Theory Answers

Solution:

```
Initialise:
    $\pi(s) \in A(s) \quad \forall s \in S$ .
    $Q(s,a) \in R \quad \forall s \in S,\ a \in A(s)$
    $Returns(s,a) \leftarrow$ empty list $\forall s \in S,\ a \in A(s)$.
    reward_t $\leftarrow$ zero array $\forall s \in S,\ a \in A(s)$
Loop for each episode:
        Random choose $S_0, A_0$
        Generate an episode $S_0, A_0$, following $\pi$
        $G \leftarrow \gamma G + R_{t+1}$
        Unless the pair $S_t, A_t$ appears
        $Q(s,a) = Q(s,a) + 0.1(\underset{Returns(s,a)}{\cancel{G}} - Q(s,a))$

        idx = np. $\underset{a}{\operatorname{argmax}}(Q(s,a))$

        reward_t [S] = reward_t[S] + $\frac{1}{episode}$ (G - reward_t[S])
```

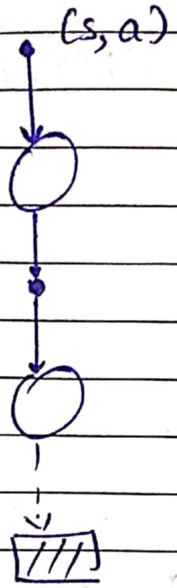
Calculating mean incrementally. And appending to the list

```
        ~~reward_store [S], episode~~
        $Returns(s,a) \leftarrow$ reward_t[S]
```

Thus Q(s,a) is incrementally updated with returns for each state and returns take the mean value of G and stores it, calculated incrementally.

Scanned by CamScanner

Solution 2 Backup diagram for q_π



Problem is to find an estimate for $q_\pi(s,a)$

Start in state $s \rightarrow$ taking action a and then follow a policy π.

Solution 3

We know ~~π~~ $v_\pi(s) = \sum_{a \in A(s)} q_\pi(s,a)\,\pi(a|s)$.

$$\frac{v_\pi(s)}{\sum_{a \in A(s)} \pi(a|s)} = q_\pi(s,a)$$

$$\frac{1}{\sum b(a|s)} \frac{\sum b(a|s)\, v_\pi(s)}{\sum \pi(a|s)} = q_\pi(s,a)$$

$$\frac{v_\pi(s)}{\sum b(a|s)\,\rho_{t:T-1}} \cancel{v_\pi(s)}) = q_\pi(s,a)$$

$$v_\pi(s) = \sum q_\pi(s,a)\,\rho_{t:T-1}\, b(a|s).$$

$$q_\pi(s,a) = \cancel{\epsilon}\ \frac{v_\pi(s)}{\sum b(a|s)\,\rho_{t:T-1}}$$

Notes

Appointment

Scanned by CamScanner

THURSDAY

11

JULY
2019

Thus $Q(s,a) = \frac{\sum \rho_{t:T-1} G(t)}{|I(s)|} \frac{1}{\sum b(a|s) \rho_{t:T-1}}$

$= \frac{G(t)}{|I(s)| \sum b(a|s)}$.

Thus $Q(s,a) = \frac{\sum G(t)}{b(a|s) |I(s)|}$

where $|I(s)|$ can be calculated as the number of times a state is visited for first visit or every visit.

Solution 3 -

Starting with the problem described in question, once we enter into a new building, some of the states encountered now will be same, . Thus starting with an initial guess, convergence can be achd faster.

Similar way can be seen over packet delivery in a channel. in case of any delays in delivery, the updates can be done as and when packets are delivered on the client side. Thus with an initial guess ~~fro~~ for values of these packet delays, the delay scene over the n/w can be seen faster.

Notes

Appointment

Solution 6

Solution 6.3 - If the state value of A only changed in initial episode, it signifies that the game ended in the left terminal state with reward 0.

Since episode = 1, there's no room for exploration and considering the aim to reach terminal state, it ends in left terminal state with reward 0, having a short action benefit.

True value for state A = $\frac{1}{6}$ = 0.17

Estimated value for state A = 0.425

% change = $\frac{0.425 - 0.17}{0.17} \times 100$ = 25%.

olution 6.4 - Had the step size parameter be wider, TD would perform better because on an average it will have more steps to explore before updating at next state and will push the agent towards the terminal state. hus the algorithms would have definitely worked better had there been a wider range of α. The improvement will be better in the initial phase but eventually there will be more fluctuations for RMSE.

Notes	Appointment

Scanned by CamScanner

Solution 6.5

The approximate value functions are initialised as 0.5 for all states and 0 for the terminal states.

RMSE = $\sqrt{(\text{true value} - \text{estimated value})^2}$

with higher α, the agent will be pushed more towards the terminal state in starting but eventually the fluctuations will start due to same value for all states and RMSE becomes more due to high α.

Large value of α implies more of $V(s)$ is updated at each timestep. Thus TD(0) becomes heavily dependent on the returns at each step of specific random walk sequence.
At smaller α, although learning takes more time to do but is affected by any perturbations at any time step.

Solution 8

Suppose action selection is greedy, then post each time step,

action ← arg max $Q(s', a)$

which will be chosen for next episode. The action selection will be same always.

Scanned by CamScanner