Temporal Difference Methods for Control

Rupam Mahmood

March 4, 2020

SARSA updates: idealized world vs. real-world

Time $t \rightarrow$

$t = 0$	**choose A_0 at S_0**	**Take A_0 & observe R_1, S_1**	**choose A_1 at S_1**	**SARSA update for S_0, A_0 using S_0, A_0, R_1, S_1, A_1**	**Take A_1 & observe R_2, S_2**	**choose A_2 at S_2**	**SARSA update for S_1, A_1 using S_1, A_1, R_2, S_2, A_2**	**Take A_2 & observe R_3, S_3** ...

In an idealized world (e.g., discrete MDP), time advances discretely and only after an action is taken

Therefore, the world stands still while an update (which can be expensive) is being computed

In the real world, time advances continuously and hence during the update

Therefore, after the update is complete, the world may not be at the same state any more, making the action stale

In the real world, action should be taken immediately after choosing/computing and the agent should wait a little bit to have its impact before observing

Mahmood A. R., Korenkevych, D., Komer, B. J., Bergstra, J. (2018). Setting up a reinforcement learning task with a real-world robot. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Difference between prediction and control often does not appear in the update

TD(0) for q_π (prediction): $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right]$

with experience being generated by following the (behavior) policy π, which is fixed

Sarsa (control): $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right]$

with experience being generated by following a soft-policy such as epsilon greedy

In both bases, the updates are exactly the same!

Therefore, the description of a method is not complete by just giving the update rule.
We also need to mention the policy for generating experience

On-policy vs. off-policy

Notice what the target of the update represents and whether the underlying policy of that matches the behavior policy

If it does, then it is on-policy

On-policy constant-α MC:

$$V^{MC}(S_t) \leftarrow V^{MC}(S_t) + \alpha \left[\underbrace{G_t}_{\textbf{target}} - V^{MC}(S_t) \right]$$

(Quiz: is it prediction or control?)

Behavior policy is π, and the target in expectation is v_π

Off-policy constant-α MC:

$$V^{MC}(S_t) \leftarrow V^{MC}(S_t) + \alpha \left[\rho_{t:T-1} G_t - V^{MC}(S_t)\right]$$

(Quiz: is it prediction or control?)

Behavior policy is b, and the target in expectation is v_π

Question:

What will be an off-policy TD(0) update for q_π?

On-policy TD(0) for v_π : $V(S_t) \leftarrow V(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right]$ **behavior policy is π**

An Off-policy TD(0) for v_π : $V(S_t) \leftarrow V(S_t) + \alpha \left[\rho_{t:t} \left(R_{t+1} + \gamma V(S_{t+1})\right) - V(S_t)\right]$ **behavior policy is b**

$$\rho_{t:t} = \frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}$$

On-policy TD(0) for q_π : $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right]$ **behavior policy is π**

Off-policy TD(0) for q_π : $?$

Expected Sarsa update relates to many methods

Expected Sarsa: $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_a \pi(a \mid S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t) \right]$

… is an on-policy prediction method for q_π	**when the behavior policy is a fixed policy π**
… is an off-policy prediction method for q_π	**when the behavior policy is a different fixed policy b**
… is an on-policy control method	**when the behavior policy is a soft policy π**
… is an off-policy control method	**when π is a soft policy and the behavior policy is different and exploratory**

Can expected SARSA be reduced to Q-learning in any particular case?

Write the pseudocode for Expected Sarsa by modifying one of the following:

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\mathit{terminal}, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\mathit{terminal}, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal