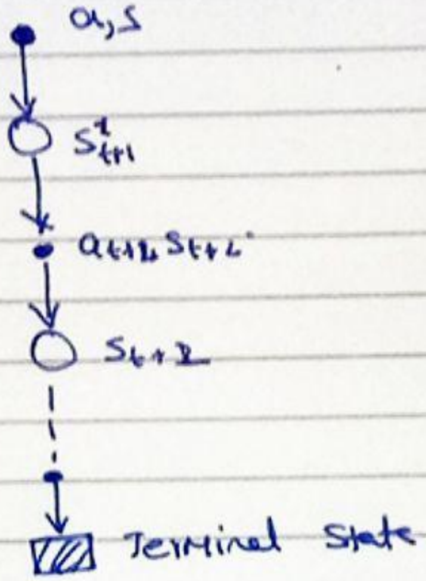
VASISHT DUDDU
2015137

Q2. Monte Carlo Estimation of $q_\pi(a,s)$



Q1. The idea to make MC more efficient is by incrementing $V(s)$ after each episode $S_1, A_1, R_1, \ldots S_T$ instead of storing the entire history requires only keeping track of previous $V(s)$ value.

For Incremental,

$$V(S_{t+1}) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)$$

Pseudo code:

Here, G_t can also be computed incrementally,

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

Similarly, for $Q(a,s)$, we have

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \frac{1}{N(S_t, A_t)} (G_t - Q(S_t, A_t))$$

Pseudo-code:

Initialize

$\pi(s) \in A(s)$ { Arbitrarily
$Q(s,a) \in R$

Loop for each episode:

(a) Choose S_0, A_0 such that all pairs have prob. > 0
(b) Generate episode from S_0, A_0 : $S_0, A_0, \ldots R_T$
(c) $G \leftarrow 0$
Loop for each $t = T-1, T-2, \ldots 0$
(d) $G \leftarrow \gamma G + R_{t+1}$
~~$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \frac{1}{}$~~
(e) $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
(f) $Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \frac{1}{N(S_t, A_t)} (G_t - Q(S_t, A_t))$
(g) $\pi(S_t) \leftarrow \underset{a}{\operatorname{argmax}}\, Q(S_t, a)$

Previously,
Execute (e), (f), (g) unless S_t, A_t appears in
$S_0, A_0, \ldots S_{t-1}, A_{t-1}$.

Using the new pseudocode, we can reduce one iteration of the (S,A) pairs in the episode.

Q3 $\rho_{t:T-1} = \dfrac{\prod_{k=t}^{T-1} \pi(A_k|S_k)\, p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)\, p(S_{k+1}|S_k, A_k)}$

$$= \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

Just like $E[\rho_{t:T-1} G_t | S_t = s] = v_\pi(s)$

by definition we can write,

$$E[\rho_{t:T-1} G_t | S_t = s, A_t = a] = q_\pi(s, a)$$

$$\rho_{t:T-1} G_t = \rho_{t:T-1} (R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \gamma^{T-t-1} R_{T})$$

$$q_\pi(s, a) = E_\pi[\rho_{t:T-1} G_t \;|S_t = s, A_t = a]$$

$$= E_\pi[\rho_t R_{t+1} + \rho_{t:T-1} G_{t+1} \;|S_t = s, A_t = a]$$

$$= \rho_t R(s, a) + \rho_t \gamma V_\pi(s)$$

$$= \rho_t R(s, a) + \rho_t \gamma \frac{\sum \rho_t G_t}{\sum \rho_t}.$$

School of Management

Q5. TD Learning Vs MC Learning

Case:

Consider I have a lot of experience driving home from work. Then I move to a new building & new parking lot but still use the same highway.

In this case, TD learning would be better than MC learning where we move to new building as this is just a change in our initial route and some of the states encountered during the rest of the episode will be same.

Example, on the drive home, many of the states are the same since we use the same highway & the value estimates for these states is very close to what we will compute for new building.

This is true for case where initial guess of value function is close to true function in which case convergence is faster.

Q8. Exercise 6.12

On choosing a greedy policy, the performance of SARSA is a little different from that of Q-Learning.

In Q-Learning: $Q(S_{t+1}, a_{t+1}) = \max_a Q(S_{t+1}, a)$

SARSA: $Q(S_{t+1}, a_{t+1}) = \varepsilon \cdot \underset{a}{\text{Mean}}\, Q(S_{t+1}, a) + (1-\varepsilon) \max_a Q(S_{t+1}, a)$

Since, Q-learning always chooses the optimal solution, it will always have the best (shortest) path to the goal.

The rate of convergences however can vary.

For instance, in the cliffhanger problem, SARSA will choose the sub-optimal path while Q-learning will choose the optimal path. However, the cummulative reward (performance) of SARSA might be better over the short run compared to Q-learning.

Over the long run, however, Q-learning performs better.

→ Excercise 6.3, 6.4, 6.5

Q6. Assuming $\alpha = 0.1$

For TD(0),

$$V(S_t) \leftarrow V(S_t) + 0.1(r_{t+1} + V(S_{t+1}) - V(S_t))$$

Computing V(A) starting from terminal state,

$$V(A) \leftarrow V(A) + 0.1(0 + \underset{\substack{\uparrow \\ V(S_{t+1}) \\ \text{↳ Terminal state}}}{0} - V(A)) = 0.9\,V(A)$$

$$= 0.45$$

On the first episode, the random walk results in 'left' action & we decreased the state value function by 0.05.

Growth of RMS Error

large values of α indicates a larger V(s) update at each time step. A larger value allows to converge to optima quickly compared to smaller value of α.

The up & down movement of RMS is due to the stochastic nature of updates made at each time step of the random walk.