Temporal Difference Methods

for Control

Rupam Mahmood

March 2, 2020

Difference between prediction and control in pseudocode

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

^ Which is different than 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 $Q(terminal, \cdot) = 0$

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)

    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma Q(S', A') - Q(S, A)\right]$

    $S \leftarrow S'$; $A \leftarrow A'$;

  until S is terminal

approximating qpi

What would you modify in the above to get the pseudo code for TD(0)?

Tabular TD(0) for estimating v_π → estimate v_π is fixed.

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

Q-learning and on-policy vs. off-policy

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

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

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

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

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

Difference between Q-learning and SARSA in pseudocode

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 $Q(terminal,\cdot) = 0$

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)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  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 $Q(terminal,\cdot) = 0$

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'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
    $S \leftarrow S'$
  until S is terminal

$t = 0$ **choose action** **Take action & observe** **choose action** **Take action & observe** **choose action** **Take action & observe** **…**

Time $t \rightarrow$

Where would you put SARSA updates?

Where would you put Q-learning updates?

What happens if we switch the time of update for both?

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 $Q(terminal,\cdot) = 0$

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 $Q(terminal,\cdot) = 0$

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

Modify the Q-learning Pseudocode minimally to get SARSA

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

The equivalence of two SARSA algorithms breaks in the real world

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 $Q(terminal, \cdot) = 0$

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

R: immediate reward

SARSA: state, action pair for next state and action

Q: action value for

6.6 Expected Sarsa

Consider the learning algorithm that is just like Q-learning except that instead of the maximum over next state–action pairs it uses the expected value, taking into account how likely each action is under the current policy. That is, consider the algorithm with the update rule

$$\begin{aligned} Q(S_t, A_t) &\leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \mathbb{E}_\pi [Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t) \Big] \\ &\leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t) \Big], \end{aligned} \tag{6.9}$$

It could be on-policy or off-policy, and for prediction or control!

Question:

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

TD Control

Wed

[Notes]

When we say a policy is fixed, we mean its state to action probability, π is stationary, that is, does not change over time. In that sense, an ϵ-greedy policy with action values being learned is not a fixed policy. Bcz action values is always changing!!

[Identifying the target policy in TD methods]

The target policy can be identified through the target of the update:

$$\text{New-estimate} \doteq \text{previous-estimate} + \text{step-size}\,[\underbrace{\text{target} - \text{previous-estimate}}_{\text{error}}]$$

α

A method described only using an update rule is under-specified. We need to describe the behavior and target policies.

For example, which method is the following?

$$Q(S_t, A_t) \doteq Q(S_t, A_t) + \alpha\left[\underbrace{R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})}_{\text{target}} - Q(S_t, A_t)\right]$$

It is known as SARSA. It is an on-policy method meaning that the target and the behavior policies are the same. But SARSA can be either a prediction or a control method. Which one is it? Depends on the target policy! How is A_{t+1} (in the target) drawn? According to fixed or ϵ-greedy policy?

[Temporal difference learning algorithms for action values]

(for state values)

	Prediction	Control
on-policy	- SARSA for prediction Also known as TD(0) for qpi - Expected SARSA - (TD(0))	- SARSA for control - Expected SARSA
off-policy	- Expected SARSA - (TD(0))	- Expected SARSA - Q-learning

- Prediction vs. control is determined based on whether the target policy is fixed or ϵ-greedy.
- On-policy vs. off-policy is determined based on whether the behavior and the target policies are the same or different.

[Questions]

(1) Give the specification of the on-policy SARSA prediction method. ⇒ give the update, and describe the policy.

(Ans)

Update rule:

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

Target policy is fixed.

Behavior policy is the same as the target policy.

(2) Give the specification of the on-policy SARSA control method. ⇒ give the same update with SARSA prediction, and describe the policy.

(Ans)

Update rule:

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

Target policy is ϵ-greedy.

Behavior policy is the same as the target policy.

(3) Is the following expected SARSA method on-policy or off-policy and for prediction or for control? ⇒ for prediction since the target policy is fixed and don't have target policy improvement

Update rule:

$$Q(S_t, A_t) \doteq Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_c \pi(c|S_{t+1}) Q(S_{t+1}, c) - Q(S_t, A_t) \right]$$

off-policy { Target policy π is fixed.

Behavior policy is different than target policy.

④ Why Q-learning is off-policy? => target policy = π, behavior policy = μ.

⑤ Why Q-learning is a control method?

⑥ Why off-policy expected SARSA or Q-learning does not require importance sampling?

(Ans) Expected Sarsa is special case becomes Q learning!!

First of all, Q-learning can be subsumed within off-policy ^ expected SARSA in the following way:

$$Q(S_t, A_t) \doteq Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \sum_c \pi(c|S_{t+1}) Q(S_{t+1}, c) - Q(S_t, A_t) \Big],$$

(greedy based on Q. all probability is assigned to the greedy action since we are using the max. behavior policy is ε-greedy.)

where the target policy π is different than the behavior policy and is greedy.

Then it suffices to argue why off-policy ^ expected SARSA does not require importance sampling.

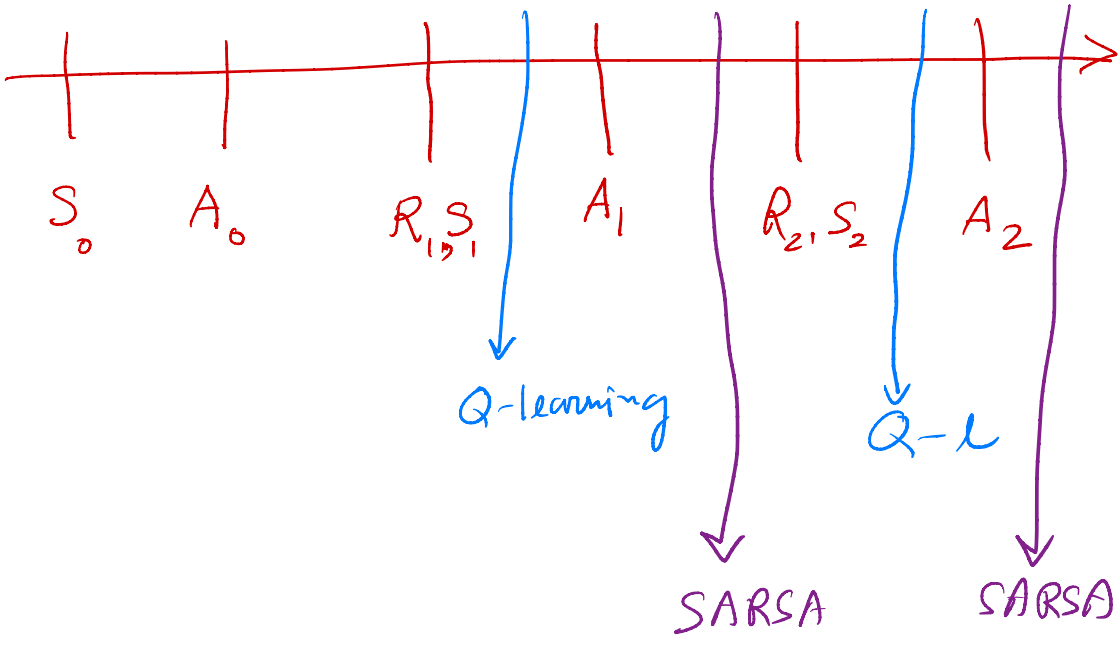
In off-policy methods, importance sampling is used to account for the discrepancy of (sampled) ^ action probabilities between the target and the behavior policies. => for expected Sarsa, no sampling action is accounted for.

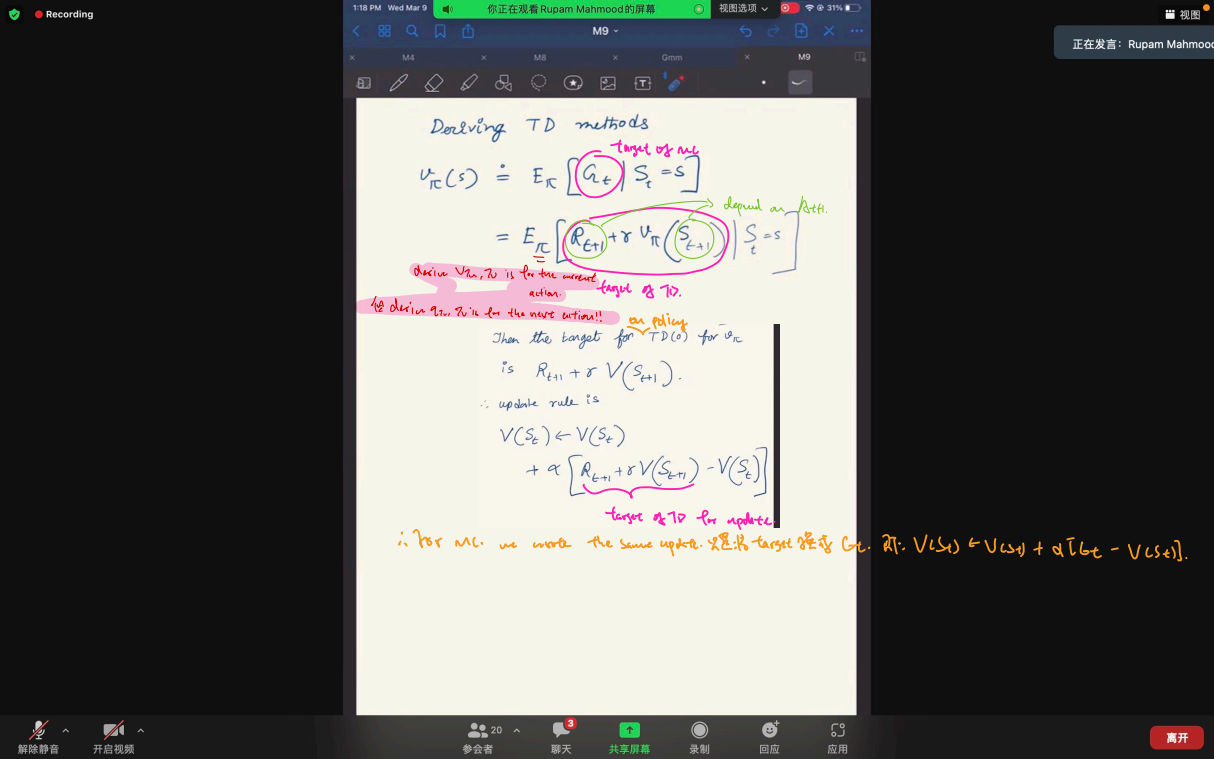
(See next page)

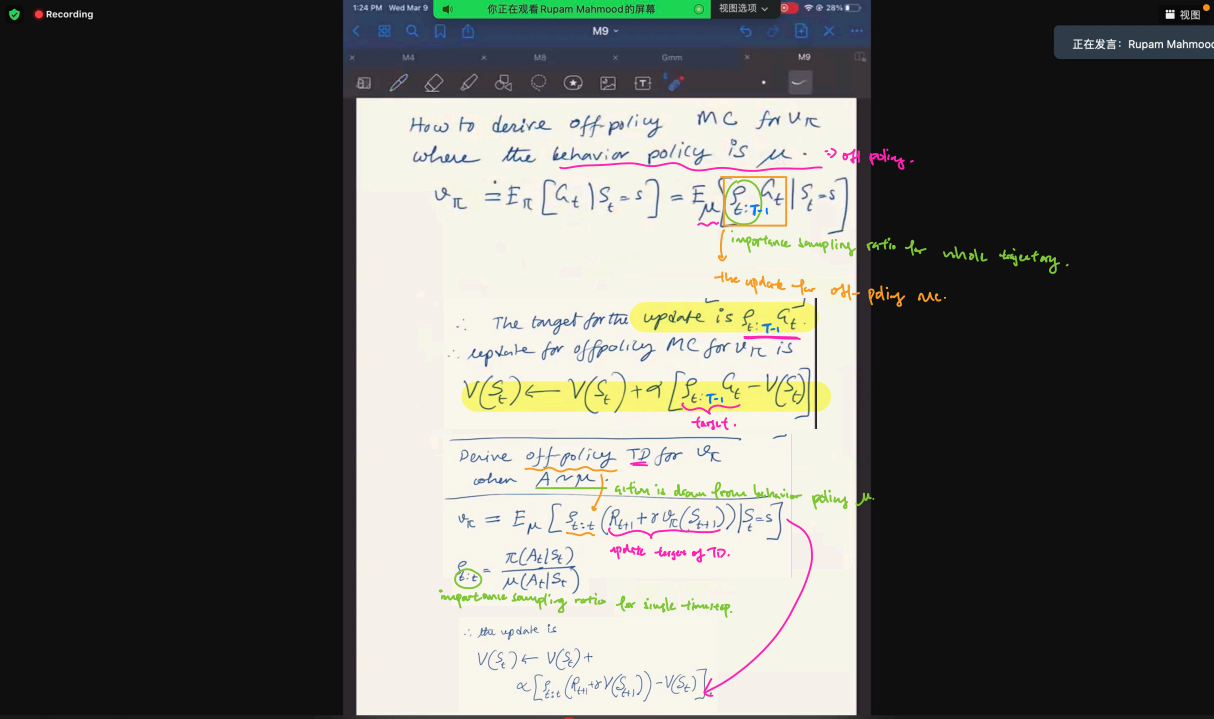
In action value methods, current action A_t is given. And the next action in expected SARSA target $\hat{A}$ is never sampled. Rather, the expectation is used directly:

$$\sum_c \pi(c \mid S_{t+1}) Q(S_{t+1}, c) = \mathbb{E}_\pi \left[Q(S_{t+1}, A_{t+1}) \mid S_{t+1} \right]$$

So no sampling discrepancy to account for.


S_0
A_0
R_1, S_1
A_1
R_2, S_2
A_2
Q-learning
Q-l
SARSA
SARSA





where p here is the state-transition probability function defined by (3.4). Thus, the relative probability of the trajectory under the target and behavior policies (the importance-sampling ratio) is

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k,A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k,A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}. \tag{5.3}$$

single time step of this importance sampling ratio.
-> multiply together give importance sampling ratio of the whole trajectory.

Derive offpolicy TD(0) for q_π
or equivalently offpolicy SARSA
for prediction, when $A \sim \mu$.

$$q_\pi(s,a) \doteq E_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

$$= E_\pi\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

desire q_π ; ?= for the next action!!

Random variable here.

$$= E_\mu\left[\rho_{t+1:t+1}\left(R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1})\right) \mid S_t = s, A_t = a\right]$$

target

∴ update

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t)$$

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

正在发言：