- Suppose someone gives you a randomized policy π in a continuing reinforcement learning programming problem and claims it is optimal. How would you construct a deterministic policy π which is optimal?
  Justify your answer.
- Is it always true that $J_\pi(s) \leq \max_a Q_\pi(s, a)$?
- Suppose all the rewards in a continuing reinforcement learning problem are in [0,1] and we have $\gamma = 0.9$.
  As in the lecture notes, we use J^* to denote the rewards-to-go under the optimal policy.
  First, give an upper bound on $||J^*||_\infty$.
  Second, how many iterations does it take until we can guarantee that value iteration produces J_t with $||J_t - J^*||_\infty \leq 0.01$ starting from $J_0 = \mathbf{0}$?
- Look at Eq. (*) in the lecture notes on Q-learning. It was derived under the assumption that the policy π is deterministic. What should the corresponding equation be when the policy is randomized?

1) I would construct a deterministic policy using temporal difference learning on the random policy. This is the best option since it is a continuous RL problem and the policy can step through the system and update the rewards-to-go until it reaches a limit. When it reaches a limit, the deterministic policy will be the J_s with the largest expected rewards.

2) Prove $J_\pi(s) \leq \max_a Q_\pi(s,a)$:

if $Q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) J_\pi(s')$

and $\max_a Q_\pi(s,a)$ chooses the best first action

then $J_\pi(s) \leq \max_a Q_\pi(s,a)$

because Q_π is determined by adding the reward of a specific action at a state with the rewards collected by following the policy. If $J_\pi(s)$ is the rewards collected by just following the policy, since $\max_a Q_\pi(s,a)$ chooses the optimal action, as long as the policy chooses the best action at state s, $J_\pi(s)$ will be equal to $\max_a Q_\pi(s,a)$. If the policy chooses a nonoptimal path initially, however, it will result in a total reward less than $\max_a Q_\pi(s,a)$ since they will follow the same policy after the determined best action.

3) a) $J = r + \gamma r + \gamma^2 r + \cdots$

$\|J^*\|_\infty = 1 + 0.9(1) + 0.9^2(1) + \cdots$

$= \frac{r}{1-\gamma} = \frac{1}{1-0.9} = \frac{1}{0.1}$

The optimal policy is to continually take the path with reward $=1$

$\|J^*\|_\infty = 10$

$$||J_t - J^*||_\infty \leq \gamma^t ||J_0 - J^*||_\infty .$$

b) $\|J_t - J^*\|_\infty \leq \gamma^t \|J_0 - J^*\|_\infty$

$\|J_t - J^*\|_\infty \leq 0.01 \leq \gamma^t(\|J_0\|_\infty - \|J^*\|_\infty)$

$0.01 \leq 0.9^t(\|0 - 10\|_\infty)$

$0.01 \leq 0.9^t(10)$

$0.001 \leq 0.9^t$

$\log(0.001) \leq \log(0.9) \cdot t$

$\frac{\log(0.001)}{\log(0.9)} \leq t$

$65.56 \leq t$

$66 \leq t$

4) $Q_{\pi,t+1}(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|s,a) Q_{\pi,t}(s', \pi(s'))$

Randomized:

$Q_{\pi,t+1}(s,a) = \sum_a \pi(a|s) r(s,a) + \gamma \sum_{s'} \left(P(s'|s,a) \sum_{a'} Q_{\pi,t}(s',a') \pi(a'|s') \right)$

- Consider an MDP with two states, A and B. In A, there are two actions you can take. Action 1 keeps you in state A, with a reward of one. Action 2 moves you to B, with a reward of zero. In state B, there is only one action to take, which keeps you in B with a reward of 2.

  You want to use temporal difference learning to evaluate the rewards-to-go of the "choose at random" policy.

  You generate the following two sample paths:

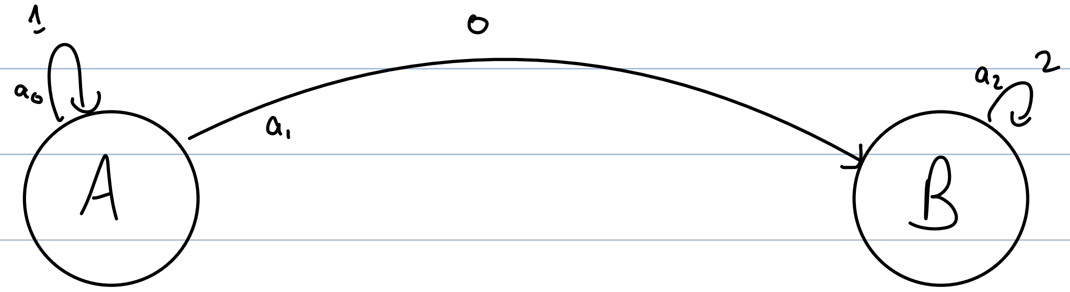
  $s_0 = A, a_0 = 1, s_1 = A, a_1 = 2, s_2 = B$

  $s_0 = A, a_0 = 2, s_1 = B, a_1 = 1, s_2 = B, a_2 = 1, s_3 = B$

  Use temporal difference learning to come up with estimates of the rewards-to-go from both states starting from [16,16].

$\gamma = \frac{1}{2}$

$\alpha_t = \frac{1}{t}$



a) $S_1 = A \quad a_1 = 1$

$S_2 = A \quad a_2 = 2$

$S_3 = B$

$$J_0 = \begin{bmatrix} 16 \\ 16 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} 16 + \frac{1}{1}\left(1 + \frac{1}{2}(16) - 16\right) \\ 16 \end{bmatrix} = \begin{bmatrix} 16 + (1 + 8 - 16) \\ 16 \end{bmatrix}$$

$$= \begin{bmatrix} 16 + (-7) \\ 16 \end{bmatrix} = \begin{bmatrix} 9 \\ 16 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} 9 + \frac{1}{2}\left(0 + \frac{1}{2}(16) - 9\right) \\ 16 \end{bmatrix} = \begin{bmatrix} 9 + \frac{1}{2}(8 - 9) \\ 16 \end{bmatrix} = \begin{bmatrix} 8.5 \\ 16 \end{bmatrix}$$

b) $S_1 = A \quad a_1 = 2$

$S_2 = B \quad a_2 = 1$

$S_3 = B \quad a_3 = 1$

$S_4 = B$

$$J_1 = \begin{bmatrix} 16 \\ 16 \end{bmatrix}$$

$$J_2 = \begin{bmatrix} 16 + \frac{1}{1}\left(0 + \frac{1}{2}(16) - 16\right) \\ 16 \end{bmatrix} = \begin{bmatrix} 16 + (8 - 16) \\ 16 \end{bmatrix} = \begin{bmatrix} 8 \\ 16 \end{bmatrix}$$

$$J_3 = \begin{bmatrix} 8 \\ 16 + \frac{1}{2}\left(2 + \frac{1}{2}(16) - 16\right) \end{bmatrix} = \begin{bmatrix} 8 \\ 16 + \frac{1}{2}(2 + 8 - 16) \end{bmatrix}$$

$$= \begin{bmatrix} 8 \\ 16 + \frac{1}{2}(-6) \end{bmatrix} = \begin{bmatrix} 8 \\ 13 \end{bmatrix}$$

$$J_4 = \begin{bmatrix} 8 \\ 13 + \frac{1}{3}\left(2 + \frac{1}{2}(13) - 13\right) \end{bmatrix} = \begin{bmatrix} 8 \\ 13 + \frac{1}{3}\left(2 + \frac{13}{2} - 13\right) \end{bmatrix}$$

$$= \begin{bmatrix} 8 \\ 13 + \frac{1}{3}\left(-\frac{9}{2}\right) \end{bmatrix}$$

$$= \begin{bmatrix} 8 \\ 13 - \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 \\ \frac{23}{2} \end{bmatrix}$$

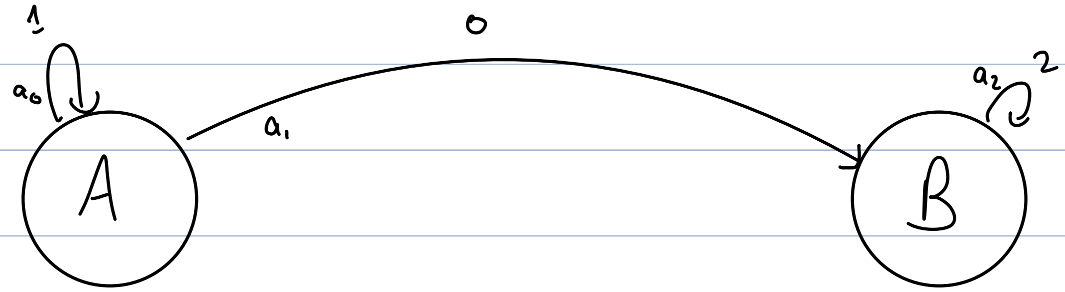
- Consider an MDP with two states, A and B. In A, there are two actions you can take. Action 1 keeps you in state A, with a reward of one. Action 2 moves you to B, with a reward of zero. In state B, there is only one action to take, which keeps you in B with a reward of 2.

  (i) Suppose you start from [16,16,16]. Starting at state A, write out the first three iterations of Q-learning with $\epsilon = 0$ starting from $s = A, a = 1$, $S_2 = A, a_2 = 2, S_3 = B, a_3 = 1$

  (ii) Write code to implement Q-learning to find the optimal Q-values in this example. Use $\alpha = \frac{1}{\sqrt{t}}$ 224

$\gamma = \frac{1}{2}$

$\alpha_t = \frac{1}{t}$



1st iteration:

$$Q_2(A,1) = 16 + \frac{1}{1}\left(1 + \frac{1}{2}\max\left(Q_1(A,1), Q_1(A,2)\right) - 16\right)$$
$$= 16 + \frac{1}{1}\left(1 + \frac{1}{2}(16) - 16\right)$$
$$= 16 + \frac{1}{1}(1 + 8 - 16) = 16 - 7 = 9$$

$$Q_2 = \begin{bmatrix} 9 \\ 16 \\ 16 \end{bmatrix}$$

2nd iteration:

$$Q_3(A,2) = 16 + \frac{1}{2}\left(0 + \frac{1}{2}\max\left(Q_2(B,1)\right) - 16\right)$$
$$= 16 + \frac{1}{2}\left(\frac{1}{2}(16) - 16\right)$$
$$= 16 + \frac{1}{2}(8 - 16) = 12$$

$$Q_3 = \begin{bmatrix} 9 \\ 12 \\ 16 \end{bmatrix}$$

3rd

iteration: $Q_4(B,1) = 16 + \frac{1}{3}\left(2 + \frac{1}{2} \max\left(Q_3(B,1)\right) - 16\right)$

$= 16 + \frac{1}{3}\left(2 + \frac{1}{2}(16) - 16\right)$

$= 16 + \frac{1}{3}(-6)$

$= 16 - 2 = 14$

$$Q_4 = \begin{bmatrix} 9 \\ 12 \\ 14 \end{bmatrix}$$