(1) We maintain a count for each state - action pair

$Q_n(S_t, A_t)$: action value estimate after ~~to~~ state S_t and action A_t has been selected n-1 times

$$Q_n(S_t, A_t) = \frac{\sum_{i=1}^{n} G_i(S_t, A_t)}{n}$$

$$= \frac{\sum_{i=1}^{n-1} G_i(S_t, A_t) + G_n(S_t, A_t)}{n}$$

$$= \frac{n-1}{n(n-1)} \sum_{i=1}^{n-1} G_i(S_t, A_t) + \frac{G_n(S_t, A_t)}{n}$$

$$= \frac{n-1}{n} \left[\frac{\sum_{i=1}^{n-1} G_i(S_t, A_t)}{n-1} \right] + \frac{G_n(S_t, A_t)}{n}$$

$$= \frac{n-1}{n} Q_{n-1}(S_t, A_t) + \frac{G_n(S_t, A_t)}{n}$$

$$Q_n(S_t, A_t) = Q_{n-1}(S_t, A_t) + \frac{1}{n}\left[G_n(S_t, A_t) - Q_{n-1}(S_t, A_t) \right]$$

$\therefore$ we can write the pseudocode of Monte Carlo ES as:-

Initialize:

$\pi(s) \in A(s)$ (arbitrarily), for all $s \in S$

$Q(s,a) \in R$ (arbitrarily) for all $s \in S$, $a \in A(s)$

$n(s,a) = 0$, for all $s \in S$, $a \in A(s)$

Loop forever (for each episode):

Choose $S_0 \in S$, $A_0 \in A(S_0)$ randomly st all pairs have probability > 0

Generate an episode from S_0, A_0 following π:

$A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G = 0$

Loop for each step of episode $t = T-1, T-2, \ldots, 0$:

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

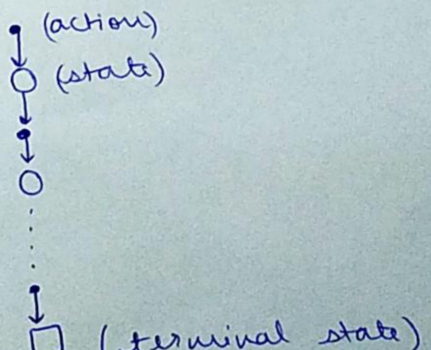
Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1, \ldots, S_{t-1}, A_{t-1}$:

$$n(S_t, A_t) = n(S_t, A_t) + 1$$

$$Q(S_t, A_t) = Q(S_t, A_t) + \frac{1}{n(S_t, A_t)} \left[G(S_t, A_t) - Q(S_t, A_t) \right]$$

$$\pi(S_t) = \arg\max_a Q(S_t, A_t)$$

(2) Backup diagram for Monte Carlo estimation of q_π:



(3) Given a starting state-action pair (S_t, A_t), probability of the subsequent state-action trajectory $S_{t+1}, A_{t+1} \ldots S_T$ occuring under policy π is:

$$Pr\{ \cancel{A_t}\, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi \}$$

$$= p(S_{t+1} \mid S_t, A_t)\, \pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})$$

$$= \frac{\pi(A_t \mid S_t)\, p(S_{t+1} \mid S_t, A_t)\, \pi(A_{t+1} \mid S_{t+1}) \cdots p(S_T \mid S_{T-1}, A_{T-1})}{\pi(A_t \mid S_t)}$$

$$= \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)\, p(S_{k+1} \mid S_k, A_k)}{\pi(A_t \mid S_t)}$$

where p is the state-transition probability function.

Relative ~~a~~ probability of the trajectory under target (π) and behaviour (b) policies:

$$\rho'_{t:T-1} = \frac{\left(\dfrac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)\, p(S_{k+1} \mid S_k, A_k)}{\pi(A_t \mid S_t)} \right)}{\left(\dfrac{\prod_{k=t}^{T-1} b(A_k \mid S_k)\, p(S_{k+1} \mid S_k, A_k)}{b(A_t \mid S_t)} \right)}$$

$$\rho'_{t:T-1} = \frac{b(A_t|S_t)}{\pi(A_t|S_t)} \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)\, p(S_{k+1}|A_k, S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)\, p(S_{k+1}|A_k, S_k)}$$

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

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

$$= \rho_{t+1:T-1}$$

$$q_\pi(s,a) = E\left[\rho'_{t:T-1} G_t \mid S_t = s,\ A_t = a\right]$$

$$= E\left[\rho_{t+1:T-1} G_t \mid S_t = s,\ A_t = a\right]$$

For first visit Monte Carlo method, (with weighted average)

$$Q(s,a) = \frac{\sum_{t \in \mathcal{T}(s,a)} \rho'_{t:T(t)-1} G_t}{\sum_{t \in \mathcal{T}(s,a)} \rho'_{t:T(t)-1}}$$

$$= \frac{\sum_{t \in \mathcal{T}(s,a)} \rho_{t+1:T(t)-1} G_t}{\sum_{t \in \mathcal{T}(s,a)} \rho_{t+1:T(t)-1}}$$

where $\mathcal{T}(s,a)$ is the set of all time steps in which state-action pair (s,a) is visited.

$T(t)$ denotes the 1st time of termination following time t

G_t is the return after t up through $T(t)$

$\therefore \{G_t\}_{t \in \mathcal{J}(s,a)}$ are returns that pertain to state action pair (s,a) & $\{\rho_{t+1:T(t)-1}\}_{t \in \mathcal{J}(s,a)}$ are the corresponding importance sampling ratios.

(5) TD method can update its estimates for state value just at the next time step, whereas Monte Carlo methods updates its state value estimates at the end of the episode.

$\therefore$ TD method would adapt much faster to a new parking lot than Monte Carlo method (which would wait till the person reach home to update its estimates)

$\therefore$ In the scenario when the person moves to a new building and a new parking lot. (but still enters the highway at the same place), TD updates are likely to be much better, atleast initially, due to its online update fashion. Whereas, Monte Carlo would struggle initially ~~due to~~ as it must wait till episode ends before it can increment the value estimates.

The same thing can occur in the original task as well (due to uncertainities of the environment - traffic delay due to construction work, rain, etc)

(8) No, Q-learning with greedy action selection is not the same as Sarsa.

In Sarsa, action taken at the next time step (A_{t+1}) is taken from Q function ($Q_t(S_t, A_t)$) ($A_{t+1} = \underset{a}{\operatorname{argmax}}\ Q_t(S_t, A_t)$ as we are doing greedy action selection)

However, in Q-learning, action taken at the next time step (A_{t+1}) is taken from the updated Q function ($Q_{t+1}(S_t, A_t)$)

$\therefore$ In Sarsa, action A_{t+1} is picked first, then Q function is updated & in Q-learning Q-function is updated first and then action A_{t+1} is picked.

$\therefore$ Q learning with greedy action selection is not the same as Sarsa. This can be seen from the case where next state $S_{t+1} = S_t$

Here, Sarsa will pick A_{t+1} greedily from $Q_t(S_t, A_t)$ whereas Q learning will pick A_{t+1} greedily from $Q_{t+1}(S_t, A_t)$ and thus the action A_{t+1} selected by both the algorithms might differ.