

SAHAS MARWAH

2020237

HW-3

1. To make the pseudocode more efficient, to calculate the mean it would be better to maintain the mean and the count for each state-action pair and update them incrementally.

Acc. to section 2.4 of the course book; given Q_n and R_n , the new avg. for all n rewards can be found by:

$$Q_{n+1} = Q_n + \frac{(R_n - Q_n)}{n}$$

Generally,

$$\text{New Est.} = \text{old est.} + \text{step size} (\text{Target} - \text{old Est.})$$

Update MC ES:

$$Q(s) \in A(s) \quad \forall s \in S$$

$$Q(s, a) \in R \quad \forall s \in S, a \in A = 0$$

$$\text{counter}(s, a) \quad \forall s \in S, a \in A = 0$$

Loop forever:

Choose $s_0 \in S, A_0 \in A(s_0)$ randomly

Generate $s_0, A_0, R_1, G, A_1, R_2, \dots, s_{T-1}, A_{T-1}, R_T$.

$$G = 0$$

Loop $t = T-1 : -1 : 0$

$$G = R_{t+1} + \gamma G$$

Until s_t, A_t appears in s_0, A_0, G, A_1 — :

~~$$Q(s_t, A_t) = Q(s_t, A_t) \times \text{counter}(s_t, A_t) + G$$~~
$$Q(s_t, A_t) = \frac{Q(s_t, A_t) \times \text{counter}(s_t, A_t) + G}{\text{counter}(s_t, A_t) + 1}$$

$$\text{counter}(s_t, A_t) = \text{counter}(s_t, A_t) + 1$$

$$nQ = \sum G_n$$

$$(n+1)Q' = \sum G_{n+1}$$

$$nQ' + Q' = \sum G_n + G_{n+1}$$

$$n(Q' - Q) = G_{n+1} - Q$$

$$Q' = \frac{nQ + G_{n+1}}{n+1}$$

classmate

Date _____

Page _____

Pseudocode is equivalent as we are calculating mean at every iteration.

$$Q(S_t, A_t) = \frac{G_1 + G_2 + \dots + G_n}{n} \quad (\text{similar to section 2.4 of the book})$$

$$Q'(S_t, A_t) = \frac{G_1 + G_2 + \dots + G_{n+1}}{n+1}$$

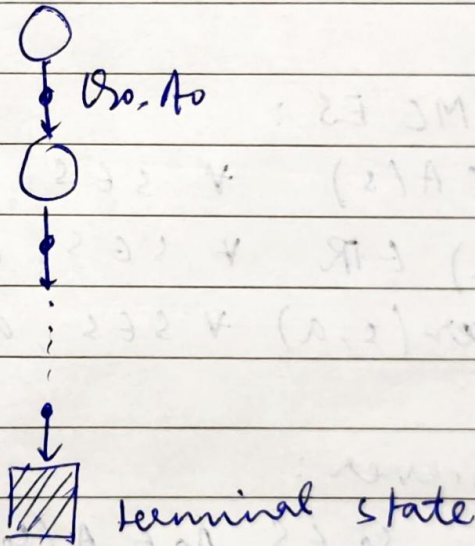
$$\text{So, } Q(S_t, A_t) = \frac{nQ(S_t, A_t) + G_{n+1}}{n+1}$$

$$= \frac{\text{Counter}(S_t, A_t) \cdot Q(S_t, A_t) + G_{n+1}}{\text{Counter}(S_t, A_t) + 1}$$

checked as pseudocode.

(2) MC est of 2π :

Backup diagram:



3- Eq. (5.6) $\Rightarrow V(s) = \sum_{t: T(t)-1} G_t$

$\sum_{t: T(t)-1}$

In terms of $Q(s, a) = ?$

considers.

$$\begin{aligned} & P(s_t, A_t, s_{t+1}, A_{t+1} \mid s_t = s, A_t = a) \\ &= P(s_t \mid A_t) \cdot P(s_{t+1} \mid s_t, A_t) \cdot P(A_{t+1} \mid s_{t+1}) \cdot P(s_{t+1} \mid s_t, A_t) \\ &= \prod_{t=1}^{T-1} (\pi(a_i \mid s_i) \cdot P(s_{i+1} \mid s_i, a_i)) \cdot P(s_{t+1} \mid s_t, a_t) \end{aligned}$$

Relative Ratio :

$$J_{t: T-1} = \frac{\pi \pi(a_i \mid s_i)}{\pi b(a_i \mid s_i)}$$

$$\frac{\pi \pi(a_i \mid s_i)}{\pi b(a_i \mid s_i)} \cdot \frac{\pi}{\pi b}$$

$$\therefore Q(s, a) = \frac{\sum_{t \in \gamma(s, a)} J_{t: T-1} G_t}{\sum_{t \in \gamma(s, a)} J_{t: T-1}}$$

5. A scenario in which TD update would be better:

From the hint let us build an example.

let X be the old parking space, Y be the new parking and H be the highway entry point.

~~OLD scenario~~

OLD scenario: $H \rightarrow \rightarrow \rightarrow X$

NEW " : $H \rightarrow \rightarrow \rightarrow Y$

As we have a lot of experience,

For MC method: $V(s_t) = V(s_t) + \alpha [G_t - V(s_t)]$

~~Here we calculate~~ Here we calculate, G_t by going on the entire episode but if we have a break in b/w we have to calculate G_t all over again. This will take a long time to converge $V \rightarrow V^*$.

On the other hand, $\gamma = 0.2$

TD Method:
$$v(s_t) = v(s_t) + \alpha [R_t + \gamma v(s_{t+1}) - v(s_t)]$$

Here, we just make prediction of our new parking space, then we can use the previous experience for rest of the holes.

Hence, $v \rightarrow v_k$ convergence is faster.

6. (6.3) In the first episode, only the value of $V(A)$ decreased. So, the episode must have ended on the left most state (terminal).

~~$TDS: \delta_t = R_{t+1} + \gamma V(s_{t+1}) - V(s_t)$~~ $V(A) = 0.5$
(graph)

~~$= 0 + 1$~~

$$V(A) = V(A) + \alpha [R_{t+1} + V(s_{t+1}) - V(A)]$$

$$= 0.5 + 0.1 [0 + 0 - 0.5]$$

$$= 0.45$$

~~Initially, from~~

So, $\Delta V(A) = 0.5 - 0.45 = \underline{\underline{0.05}}$

All other estimates: $s_t = 0 + \gamma(0.5) - 0.5$

So, $V(k) = V(k) + 0.1 (0 + \underbrace{V(s_t)}_{\text{same}} - V(s_{t-1}))$

$$= V(k) + 0$$

∴ No change.

(6.4) From the ~~plot~~ ^{graph} we infer that on increasing α , we get noisy plots. Noisy plots are also non-converging.

~~At $\alpha = 0.02, 0.03$ we see the~~ When $\alpha = 0.05$ in TD, the value is converging enough. In case of MC, $\alpha = 0.04$ is quite noisy.

Hence, I do not think there is a fixed value of α , at which either algo would perform better.

(6.5) RMS error of TD goes down and then up at high α as, $v_{\pi}(c)$ diverges from its initial estimate.

But, as we go on with the experiment the $v_{\pi}(c)$ diverges more, in turn, increasing the RMS error.

This ~~does~~ may not always occur and only be a function of how the values were initialized.

8. Given: Action selection is greedy.
Compare SARSA and Q-learning.

Let us look at Q-learning:

Here we just find $\max_a \{Q(s', a) \mid a \in A(s')\}$
instead of $Q(s', A')$.

SARSA:

Here, we find $Q(s', A')$ by R and updating $Q(s, A)$
by finding A' through s' .

greedily:
So, as the order of updating Q and finding A'
is different, it is not guaranteed the solutions
of the two algos will converge to the same
new action.

Hence, the algos are different.