Reinforcement Learning

Assignmnet 1

Soroush Naseri
Ferdowsi university of Mashhad

April 13, 2023

Problem 1 .

We calculate the returns using equasion below :

$$G_T = r + \Sigma\gamma^i r_i = \gamma G_{T+1} + r$$

Firstly consider that $r_m = 2$:
13 to 24 :

$$G_T = 2 + 2\gamma + 2\gamma^2 + 2\gamma^3 + 2\gamma^4 + 4\gamma^5$$
$$= 2 + 2 * 0.9 + 2 * (0.9)^2 + 2 * (0.9)^3 + 2 * (0.9)^4 + 4 * (0.9)^5 = 10.55$$

7 to 28 :

$$G_T = 2 + 2 * 0.9 + 2 * (0.9)^2 - 4 * (0.9)^5 = 2.54$$

if $rr_m = 0$:
for 13 to 24 :

$$G_T = 0 + 0 * 0.9 + 0 * (0.9)^2 + 0 * (0.9)^3 + 0 * (0.9)^4 + 4 * (0.9)^5 = 2.36$$

for 7 to 28 :

$$G_T = 0 + 0 * 0.9 + 0 * (0.9)^2 - 4 * (0.9)^5 = -2.916$$

if $r_m = -1$:
for 13 to 24 :

$$G_T = -1 + -1 * 0.9 + -1 * (0.9)^2 + -1 * (0.9)^3 + -1 * (0.9)^4 + 4 * (0.9)^5 = -2.59$$

for 7 to 28 ;

$$G_T = -1 + -1 * 0.9 + -1 * (0.9)^2 - 4 * (0.9)^5 = -5.626$$

if $r_m = -4$: for 13 to 28 :

$$G_T = --4 + -4 * 0.9 + -4 * (0.9)^2 + -4 * (0.9)^3 + -4 * (0.9)^4 + 4 * (0.9)^5 = -17.44$$

and for 7 to 28 :

$$G_T = -4 + -4 * 0.9 + -4 * (0.9)^2 - 4 * (0.9)^5 = -13.756$$

Problem 2 .

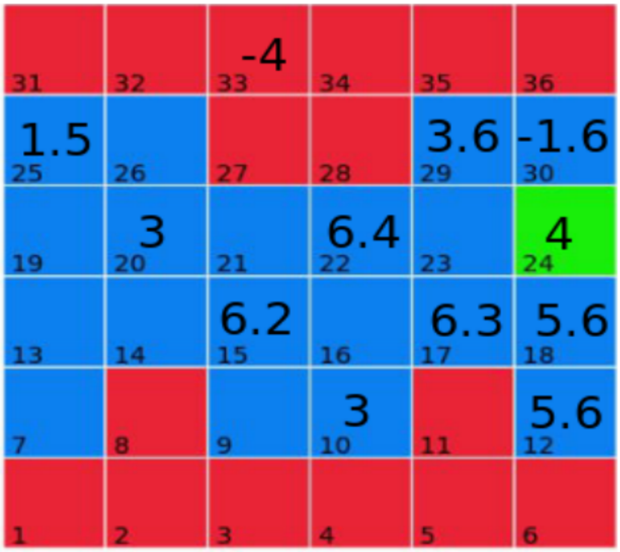
we have to compute this values by following equasion :

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}[G_t|S_t = s] forall s \in \mathcal{S} && (1) \\ &= \mathbb{E}[R_t + \gamma G_{t+1}|S_t = s] && (2) \\ &= \sum_a \pi(a|s) \sum_{s',r'} p(s', r|s, a)[r + \gamma v_\pi(s'] && (3) \end{aligned}$$

and for Q we have :

$$q_\pi(s, a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma v_\pi(s'))$$

Firstly for $r_m = 2$ we compute the values of states:
befor computing the value of state 25 , we have to compute value of $12, 20, 27, 15, 2, 10, 17, 5,$
so :
$v(s = 18) = 2 + 0.9 * v(s = 24) = 2 + 0.9 * 4 = 5.6$
$v(s = 12) = 2 + 0.9 * v(s = 18) = 2 + 0.9 * 5.6 = 7.04$
$v(s = 17) = 0.5(2 + 0.9 * v(s = 12) + 0.9 * v(s = 24)) = 6.9$
$v(s = 30) = 2 + 0.9 * v(s = 36) = 2 + 0.9 * -4 = -1.6$ $v(s = 29) = 2 + 0.5(0.9 * v(s = 36) + 0.9 * v(s = 24)) = 2$
$v(s = 22) = 2 + 0.5(0.9 * v(s = 29) + 0.9 * v(s = 17)) = 6$
$v(s = 10) = 2 + 0.5(0.9 * v(s = 17) + 0.9 * v(s = 5) = 3.3$
$v(s = 15) = 2 + 0.5(0.9 * v(s = 22) + 0.9 * v(s = 10) = 6.1$
$v(s = 20) = 2 + 0.5(0.9 * v(s = 27) + 0.9 * v(s = 15) = 3$
$v(s = 25) = 2 + 0.5(0.9 * v(s = 32) + 0.9 * v(s = 20) = 1.5$



(a) $r_m = 2$

(b) $r_m = 0$

Figure 1: The same cup of coffee. Two times.

Here we have four cases for $r_m = 2, 0, -1, -4$ .in each case first we calculate the returns from terminal states for the states that we need. The value of Green state is 4 and all of the red states are -4 .

a. r_m = 2 : in this case the table of the values like below :

b. r_m = 0 : in this case the table of the values like above in Figure2: b. for r_m = -1 and r_m = -4 you can see states value in the table below:

(a) r_m = -1 (b) r_m = -4

Figure 2: The same cup of coffee. Two times.

]

Here we are going to compute the quality of given actions.
for $r_m = 2$ we have :
$Q(23, right - up) = 2 + 0.9 * v(s = 30) = 2 + 0.9 * -4 = -1.6$
$Q(15, right - up) = 2 + 0.9 * v(s = 22) = 2 + 0.9 * 6 = 7.4$
for $r_m = 0$ we have :
$Q(23, right - up) = 0 + 0.9 * v(s = 30) = 0 + 0.9 * -3.6 = -3.24$
$Q(15, right - up) = 0 + 0.9 * v(s = 22) = 0 + 0.9 * 6 = 7.4$
for $r_m = -1$ we have :
$Q(23, right - up) = -1 + -1.9 * v(s = 30) = -1 + 0.9 * -4.6 = -5.14$
$Q(15, right - up) = -1 + 0.9 * v(s = 22) = -1 + 0.9 * -0.82 = -1.73$
for $r_m = -4$ we have :
$Q(23, right - up) = -4 + -0.9 * v(s = 30) = -4 + 0.9 * -7.6 = -10.84$
$Q(15, right - up) = -4 + 0.9 * v(s = 22) = -4 + 0.9 * -7.6 = -10.84$

Problem 3.

Firstly we discuss about optimal policy and then we study the affect of diffrent discount factor .
Assume that $r_m = 2$ .in this case the agent prefers to hang out in the environmnet as much as possible beacuse it gets a positive reward .so we can find 2 optimal policy :
1.25 $\rightarrow$ 20 $\rightarrow$ 15 $\rightarrow$ 10 $\rightarrow$ 17 $\rightarrow$ 12 $\rightarrow$ 18 $\rightarrow$ 24
and another one is :
2.25 $\rightarrow$ 20 $\rightarrow$ 15 $\rightarrow$ 22 $\rightarrow$ 17 $\rightarrow$ 12 $\rightarrow$ 18 $\rightarrow$ 24. Second if $r_m = 0$ then it is not

important for our agent whitch way will be chosen . so optimal policy in this case is any way that reatches the green state. but in reality agetn choose : $25 \rightarrow 20 \rightarrow 15 \rightarrow 22 \rightarrow 17 \rightarrow 24$ and its reason is that our agent select the states with higher value and value of states near the reds are lower so it always avoid to be in thease states .
Third if r_m = -1 : in this case agent attempts to choose the closest way to green state beacuse it gets punishment by each action . we have two ways with minimum punishment :
$25 \rightarrow 20 \rightarrow 15 \rightarrow 22 \rightarrow 17 \rightarrow 24$ and another one is :
$25 \rightarrow 20 \rightarrow 15 \rightarrow 10 \rightarrow 17 \rightarrow 24$.
However due to the calue of states that was calculated in the previous problem the way that agent chose will :
$25 \rightarrow 20 \rightarrow 15 \rightarrow 22 \rightarrow 17 \rightarrow 24$.
Finally if r_m = -4 : it is better for agent to go to the red states as soon as possible beacuse it gets the larg punishmnet for being in the environmnet and it wants to finish the episode .so optimal policy is :
$25 \rightarrow 32$ and it is unique .
Here we are talking about the influences of discount factor on policy .
The discount factor, denoted by gamma (γ), is a value between 0 and 1 that represents the relative importance of future rewards compared to immediate rewards. A higher discount factor places more importance on future rewards, while a lower discount factor places more importance on immediate rewards. In the context of an MDP, the discount factor impacts the calculation of the expected total reward for each possible action in each state. The optimal policy is derived by selecting the action that maximizes the expected total reward in each state. If the discount factor is lower, the agent will tend to choose actions that yield greater immediate rewards, since future rewards are given less importance in the calculation. Conversely, if the discount factor is higher, the agent will prioritize actions that yield greater long-term rewards, even if they have lower immediate reward values. Therefore, the discount factor plays a crucial role in determining the optimal policy for a given MDP.
if γ= 0 then agent just pay attention to its temprol reward and the policy wont be optimal .in each state the agent choose an action with maximmum reward and doesnt consider its long term rewards .
if $0 < \gamma < 1$: then if our accuracy is high enough then we will reach the optimal opilicy .if not we may face a situation that actions have teh same quality , so low accuracy can affect our policy .
and as i mentioned befor if γ is less , it leads to considered short term rewards .for example in this problem if r_m = -1 and γ = 0.001 then our policy may be changed beacuse our agent wont consider the affect of green state. if γ = 1 : in episodic tasks agent reach the optimal policy . .

Problem 4.

For $0, -1$ we wiil have the shortest path to the green state.

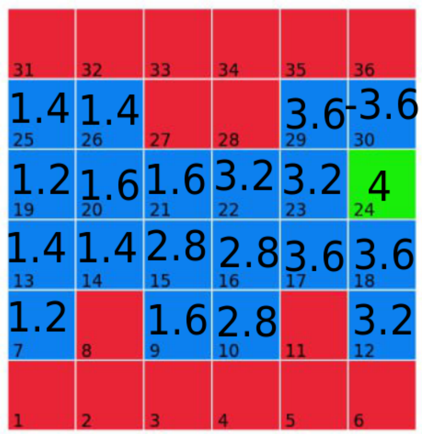
as i discussed in the past r_m = 2 leads to the longest path and r_m = -4 leads

to its closest red state (32) .
Here the equasions for optimal q and v :

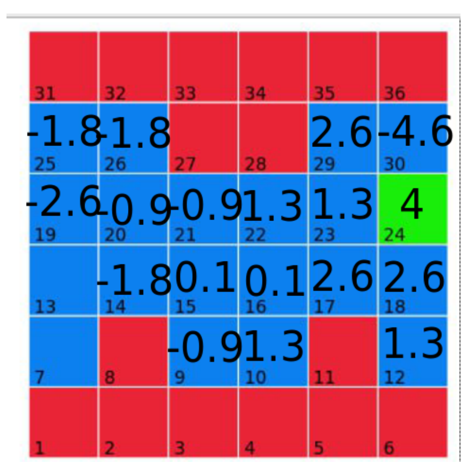
$$v_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma v_*(s'))$$

$$q_*(s, a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \max_{a' \in \mathcal{A}(s')} q_*(s', a'))$$

Now we calculate the optimal values for mentioned satets .
for $r_m = 0$:



and for r_m = -1 we have :



now we calculate the quality of the actions :
First for $r_m = 0$:
$q_*(15, up - right) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \max_{a' \in \mathcal{A}(s')} q_*(s', a')) = 0 +$

$0.9 * 3.2 = 2.8$
$q_*(23, up - right) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \max_{a' \in \mathcal{A}(s')} q_*(s', a')) = 0 + 0.9 * 3.6 = 3.2$

and for r_m = -1 we have :
$q_*(15, up - right) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \max_{a' \in \mathcal{A}(s')} q_*(s', a')) = -1 + 0.9 * 1.31 = 0.1$
$q_*(23, up - right) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma \max_{a' \in \mathcal{A}(s')} q_*(s', a')) = -1 + 0.9 * 2.6 = 1.3$

Problem 5.

you can see the equasion of policy and value itration here :

Algorithm 1: Policy Iteration

Input: MDP, small positive number θ
Output: policy $\pi \approx \pi_*$
Initialize π arbitrarily (e.g., $\pi(a|s) = \frac{1}{|\mathcal{A}(s)|}$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$)
policy-stable $\leftarrow$ $false$
repeat
  $V \leftarrow$ **Policy_Evaluation**(MDP, π, θ)
  $\pi' \leftarrow$ **Policy_Improvement**(MDP, V)
  if $\pi = \pi'$ **then**
    policy-stable $\leftarrow$ *true*
  end
  $\pi \leftarrow \pi'$
until *policy-stable* = *true*;
i **return** π

Table 1: Policy Iteration

-4	-4	-4	-4	-4	-4
-3	-3	-4	-4	1	-2.6
-3.5	-3	-3	-3.5	-3.5	4
-3	-3.5	-3.5	-3	0.55	4.6
-3	-4	-3	-3	-4	-3.5
-4	-4	-4	-4	-4	-4

Algorithm 2: Value Iteration

Input: MDP, small positive number θ
Output: policy $\pi \approx \pi_*$
Initialize V arbitrarily (e.g., $V(s) = 0$ for all $s \in \mathcal{S}^+$)
repeat
  $\Delta \leftarrow 0$
  for $s \in \mathcal{S}$ **do**
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma V(s'))$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
  end
until $\Delta < \theta$;
$\pi \leftarrow$ **Policy_Improvement**(MDP, V)
return π

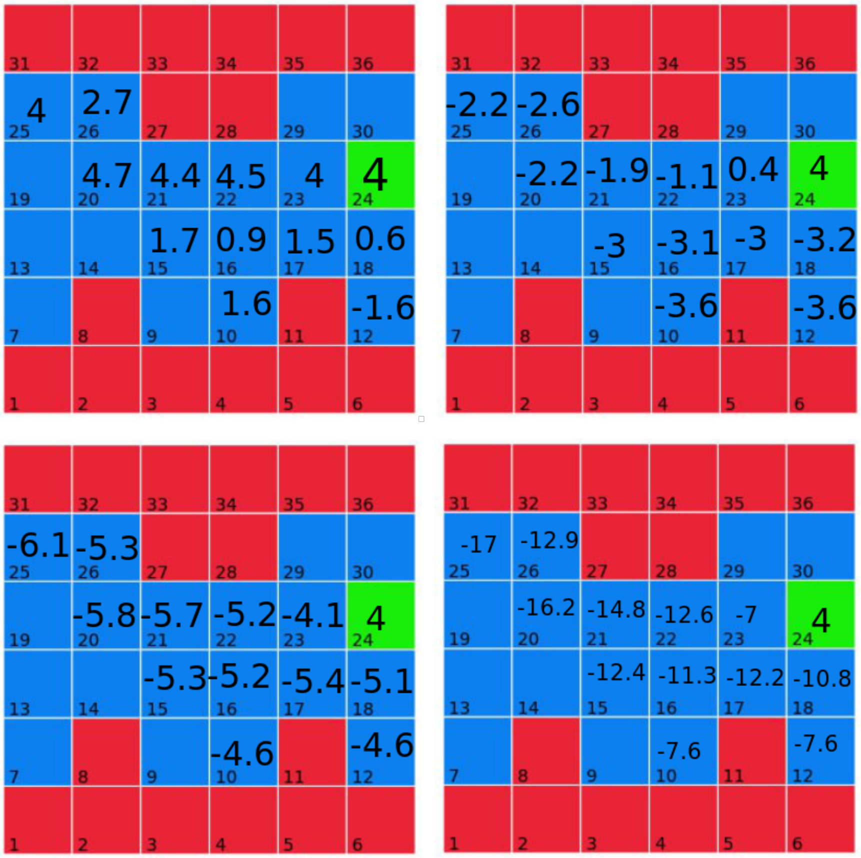
Table 2: Value Iteration

-4	-4	-4	-4	-4	-4
-2.6	-2.6	-4	-4	4.6	-2.6
-3.5	-2.6	-2.6	-3.5	-3.5	4
-2.6	-3.5	-3.5	-2.6	4.6	4.6
-2.6	-4	-2.6	-2.6	-4	-3.5
-4	-4	-4	-4	-4	-4

The policy in both value and policy iteration is choosing an action with higher value.
if we put $\gamma = 0$ it has an impact on the policy as its attention just to the temporal reward and due to the reward of both actions are $+1$ the policy will be choosing the actions with same probability .so adjust the γ has influence the policy .

Problem 6.

you can see the value of each state below :
we use $v_\pi(s) = \sum p(s, r|s^{'}, a)[r + \gamma v_\pi(s)]$ and we start from the terminal states and propagate the value to another states.



a.up-right:$r_e = 0$

b.up-left:$r_e = 2$, c.bottom-right:$r_e = -4$, d.bottom-left :$r_e = -1$

due to the figure we can detemine the optimal policy .

for $r_e = 0$: $25 \to 20 \to 21 \to 22 \to 23 \to 24$ and its unique and its also choose the closest way to green state.

for $r_e = +2$: $25 \to 27 \to 21 \to 22 \to 23 \to 24$ an its a unique policy , agent with this reward and MDP prefer to hang out in the environmnet and finally reach the green state but as its policy it has just this way , however we have to attention that this policy : $25 \to 20 \to 15 \to 16 \to 17 \to 18 \to 12 \to 6$ can be an optimal policy and this way gets reward as much as it gets in the previous policy .

for r_e = -1 : $25 \to 26 \to 27$ in this case the agent prefer to reach the cosest resd state hence it wants to end the episode as soon as possible beacise it gets a negetive reward for each action .and it is unique .

for $r_e = -4$: the optimal policy is $25 \to 26 \to 27$ it is unique.

γ has impact on all of the moods above. For example if you put $\gamma = 0$ policy will be random . and policy will coose actions with same probability .

and if $\gamma = 1$ in episodic task , the policy will be optimal. overally A higher discount factor places more importance on future rewards, while a lower discount factor places more importance on immediate rewards. In the context of an MDP, the discount factor impacts the calculation of the expected total reward for each possible action in each state. The optimal policy is derived by selecting the action that maximizes the expected total reward in each state. If the discount factor is lower, the agent will tend to choose actions that yield greater immediate rewards, since future rewards are given less importance in the calculation. Conversely, if the discount factor is higher, the agent will prioritize actions that yield greater long-term rewards, even if they have lower immediate reward values. Therefore, the discount factor plays a crucial role in determining the optimal policy for a given MDP.

as i mentioned above $r_e = +2 and 0$ return the closest way to green state.

Problem 7.

In the first MDP with $r_m = 0$ the maximum reward the the agent gets is $+4$. in second MDP if $r_e < 0$, then the total reward will be less than $+4$. so r_e has to greater than 0 .on the other hand if it becomes so large , than the policy prefer to choose the actions to hang out in the environmnet as much as possible and finally reaching the red state (it starts from 20 to 6). if we want that agent reaching the green state , the value of 18 has to less than 24 so we have the equasions below :

$$v_\pi(s_{18}) = r_e + \gamma(v_\pi(s_{12})) = r_e + \gamma(r_e + (\gamma * -4)) = 1.9 * r_e + 0.81 * (-4) = 1.9 * r_e - 3.24)$$

$$1.9 * r_e - 3.24 < 4 \Rightarrow 1.9 * r_e < 7.24 \Rightarrow r_e < 3.81$$

so r_e have to be in range of $[0, 3.81]$.

Problem 8.

a.The states that can reach the green state by non-productive actions are : $25, 19, 13, 7, 26, 20, 14, 21, 15, 9, 22, 16, 10, 29, 23, 17, 18, 12$ so all of the blue states except 30 can reach the goal state ,and $25, 26, 29, 30, 19, 20, 21, 22, 23$ from the productive MDP cdn reach the goal and as we see state 30 is in the productive MDP but it is not in non-productive set and other states are exist in both sets .

Problem 9.

yes it can be changed , consider an MDP whitch its rewardds are negetive , in this MDP agent prefer to complete the sequense as soon as possible and if we add C in order to convert the rewards of the last MDP into positives then agent prefer to hange out in the environmnet as much as possible and optimal policy well be changed .
we have the equasion below :

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}[G_t|S_t = s] for all s \in \mathcal{S} && (1) \\ &= \mathbb{E}[R_t + \gamma G_{t+1}|S_t = s] && (2) \\ &= \sum_a \pi(a|s) \sum_{s^,r^,} p(s^,, r|s, a)[r + \gamma v_\pi(s^,] && (3) \end{aligned}$$

and if we add a constant value C to all of the rewards then we have :

$v_\pi(s) = \sum_a \pi(a|s) \sum_{s^,r^,} p(s^,, r|s, a)[r + C\gamma v_\pi(s^,]$

$= v_\pi(s) = \sum_a \pi(a|s) \sum_{s^,r^,} p(s^,, r|s, a)[r + C + \gamma v_\pi(s^,]$
$= v_\pi(s) = \sum_a \pi(a|s) \sum_{s^,r^,} p(s^,, r|s, a)[r+\gamma v_\pi(s^,]+\sum_a \pi(a|s) \sum_{s^,r^,} p(s^,, r|s, a)[C+ \gamma v_\pi(s^,]$
ant $\sum_a \pi(a|s) \sum_{s^,r^,} p(s^,, r|s, a)[C]$
in our problem with productive MDP , as mentioned in the past the optimal policy with $r_e = -4$ try to reach the closest red state but if we asume that $C = 10$ agent prefer to hang out in the environment and policy will be changed.$r_e = 6, r_r = 4, r_a = 14, C = 10$

Problem 10.

Thw strategy that i choose is :
1. One of the important point is that all of the rewards have to be negetive , hence if thw application wants to find the closet way to distanation it has to get a negetive reward for hanging out acrooss the environmnet .so i assign a

negetive reward to each action .
2 .another thing that have to be considered is traffic .i divide the situations into 3 parts , green , orange and red . thease coulors illustrate the intensity of traffic . when i want to assign the rewards , at first i find out the colour of each satate and assign -1 , -2 and -3 to green , orange and red respectively .
if i follow thease rules i will find a way with minimum distance and lower traffic .