REINFORCEMENT LEARNING - CSE564

Homework 2

Suchet Aggarwal

2018105

**** Note: All code questions are implemented in 2018105_code.ipynb, with comments specifying each question number.**

Q1.

Q1.

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

In general;

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

$$= \frac{\frac{p(r, s', s, a)}{p(s,a)}}{\frac{p(s', s, a)}{p(s,a)}}$$

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

$$\Rightarrow \quad p(s', r \mid s, a) = p(s' \mid s, a) \cdot p(r \mid s', s, a)$$

In this example the reward is fixed for given transition $s \xrightarrow{a} s'$, $\therefore$ $p(r \mid s', s, a) = 1$

$$\therefore \quad p(s', r \mid s, a) = p(s' \mid s, a)$$

Q2. Code Attached

Q3.

Q3. (a) The signs of the rewards aren't important rather the intervals between them are.

From Eq (3.8)

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$
$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

adding c to each R_i $\forall\, i$

$$G_t' = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} + c)$$
$$= G_t + c \cdot \sum_{k=0}^{\infty} \gamma^k$$
$$= G_t + \frac{c}{1-\gamma}$$

Now $v_\pi(s) = E(G_t \mid S_t = s)$

$$v_\pi'(s) = E(G_t' \mid S_t = s)$$
$$= E\left(G_t + \frac{c}{1-\gamma} \mid S_t = s\right)$$
$$= E(G_t \mid S_t = s) + \frac{c}{1-\gamma}$$

(Second term is constant)

$$v_\pi'(s) = v_\pi(s) + \frac{c}{1-\gamma}$$

The last step proves adding constant c to all Rewards just adds a constant v_c to all $v_\pi(s)$, and doesn't affect relative values

Q3. (b)

In case of an Episodic task, the signs of reward are important, mainly because it conveys to the agent to complete the task within a time frame by giving negative rewards.
If such rewards are made positive the length of the Episode may change that can alter the relative difference in rewards.

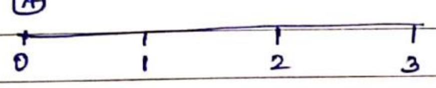
$$G_t = R_{t+1} + R_{t+2} + \cdots \quad R_{t+T}$$

$$G'_t = (R_{t+1} + c) + \cdots \quad (R_{t+k} + c)$$

~~here K ≠ T for all states~~

here, k may not be same as T for $s \in S$
thus the V_π values would get changed without maintaining proper relative order.

For Example : Consider a robot (agent), and the task is to move from position 0 to position 3, with allowed actions left and right.



with reward -1 at all steps.
Here obviously, optimal strategy is to move right always, with return $= -3$
However if all rewards are made $+1$, then practically maximum return becomes infinity with agent just moving between 0, 1, 2

Q4. Code Attached

Q5. The optimal value function is the max over all actions in the optimal action value function

$$v^*(s) = max_{a \in A(s)} q^*(s, a) \quad \forall s \in S$$

Q6. Code Attached

Q7. Code Attached

Q8.

Q8. Yes, R_{t+2} is dependant on S_t, A_t

Consider $P(R_{t+2} \mid S_t, A_t)$

$$= \sum_{S_{t+2}} P(S_{t+2}, R_{t+2} \mid S_t A_t)$$

$$= \sum_{S_{t+2}} \sum_{S_{t+1}} \sum_{R_{t+1}} \sum_{A_{t+1}} P(S_{t+2}, R_{t+2} \mid S_{t+1}, R_{t+1}, A_{t+1}, S_t, A_t) \cdot P(S_{t+1}, R_{t+1}, A_{t+1} \mid S_t, A_t)$$

$$= \sum_{S_{t+2}} \sum_{S_{t+1}} \sum_{R_{t+1}} \sum_{A_{t+1}} P(S_{t+2}, R_{t+2} \mid S_{t+1}, A_{t+1}) \cdot P(A_{t+1} \mid S_{t+1}) \cdot P(S_{t+1}, R_{t+1} \mid S_t A_t)$$

(by markov assumption :

$P(S_{t+2}, R_{t+2} \mid S_{t+1}, R_{t+1}, A_{t+1}, S_t, A_t) =$

$P(S_{t+2}, R_{t+2} \mid S_{t+1}, A_{t+1})$

probability is independent of previous state reward)

and ($P(S_{t+1}, R_{t+1}, A_{t+1} \mid S_t, A_t) =$

$P(A_{t+1} \mid S_{t+1}) \cdot P(S_{t+1}, R_{t+1} \mid S_t, A_t)$

as action at t+1 depends only on S_{t+1})

$$P(R_{t+2} \mid S_t, A_t) = \sum_{S_{t+2}} \sum_{S_{t+1}} \sum_{R_{t+1}} \sum_{A_{t+1}} P(S_{t+2}, R_{t+2} \mid S_{t+1}, A_{t+1}) \cdot \pi(A_{t+1} \mid S_{t+1}) \cdot P(S_{t+1}, R_{t+1} \mid S_t, A_t)$$

Q9.

Q9.

$$E(R_{t+2} \mid S_t, A_t) =$$

$$= \sum_{r} r \cdot p(R_{t+2} = r \mid S_t, A_t)$$

from Question 8.

$$= \sum_{r} r \cdot \sum_{S_{t+2}} \sum_{S_{t+1}} \sum_{R_{t+1}} \sum_{A_{t+1}} p(S_{t+2}, R_{t+2} = r \mid S_{t+1}, A_{t+1})$$

$$\cdot \pi(A_{t+1} \mid S_{t+1})$$

$$\cdot p(S_{t+1}, R_{t+1} \mid S_t, A_t)$$

(For ease of read; read
S_{t+2} as $S_{t+2} = s''$
R_{t+2} as $R_{t+2} = r$
R_{t+1} as $R_{t+1} = r'$
A_{t+1} as $A_{t+1} = a'$
S_{t+1} as $S_{t+1} = s'$
S_t as $S_t = s$
A_t as $A_t = a$)

Q10.

$$
\begin{aligned}
v_\pi(s) &= E_\pi(G_t \mid S_t = s) \\
&= E_\pi(R_t + \gamma\, G_{t+1} \mid S_t = s) \\
&= \sum_a \pi(\mathrm{a}|\mathrm{s}) \sum_{s',r} p(s', r \mid s, a)\, [r + \gamma E_\pi(R_t + \gamma\, G_{t+1} \mid S_{t+1} = s')] \\
&= \sum_a \pi(\mathrm{a}|\mathrm{s}) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')]
\end{aligned}
$$

Q11.

classmate

Date

Page

Q11. Given $R_1 = 2$, $R_2 = -1$, $R_3 = 10$, $R_4 = -3$

Also

$$G_t = R_{t+1} + \gamma G_{t+1}$$

$$G_4 = 0$$

$$\Rightarrow \quad G_3 = -3$$

$$G_2 = 10 + 0.5(-3) = 8.5$$

$$G_1 = -1 + 0.5(8.5) = 3.25$$

$$G_0 = 2 + 0.5(3.25) = 3.625$$

(Considering the episode length to be 4)

Now in generality:

$$G_t = \sum_{k=0}^{\infty} \gamma^k . R_{t+1+k}$$

Here $R_{t+1+k} \; \forall k = c$

$$\therefore \quad G_t = c . \sum_{k=0}^{\infty} \gamma^k$$

$$G_t = \frac{c}{1-\gamma}$$

Q12. The optimal policy can be obtained by greedily picking all actions from a state a that maximise the overall return using the optimal value function itself as follows:

$$\pi^*(s) = argmax_{a \in A(s)}(E(R_t \ + \ \gamma v^*(s') \, | S_t = s, A_t \ = \ a)) \quad \forall s \ \in S$$

Q13.

TO-DO

Q14.

Q14. <u>To Prove:</u>

1) Policy improvement step either improves the policy or the current policy is optimal.

Equivalently we prove:

① Policy improvement either improves the policy or leaves it unchanged.

② If the policy is unchanged, then current policy is optimal.

Proof for ①:

From policy evaluation

$$v_{\pi_k}(s) = T_{\pi_k}(v_{\pi_k}(s))$$

After policy improvement. $T_{\pi_{k+1}} v_{\pi_k}(s) = T v_{\pi_k}$

$$T_{\pi_{k+1}}(v_{\pi_k}(s)) \geq v_{\pi_k}(s)$$

$$\left(\text{i.e. } q_{\pi_k}(s, \pi_{k+1}(s)) \geq q_{\pi_k}(s, \pi_k(s)) = v_{\pi_k}(s)\right)$$

Since $T_{\pi_{k+1}}$ is monotonically increasing

$$T^2_{\pi_{k+1}}(v_{\pi_k}(s)) \geq T_{\pi_{k+1}}(v_{\pi_k}(s)) \geq v_{\pi_k}(s)$$

$$\vdots$$

$$T^N_{\pi_{k+1}}(v_{\pi_k}(s)) \geq v_{\pi_k}(s)$$

as $N \to \infty$ $\quad T^{N}_{\pi_{k+1}}(v_{\pi_k}(s)) = v_{\pi_{k+1}}$

(i.e. $\because v_{\pi_{k+1}}$ is a stationary pt. of $T_{\pi_{k+1}}$)

$$\Rightarrow \lim_{N \to \infty} T^{N}_{\pi_{k+1}}(v_{\pi_k}(s)) \geq v_{\pi_k}(s)$$

$$\Rightarrow v_{\pi_{k+1}}(s) \geq v_{\pi_k}(s) \quad \forall s$$

$\therefore$ π_{k+1} (new policy) improves the value function (for π_k) or leaves it unchanged.

Proof for ②

for policy improvement

$$T_{\pi_{k+1}}(v_{\pi_k}(s)) = T(v_{\pi_k}(s))$$

(i.e. $\pi_{k+1}(s) = \underset{a \in A(s)}{\arg\max}\ E(R_{t+1} + \gamma v_{\pi}(s') \mid S_t = s, A_t = a)$)

Suppose policy remains unchanged

i.e. $v_{\pi_k}(s) = v_{\pi_{k+1}}(s)$

$$\Rightarrow T_{\pi_{k+1}}(v_{\pi_k}(s)) = T_{\pi_{k+1}}(v_{\pi_{k+1}}(s))$$

(stationary pt.) $\quad = v_{\pi_{k+1}}(s)$

$$= v_{\pi_k}(s)$$

Also

$$T_{\pi_{k+1}}(v_{\pi_k}(s)) = T(v_{\pi_k}(s))$$

$$v_{\pi_k}(s) = T(v_{\pi_k}(s))$$

i.e. v_{π_k} is a stationary pt for T

i.e.

$$v_{\pi_k}(s) = \max_{a \in A(s)} \sum_{s',r} p(s',r \mid s,a)\left(r + \gamma v_{\pi_k}(s')\right)$$

which is the bellman optimality condition.

$\therefore$ π_k is the optimal policy

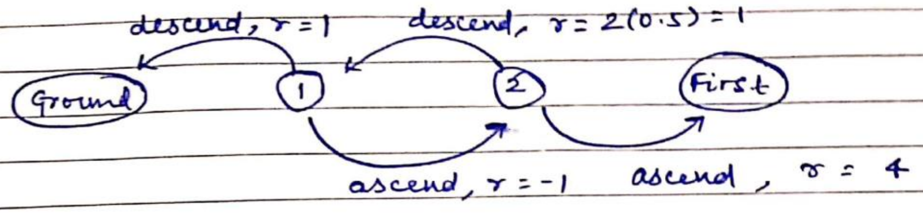
($\because$ V_{π_k} is the optimal value function)

$$\therefore \quad \pi^* = \pi_k$$

$$V^* = V_{\pi_k}$$

Q15.

Q15. for $n = 2$
the MDP is given by :



Also the robot ascends, descends with equal prob = 0.5

$\therefore \quad \pi_0(\text{ascend} \mid S = 1) = 0.5$
$\pi_0(\text{descend} \mid S = 1) = 0.5$
$\pi_0(\text{ascend} \mid S = 2) = 0.5$
$\pi_0(\text{descend} \mid S = 2) = 0.5$

Policy iteration :

iteration 1 : Policy evaluation :

$V_{\pi_0}(\text{Ground}) = V_{\pi_0}(\text{First}) = 0$ (terminal states)

$$V_{\pi_0}(1) = \pi(\text{descend} \mid 1)(1 + V_{\pi_0}(\text{Ground})) + \pi(\text{ascend} \mid 1)(-1 + V_{\pi_0}(2))$$

$$V_{\pi_0}(2) = \pi(\text{descend} \mid 2)\cdot(1 + V_{\pi_0}(1)) + \pi(\text{ascend} \mid 2)\cdot(4 + V_{\pi_0}(\text{first}))$$

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

Solving $V_{\pi_0}(1)$ and $V_{\pi_0}(2)$

$$V_{\pi_0}(1) = 0.5\, V_{\pi_0}(2)$$
$$V_{\pi_0}(2) = 2.5 + 0.5\, V_{\pi_0}(1)$$

$$\Rightarrow \quad V_{\pi_0}(1) = \frac{5}{3}$$
$$V_{\pi_0}(2) = \frac{10}{3}$$

Policy improvement :

$$\pi_1(1) = \underset{\{\text{descend}, \text{ascend}\}}{\text{argmax}} \left(1 , \overbrace{-1 + V_{\pi_0}(2)}^{\frac{7}{3}} \right)$$

$$= \text{ascend}$$

$$\pi_1(2) = \underset{\{\text{descend}, \text{ascend}\}}{\text{argmax}} \left(\underbrace{1 + V_{\pi_0}(1)}_{\frac{8}{3}} , 4 \right)$$

$$= \text{ascend}$$

iteration 2 : Policy eval.

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

$$\Rightarrow \quad V_{\pi_1}(1) = 3$$

Policy update:

$$\pi_2(1) = \underset{\{descend,\, ascend\}}{\text{argmax}} \left(1,\ -1+4 \right)$$

$$= \text{ascend}$$

$$\pi_2(2) = \underset{\{descend,\, ascend\}}{\text{argmax}} \left(v_1(1)+1,\ 4 \right)$$

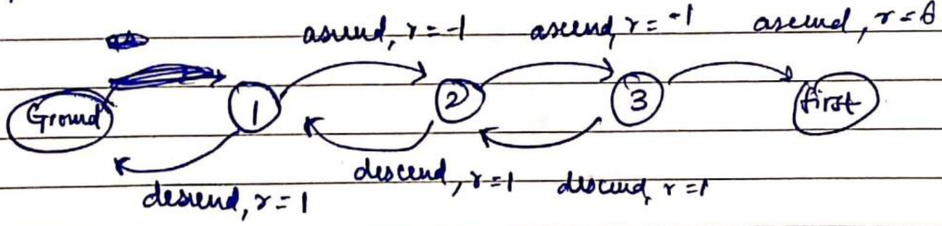
$$= \{\text{ascend},\ \text{descend}\}$$

we pick ascend

* Note if we pick descend, then next time we'll again have a choice, this way $\pi_{k+1} \neq \pi_k$ and policy iteration would never end.

The no. of iterations would always be = 2, for any n. ~~as for the last step~~

Proof: consider $n = 3$



assuming $\pi(asc \mid s) = 0.5$

$\pi(desc \mid s) = 0.5$ for $s \in \{1, 2, 3\}$

iteration 1: Policy eval

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

$$v_{\pi_0}(2) = 0.5(1 + v_{\pi_0}(1)) + 0.5(-1 + v_{\pi_0}(3))$$

$$v_{\pi_0}(3) = 0.5(1 + v_{\pi_0}(2)) + 0.5(6)$$

on solving:

$$v_{\pi_0}(1) = 0.5\, v_{\pi_0}(2)$$

$$v_{\pi_0}(2) = 0.5\, v_{\pi_0}(1) + 0.5\, v_{\pi_0}(3)$$

$$v_{\pi_0}(3) = 3.5 + 0.5\, v_{\pi_0}(2)$$

$$v_{\pi_0}(1) = 7/4$$
$$v_{\pi_0}(2) = 7/2$$
$$v_{\pi_0}(3) = 21/4$$

Policy update

$$\pi_1(1) = \underset{\{d, a\}}{\operatorname{argmax}} \left(1, -1 + 7/2\right)$$
$$= \text{ascend}$$

$$\pi_1(2) = \underset{\{d, a\}}{\operatorname{argmax}} \left(1 + \frac{7}{4}, -1 + \frac{7}{2}\right)$$
$$= \text{ascend}$$

$$\pi_1(3) = \underset{\{d, a\}}{\operatorname{argmax}} \left(1 + \frac{7}{2}, 6\right)$$
$$= \text{ascend}$$

Iteration 2: Policy eval

$$v_{\pi_1}(1) = -1 + v_{\pi_1}(2)$$
$$v_{\pi_1}(2) = -1 + v_{\pi_1}(3)$$
$$v_{\pi_1}(3) = 6$$

$$\Rightarrow$$

$$v_{\pi_1}(1) = 4$$
$$v_{\pi_1}(2) = 5$$
$$v_{\pi_2}(3) = 6$$

Policy update

$$\pi_2(1) = \underset{\{d,a\}}{\text{argmax}} \; (1, \; -1+5)$$

$$= \text{ascend}$$

$$\pi_2(2) = \underset{\{d,a\}}{\text{argmax}} \; (1+4, \; -1+6)$$

* Can pick both, same as $n=2$; we pick ascend for termination

$$= \text{ascend}$$

$$\pi_2(3) = \underset{\{d,a\}}{\text{argmax}} \; (1+5, \; 6)$$

Same for $s=3$

$$= \text{ascend}$$

Thus it ends. after two iterations.
this can be generalised for any n
where always going up is optimal.