$\pi_2(2) = \underset{\uparrow,\downarrow}{\text{argmax}} \{4, 1+3\} = \uparrow, \downarrow$

$\therefore \pi_2(\uparrow|1) = 1 \quad \pi_2(\downarrow|1) = 0$

$\pi_2(\uparrow|2) = 0.5 \quad \pi_2(\downarrow|2) = 0.5$

3nd iteration

$V_{\pi_2}(1) = 0 + 1 \times (V_{\pi_2}(2) - 1)$

$V_{\pi_2}(2) = 4 \times 0.5 + 0.5 \times (1 + V_{\pi_2}(1))$

$\Rightarrow x = y - 1$ and $y = 2.5 + \frac{x}{2}$

$x + 1 = 2.5 + \frac{x}{2}$

$\frac{x}{2} = 1.5 \Rightarrow x = 3$

$y = 4.$

$V_{\pi_2}(1) = 3 \quad V_{\pi_2}(2) = 4.$

Date: __/__/____

Policy improvement -

$\pi_3(1) = \underset{\uparrow,\downarrow}{\text{argmax}} \{1\times(4-1), 1\} = \uparrow$

$\pi_3(2) = \underset{\uparrow,\downarrow}{\text{argmax}} \{4, 1+3\} = \uparrow,\downarrow$

$\therefore \pi_3 = \pi_2$

$\therefore$ By ~~optimal~~ policy improvement theorem,
$\pi_2 = \pi_3 = \pi_*$.

$\therefore$ Optimal policy is -

$\pi_*(\uparrow|1) = 1 \qquad \pi_*(\uparrow|2) = 0.5$
$\pi_*(\downarrow|1) = 0 \qquad \pi_*(\downarrow|2) = 0.5$

The states above 1 shows an equal policy for both ↑ and ↓. This is because $\gamma = 1$, which means that the number of steps taken to reach the top don't play a role. Every policy where we don't go down from 1 plays out and returns the exact same value.

Date: __/__/____

(19) Let H = Healthy, S = Sick.