Q1

Follow the room example, perform Q learning using the deterministic model-free equation to update the Q function for the room configuration below. Use the same reward and Q function for the room configuration below. Use the same reward and Q function initialization, Note the goal is to find the optimal policy that produces the shorest path to outside from each room.

- Identify the states, provide a generic reward function, and initial Q function
- Follow the value-iteration pseudo code in the lectures note to enumerate each state and action, show the Q function after each iteration, and produce the final learnt policy pi with respect to each state, ie $a = \pi(s)$

Identify the state:

- For the state: $s0 = 0$ room, $s1 = 1$ room .....
- For the action: $a0 =$ going to 0 room, $a1 =$ goting to the 1 room ....

Reward function:

$$R = \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 100 & -1 & -1 & -1 \\ s1 & 100 & -1 & 0 & 0 \\ s2 & -1 & 0 & -1 & 0 \\ s3 & -1 & 0 & 0 & -1 \end{bmatrix}$$

Initial Q function:

$$Q^0 = \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 0 & 0 & 0 & 0 \\ s1 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Value iteration:

Policy: $\pi(s) = \max_a Q(s, a)$

GAME 1, start at s_1

- Iteration 1: $a0 = \pi(s_1)$, $a0 = \pi(s_0)$ (sample from uniform distribution, due to all value are the same)

$$
\begin{aligned}
Q(s_1, \pi(s_1) =& R(s_1, \pi(s_1)) + \gamma * Q(s_0, \pi(s_0)) \\
Q(s_1, a_0) =& 100 + .8 * 0 \\
Q^1 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 0 & 0 & 0 & 0 \\ s1 & 100 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 2: $a0 = \pi(s_1)$, $a0 = \pi(s_0)$ (sample from uniform distribution, due to all value are the same)

$$
\begin{aligned}
Q(s_0, \pi(s_0) =& R(s_0, \pi(s_0)) + \gamma * Q(s_0, \pi(s_0)) \\
Q(s_1, a_0) =& 100 + .8 * 100 \\
Q^2 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 180 & 0 & 0 & 0 \\ s1 & 100 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

GAME 2, start at s_2

- Iteration 3: $a3 = \pi(s_2)$ (sample from uniform distribution, due to all value are the same)

$$
\begin{aligned}
Q(s_2, \pi(s_2) =& R(s_2, \pi(s_2)) + \gamma * Q(s_3, \pi(s_3)) \\
Q(s_2, a_3) =& 0 + .8 * 0 \\
Q^3 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 180 & 0 & 0 & 0 \\ s1 & 100 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 4: $a1 = \pi(s_3)$ (sample from uniform distribution, due to all value are the same)

$$
\begin{aligned}
Q(s_3, \pi(s_3)) =& R(s_3, \pi(s_3)) + \gamma * Q(s_1, \pi(s_1)) \\
Q(s_3, a_1) =& 0 + .8 * Q(s_1, a_0) \\
=& 0 + .8 * 100 \\
Q^4 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 180 & 0 & 0 & 0 \\ s1 & 100 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 80 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 5:

$$
\begin{aligned}
Q(s_1, \pi(s_1)) =& R(s_1, \pi(s_1)) + \gamma * Q(s_0, \pi(s_1)) \\
Q(s_1, a_0) =& 100 + .8 * 180 \\
Q^5 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 180 & 0 & 0 & 0 \\ s1 & 244 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 80 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 6:

$$
\begin{aligned}
Q(s_0, \pi(s_0)) =& R(s_0, \pi(s_0)) + \gamma * Q(s_0, \pi(s_0)) \\
Q(s_1, a_0) =& 100 + .8 * 180 \\
Q^6 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 244 & 0 & 0 & 0 \\ s1 & 244 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 80 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

GAME 3, start at s_2

- Iteration 7: $a3 = \pi(s_2)$ (sample from uniform distribution, due to all value are the same)

$$
\begin{aligned}
Q(s_2, \pi(s_2) =& R(s_2, \pi(s_2)) + \gamma * Q(s_3, \pi(s_3)) \\
Q(s_2, a_3) =& 0 + .8 * 80 \\
Q^7 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 244 & 0 & 0 & 0 \\ s1 & 244 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 64 \\ s3 & 0 & 80 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 8:

$$
\begin{aligned}
Q(s_3, \pi(s_3) =& R(s_3, \pi(s_3)) + \gamma * Q(s_1, \pi(s_1)) \\
Q(s_3, a_1) =& 0 + .8 * Q(s_1, a_0) \\
=& 0 + .8 * 244 \\
Q^8 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 244 & 0 & 0 & 0 \\ s1 & 244 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 64 \\ s3 & 0 & 195 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 9:

$$
\begin{aligned}
Q(s_1, \pi(s_1) =& R(s_1, \pi(s_1)) + \gamma * Q(s_0, \pi(s_0)) \\
Q(s_3, a_1) =& 100 + .8 * Q(s_0, a_0) \\
=& 100 + .8 * 244 \\
Q^9 =& \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 244 & 0 & 0 & 0 \\ s1 & 295 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 64 \\ s3 & 0 & 195 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

- Iteration 10:

$$
\begin{aligned}
Q(s_0, \pi(s_0)) =& R(s_0, \pi(s_0)) + \gamma * Q(s_0, \pi(s_0)) \\
Q(s_0, a_0) =& 100 + .8 * Q(s_0, a_0) \\
=& 100 + .8 * 244
\end{aligned}
$$

$$
Q^{10} = \begin{bmatrix}
S & a0 & a1 & a2 & a3 \\
s0 & 295 & 0 & 0 & 0 \\
s1 & 295 & 0 & 0 & 0 \\
s2 & 0 & 0 & 0 & 64 \\
s3 & 0 & 195 & 0 & 0
\end{bmatrix}
$$

I would like to stop here. Basically the policy has been converged, but the value function is not, the optimal value function would be:

$$
Q^* = \begin{bmatrix}
S & a0 & a1 & a2 & a3 \\
s0 & 500 & 0 & 0 & 0 \\
s1 & 500 & 0 & 0 & 0 \\
s2 & 0 & 0 & 0 & 320 \\
s3 & 0 & 400 & 0 & 0
\end{bmatrix}
$$

I get this equation because $Q(s_0, \pi(s_0)) = 100 + .8 * Q(s_0, \pi(s_0))$. From this example, we can see the value of the discount factor $\gamma = .8$. Because without shi discount factor, the $Q(s_0, \pi(s_0)$ will keep increasing and never converge.

Now we can talk about the policy:

Unfortunately, we may find that this Q^10 value function is already converge with $a_0 = \pi(s_1)$, $a_1 = \pi(s_3)$, $a_3 = \pi(s_2)$. The problem is, as we know, the optimal convergence should be $a_0 = \pi(s_1)$, $a_1 = \pi(s_3)$, $a_1 = \pi(s_2)$. If iteration 7 sample a_1

$$
\begin{aligned}
Q(s_2, \pi(s_2)) =& R(s_2, \pi(s_2)) + \gamma * Q(s_1, \pi(s_1)) \\
Q(s_2, a_3) =& 0 + .8 * 244
\end{aligned}
$$

$$
Q^{7'} = \begin{bmatrix}
S & a0 & a1 & a2 & a3 \\
s0 & 244 & 0 & 0 & 0 \\
s1 & 244 & 0 & 0 & 0 \\
s2 & 0 & 195 & 0 & 0 \\
s3 & 0 & 80 & 0 & 0
\end{bmatrix}
$$

The world will be totally different, if we can do this sample. This is the problem of using the deterministic policy all the time, it may lose the chance to get to the optimal policy. And this should be solved by implementing a slightly stochastic policy.

Ans to Q1

Learnt policy (not optimal due to sampling):

- At room 0: go to the room 0
- At room 1: go to the room 0
- At room 2: go to the room 3
- At room 3: go to the room 1

Q2 Policy iteration

Initial policy weight

$p(a|s,w) = \text{softmax}(W[s])\$$

$$W^0 = \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 0 & 0 & 0 & 0 \\ s1 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

GAME 1, start at $s_1 \to s_0$

Notation: $p(\tau_{i,t}|w)$, the first subscript is the game_i, the second is the timestep

1. $s_1 \to s_0$

- $\log p(\tau_{1,1}|w) = \log p(a_0|s_1,w) + \log p(a_0|s_0,w)$
- $R(\tau_1) = r_{1,1} = 100$

1. $s_0 \to s_0$

- $\log p(\tau_{1,2}|w) = \log p(a_0|s_1,w) + \log p(a_0|s_0,w)$
- $R(\tau_1) = r_{1,2} = 200$

GAME 2, start at $s_2 \to s_3 \to s_1 \to s_0$

1. $s_2 \to s_3$

- $\log p(\tau_{2,1}|w) = p(a_3|s_2, w)$
- $R(\tau_{2,1}) = r_{2,1} = 0$

1. $s_3 \to s_1$

- $\log p(\tau_{2,2}|w) = p(a_3|s_2, w) + p(a_1|s_3, w)$
- $R(\tau_2) = \sum_{i=1}^{2} r_{2,i} = 0$

1. $s_1 \to s_0$

- $\log p(\tau_{2,3}|w) = p(a_3|s_2, w) + p(a_1|s_3, w) + p(a_0|s_1, w)$
- $R(\tau_2) = \sum_{i=1}^{3} r_{2,i} = 0 + 0 + 100 = 100$

1. $s_0 \to s_0$

- $\log p(\tau_{2,3}|w) = p(a_3|s_2, w) + p(a_1|s_3, w) + p(a_0|s_1, w) + \log p(a_0|s_0, w)$
- $R(\tau_2) = \sum_{i=1}^{4} r_{2,i} = 0 + 0 + 100 + 100 = 200$

GAME 3, start at $s_2 \to s_3 \to s_1 \to s_0$

1. $s_2 \to s_3$

- $\log p(\tau_{3,1}|w) = p(a_3|s_2, w)$
- $R(\tau_{3,1}) = r_{3,1} = 0$

1. $s_3 \to s_1$

- $\log p(\tau_{3,2}|w) = p(a_3|s_2, w) + p(a_1|s_3, w)$
- $R(\tau_3) = \sum_{i=1}^{2} r_{3,i} = 0$

1. $s_1 \to s_0$

- $\log p(\tau_{3,3}|w) = p(a_3|s_2, w) + p(a_1|s_3, w) + p(a_0|s_1, w)$
- $R(\tau_3) = \sum_{i=1}^{3} r_{3,i} = 0 + 0 + 100 = 100$

1. $s_0 \to s_0$

- $\log p(\tau_{3,3}|w) = p(a_3|s_2, w) + p(a_1|s_3, w) + p(a_0|s_1, w) + \log p(a_0|s_0, w)$
- $R(\tau_3) = \sum_{i=1}^{4} r_{3,i} = 0 + 0 + 100 + 100 = 200$

According to these 3 trajectories:

$$R^0(w) = \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 0 & 0 & 0 & 0 \\ s1 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R(w) = \sum_i^3 \log p(\tau_i|w) * R(\tau_i)$$

$$= \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & \log(a_0|s_0,w)*(200) & 0 & 0 & 0 \\ s1 & \log(a_0|s_1,w)*(100+200) & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} S & a0 & a1 & a2 & \\ s0 & \log(a_0|s_0,w)*(200) & 0 & 0 & \\ s1 & \log(a_0|s_1,w)*(100+200) & 0 & 0 & \\ s2 & 0 & 0 & 0 & \log(a_3|s_2,w)*(\\ s3 & 0 & \log(a_1|s_3,w)*(0+100+200) & 0 & \end{bmatrix}$$

$$+ \begin{bmatrix} S & a0 & a1 & a2 & \\ s0 & \log(a_0|s_0,w)*(200) & 0 & 0 & \\ s1 & \log(a_0|s_1,w)*(100+200) & 0 & 0 & \\ s2 & 0 & 0 & 0 & \log(a_3|s_2,w)*(\\ s3 & 0 & \log(a_1|s_3,w)*(0+100+200) & 0 & \end{bmatrix}$$

$$= \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & 3*\log(a_0|s_0,w)*200 & 0 & 0 & 0 \\ s1 & 3*\log(a_0|s_1,w)*300 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 2*\log(a_3|s_2,w)*300 \\ s3 & 0 & 2*\log(a_1|s_3,w)*300 & 0 & 0 \end{bmatrix}$$

Determinic policy

To maximize this: $w^* = \arg\max_w R(w) = \begin{bmatrix} S & a0 & a1 & a2 & a3 \\ s0 & \infty & 0 & 0 & 0 \\ s1 & \infty & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & \infty \\ s3 & 0 & \infty & 0 & 0 \end{bmatrix}$

So the policy is:

- At room 0: go to the room 0
- At room 1: go to the room 0
- At room 2: go to the room 3
- At room 3: go to the room 1

So I can get the same policy with Q1. But note that, there I control the sampling, ensuring the trajectories are exactly the same with Q1. If we make another sampling, where at state 2, the agent choose to go to room1, the policy we got from Q2 might be different from Q1.