Aditya Kelvianto Sidharta
aks2266

EECS 6892

Homework 2

1.) Q-learning Algorithm

- Initialize $Q(s,a) = 0 \quad \forall s, a$
- $s_t = s_0$
- loop
  - $a_t \sim \pi_b(s_t)$
  - Observe (r_t, s_{t+1})
  - $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right)$
  - $\pi_b(s_t) = \text{argmax}_a Q(s_t, a)$
  - $t = t + 1$
- end loop

1.1)

$S = \{s_1, s_2, s_3\}$

$A = \{a_1, a_2, a_3\}$

$\gamma = 0.5$

$P(s_i \mid a_i, s) = 1, \quad \forall s = (s_1, s_2, s_3)$

$\forall i = (1, 2, 3)$

$R(s_1) = 1, \quad R(s_2) = 2, \quad R(s_3 = 3)$

→ Initialize $Q(s_i, a_j) = 0 \quad \forall i, j = (1, 2, 3)$

→ Random action prioritize non-greedy. smallest i if tie

→ Assuming that we start in s_1, $s^{k_0} = s_1$

$k = 0$

$$Q(s_1, a_1) = Q(s_1, a_2) : Q(s_1, a_3) = 0.$$

Thus, it is a tie, and thus we will choose a_1 as part of random process

→ As $P(s_1 | a_1, s_1) = 1$, and $R(s_1) = 1$,
we will observe $s^{k_1} = s_1$, $r^{k_0} = 1$

→ Update $Q(s_1, a_1)$

$$Q(s_1, a_1) = 0 + 0.5(1 + 0.5 \times 0 - 0)$$
$$= 0.5$$

→ thus, $\pi_b(s_1) = a_1$

$k = 1$

$Q(s_1, a_1) = 0.5$

$Q(s_1, a_2) = Q(s_1, a_3) = 0.$

Thus, a_2 will only be chosen with probability ε (random).

$P(s_2 \mid s_1, a_2) = 1$, and $R(s_2) = 1$

And thus, we observe $S^{t_2} = s_2$, $r^{k_1} = 2$

Update $Q(s_1, a_2)$ (max of Q in s_2)

$Q(s_1, a_2) = 0 + 0.5(2 + 0 - 0)$

$= 1$

In conclusion, $(a_1, 1, s_1)$ is observed because of greedy (tie), and $(a_2, 2, s_2)$ is chosen from random action with prob. ε.

1.2) → Off policy learning

→ All $Q(s_i, a_j) = 6 \quad \forall i, j = (1, 2, 3)$

Assume that we observe the following trajectory

$(s_1, a_1, 1, s_1, a_2, 2, s_2)$

$s^{t_0}, a^{t_0}, r^{t_0}, s^{t_1}, a^{t_1}, r^{t_1}, s^{t_2}$

There will be 2 updates on the Q value, once on $Q(s_1, a_1)$ and $Q(s_1, a_2)$

$$Q(s_1, a_1) = 6 + 0.5 \cdot (1 + 0.5 \times 6 - 6)$$
$$= 6 - 1 = 5$$

$$Q(s_1, a_2) = 6 + 0.5 (2 + 0.5 \times 6 - 6)$$
$$= 6 - 0.5 = 5.5$$

2.) - Q values for a finite horizon MDP

- $\gamma = 1$
- $\pi^*_{H-t}(s)$ is the action taken by an optimal policy (s) at time t.
- $Q^*_{H-t}(s,a)$ is the Q value of an optimal policy by taking action a in state s at time t.
- $V^*_{H-t}(s,a)$ is the Value function of an optimal policy in state s

2.1) Assume that we choose suboptimal action for first action, but optimal policy onwards, the value of policy π compared to the optimal one is

$$Q^*_H(s,a) - Q^{\pi}_H(s,a) = V^*_H(s) - r_H(s,a) + \sum_{s' \in S} p_H(s'|a,s) V^*_{H-1}(s')$$

2.2) Assuming that the policy is dererministic,

$$V_H^*(s_0) = Q_H^*(s_0, \pi_H^*(s_0)) + V_{H-1}^*(s_1)$$

$$= Q_H^*(s_0, \pi_H^*(s_0)) + Q_{H-1}^*(s_1, \pi_{H-1}^*(s_1)) + V_{H-2}^*(s_2)$$

$$\vdots$$

$$= Q_H^*(s_0, \pi_H^*(s_0)) + \vdots$$

$$Q_1^*(s_{H-1}, \pi_1^*(s_{H-1}))$$

$$= \sum_{t=0}^{H-1} \left(Q_{H-t}^*(s_t, \pi_{H-t}^*(s_t)) \right)$$

Similarly, as we know that $Q_H^*(s,a)$ is the true function value for state s, action a in timestep H, then the we can calculate $V_H^\pi(s_0)$ in a similar manner

$$V_H^{\pi}(s_0) = E_{\pi}\left[Q_H^{*}(s_0, \pi_H(s_0)) + V_{H-1}^{\pi}(s_1) \right]$$

$$= E_{\pi}\left[Q_H^{*}(s_0, \pi_H(s_0)) + Q_{H-1}^{*}(s_1, \pi_{H-1}(s_1)) + V_{H-2}^{\pi}(s_2) \right]$$

$$\vdots$$

$$= E_{\pi}\left[Q_H^{*}(s_0, \pi_H(s_0)) + \right.$$

$$\vdots$$

$$\left. Q_1^{*}(s_{H-1}, \pi_1(s_{H-1})) \right]$$

$$= E_{\pi}\left[\sum_{t=0}^{H-1} \left(Q_{H-t}^{*}(s_t, \pi_{H-t}(s_t)) \right) \right]$$

Thus,

$$|V_H^{*}(s_0) - V_H^{\pi}(s_0)|$$

$$= E_{\pi}\left[\sum_{t=0}^{H-1} \left(Q_{H-t}^{*}(s_t, \pi_{H-t}^{*}(s_t)) - Q_{H-t}^{*}(s_t, \pi_{H-t}(s_t)) \right) \right.$$

3) Assuming that we have $\hat{Q}$, such that

$$|\hat{Q}_{H-t}(s,a) - Q^*_{H-t}(s,a)| \leq \varepsilon,$$

we follow the policy $\tilde{\pi}_{H-t}(s) = \underset{a}{\text{argmax}}\, \hat{Q}_{H-t}(s,a)$

$$|V^*_H(s_0) - V^{\hat{\pi}}_H(s_0)|$$

$$= E_\pi \left[\sum_{t=0}^{H-1} Q^*_{H-t}(s_t, \pi^*_{H-t}(s_t)) - Q^*_{H-t}(s_t, \tilde{\pi}_{H-t}(s_t)) \right]$$

because $\tilde{\pi}_{H-t}(s) \in \underset{a}{\text{argmax}}\, \hat{Q}_{H-t}(s,a)$,

we know that $V^{\hat{\pi}}_{H-t}(s) = \underset{a}{\max}\, \hat{Q}_{H-t}(s,a)$

$\forall\, s,\ t = 1 \ldots H-1$

and thus, estimation of $V^*_H(s_0)$

$$= \sum_{t=0}^{H-1} \left| Q^*_{H-t}(s_t, \pi^*_{H-t}(s_t)) - \hat{Q}_{H-t}(s_t, \pi^*_{H-t}(s_t)) \right|$$

$$\leq \sum_{x=0}^{H-1} \varepsilon$$

$$= H\varepsilon$$

Similarly, estimation of $V_H^{\hat{\pi}}$ can be given as

$$E_\pi \Big[\sum_{x=0}^{H-1} Q^*_{H-t}(s_t, \underset{a}{\operatorname{argmax}}\, \hat{Q}_{H-t}(s_t, a)) \; \hat{Q}_{H-t}(s_t, \underset{a}{\operatorname{argmax}}\, \hat{Q}_{H-t}(s_t, a)) \Big]$$

$$\leq E_\pi \Big[\sum_{x=0}^{H-1} \varepsilon \Big]$$

$$= H\varepsilon$$

and then,

$$|V_H^*(s_0) - V_H^{\hat{\pi}}(s_0)| \leq 2\varepsilon H$$

3.) In the online learning, when we are still not in full exploitation phase (i.e: the ε in the ε greedy policy is not zero), then Q learning performance will be worse than SARSA. The reason is because Q learning is an "optimistic" algorithm - it updates the current Q value based on current reward and the best possible Q value for next state. And thus, even though it is correct in determining that the optimal policy is to walk near the cliff, the ε greedy policy will inadvertantly make the agent fall when it choose a random policy. SARSA takes this into account (the possibility of falling under ε-greedy policy) in the update of it's Q values, and thus choose to take a safer path, far away from the cliff edge, even though it is non optimal. However, both of these algorithm will converge to the optimal policy once the ε value converge to zero.