WDS

DataScience@Web

强化学习-Sarsa算法

目录

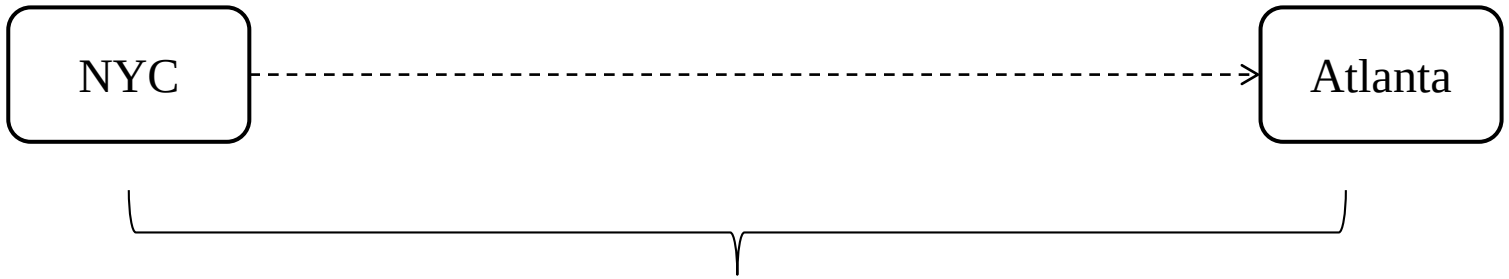
蒙特卡洛算法的优缺点

优点：免模型策略，通过多次采样产生多条轨迹近似期望，克服了模型未知给策略造成的困难。

缺点：每次需要执行一条完整的轨迹(trajectory)，在当前轨迹完成后得到最终的折扣奖励才能对之前轨迹的每个step进行策略更新，无法在进行一个step后就实时更新策略。

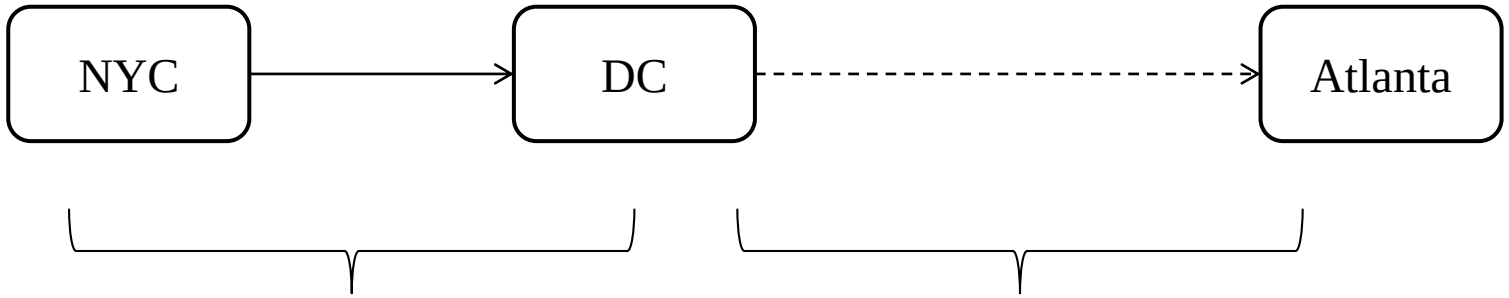
时序差分算法

Example: 从纽约开车到亚特兰大



Estimate： 1000min

Prediction



Actual:300min

Estimate： 600min

Target

时序差分算法

$G = R + \gamma \cdot R_1 + \gamma^2 R_2 + \gamma^3 R_3 + \gamma^4 R_4 + \ldots$ Prediction $\Leftarrow V(s_t; \pi) = E(G_t|s_t)$

$G_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \gamma^4 R_{t+4} + \ldots$

$\gamma \cdot (R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots) = G_{t+1}$

$G_t = R_t + \gamma \cdot G_{t+1}$

+

$G_t = R_t + \gamma \cdot G_{t+1}$

$E(R_t) + \gamma \cdot E(G_{t+1}|s_t) \rightarrow E(R_t) + \gamma \cdot V(S_{t+1}; \pi)$

$\approx\approx r_t + \gamma \cdot V(s_{t+1}; \pi)$

TD Target

时序差分算法

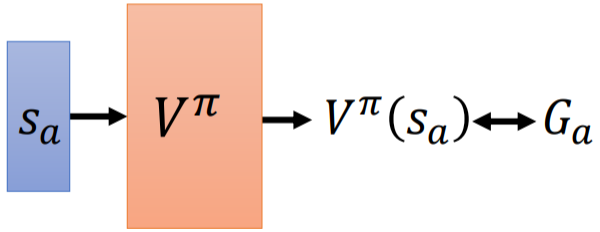
Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Initialize $V(s)$ arbitrarily (e.g., $V(s) = 0$, for all $s \in \mathcal{S}^+$)
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha\left[R + \gamma V(S') - V(S)\right]$
        $S \leftarrow S'$
    until S is terminal

TD Error(δs)

时序差分算法 VS 蒙特卡洛算法

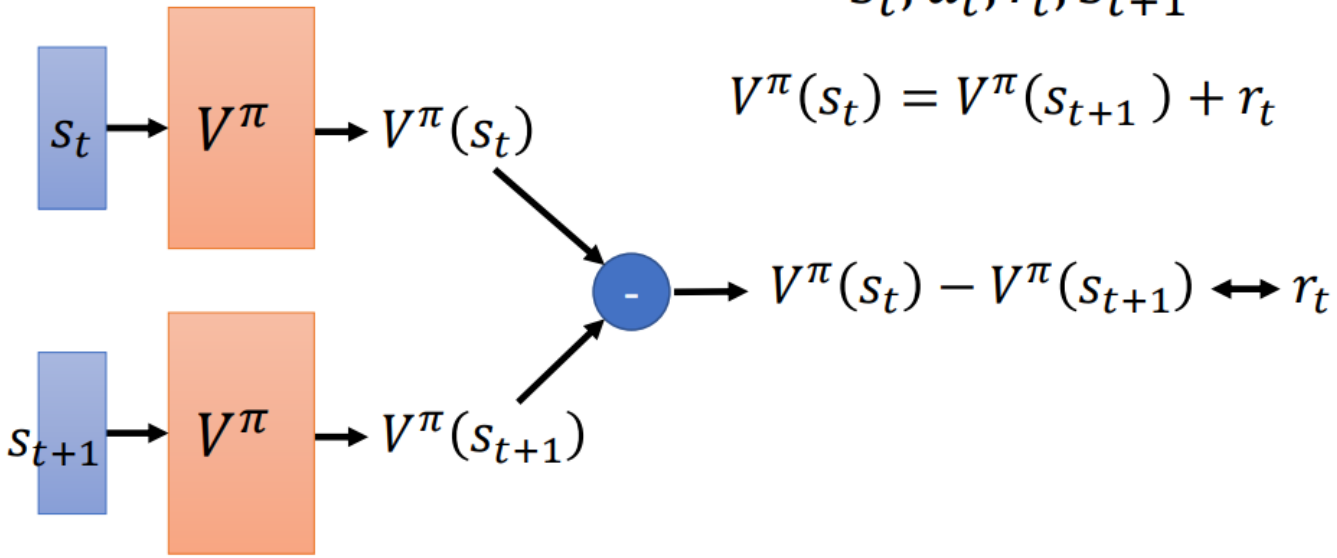
Monte-Carlo (MC) based approach



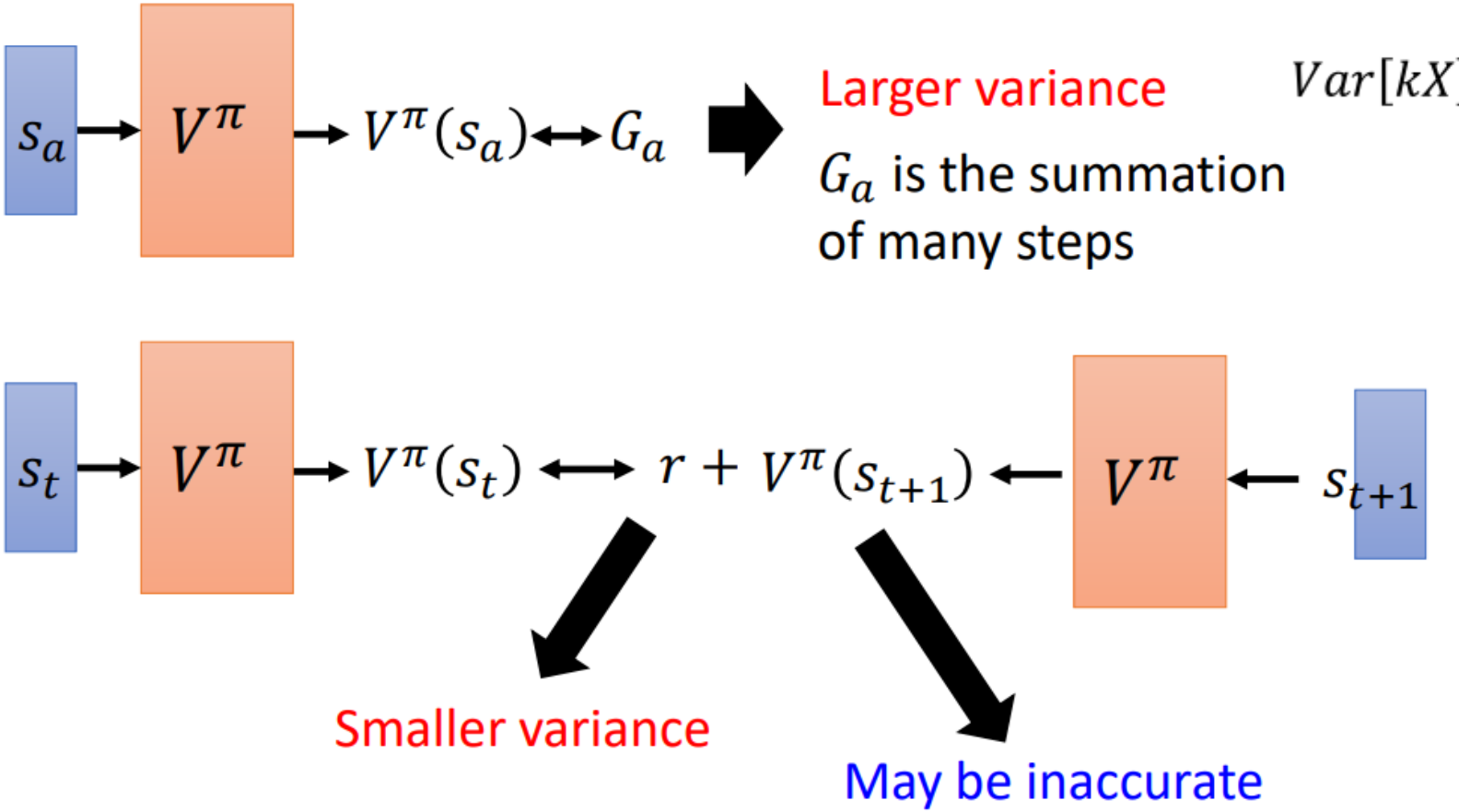
Temporal-difference (TD) approach

$$\cdots s_t, a_t, r_t, s_{t+1} \cdots$$

$$V^\pi(s_t) = V^\pi(s_{t+1}) + r_t$$



时序差分算法 VS 蒙特卡洛算法



时序差分算法 VS 蒙特卡洛算法

- $s_a, r = 0, s_b, r = 0$, END
- $s_b, r = 1$, END
- $s_b, r = 1$, END
- $s_b, r = 1$, END
- $s_b, r = 1$, END
- $s_b, r = 1$, END
- $s_b, r = 1$, END
- $s_b, r = 0$, END

$$V^{\pi}(s_b) = 3/4$$

$$V^{\pi}(s_a) = ? \quad 0? \quad 3/4?$$

Monte-Carlo: $V^{\pi}(s_a) = 0$

Temporal-difference:

$$V^{\pi}(s_a) = V^{\pi}(s_b) + r$$

3/4 3/4 0

SARSA算法

SARSA: state-action-reward-state-action

Use (s_t ,a_t ,r_t , s_{t+1} ,a_{t+1}) for updating Q-function

$V(s_t)$：给定状态s_t, 后续整体trajectory总回报的期望值

$Q(s_t, a_t)$：给定状态s_t, 采取动作a_t,后续整体trajectory总回报的期望值

SARSA算法

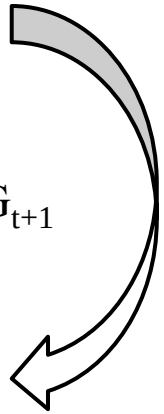
折扣回报推导公式：$G_t = R_t + \gamma \cdot G_{t+1}$

Prediction

$Q(s_t, a_t; \pi) = E(G_t | s_t, a_t)$

+

$G_t = R_t + \gamma \cdot G_{t+1}$



$E(R_t + \gamma \cdot G_{t+1} | s_t, a_t) \rightarrow E(R_t + \gamma \cdot Q(S_{t+1}, A_{t+1}))$

$\approx\approx r_t + \gamma \cdot Q(s_{t+1}, a_{t+1}; \pi)$ ⇨ TD Target

算法更新步骤

(1) 处于当前状态s_t, 按照策略π执行动作a_t, 得到奖励r_t, 状态转移为s_{t+1}

(2) 按照策略π执行动作a_{t+1}

(3) TD target: $y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$

(4) TD error: $\delta_t = y_t - Q_\pi(s_t, a_t)$

(5) Update: $Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \cdot \delta_t$

SARSA算法

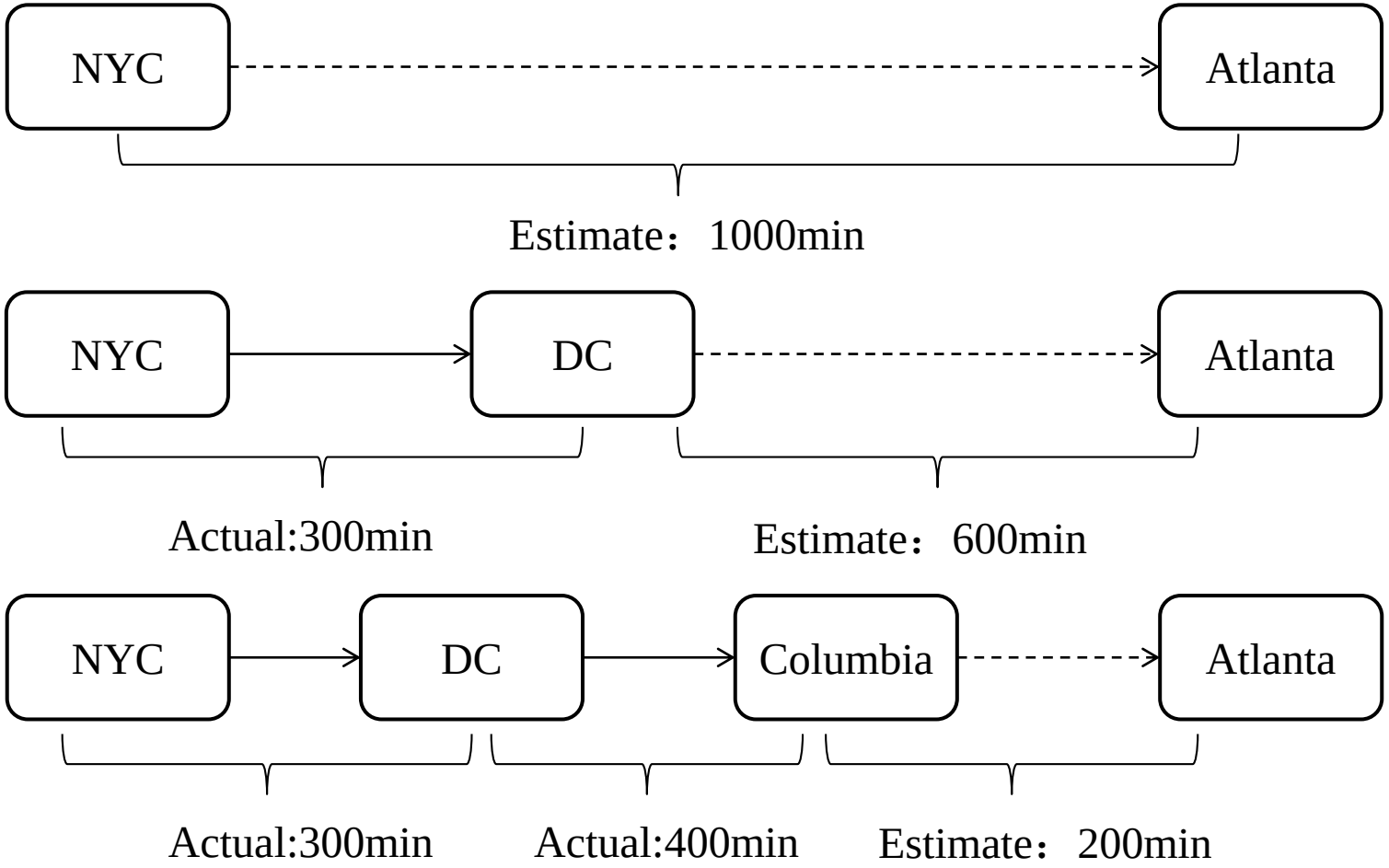
Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Repeat (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

TD算法：

$$V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$$

multi-step TD



multi-step TD

Multi-step Discount Reward: $G_t = \sum_{i=0}^{m-1} y^i R_{t+i} + \gamma^m \cdot G_{t+m}$

One-step TD target for Sarsa:

$$y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$$

m-step TD target for Sarsa:

$$y_t = \sum_{i=0}^{m-1} y^i r_{t+i} + \gamma^m \cdot Q_\pi(s_{t+m}, a_{t+m})$$

Forward-view TD(λ)

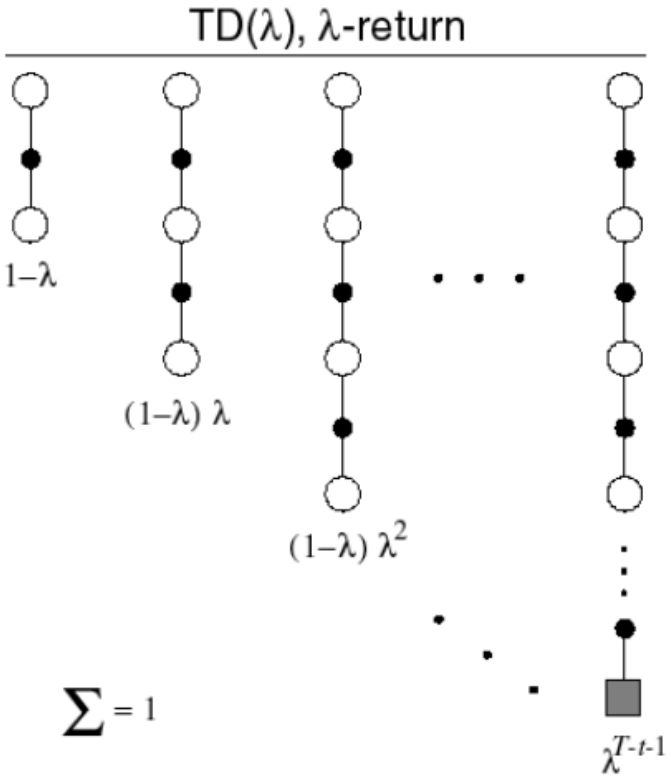
- Consider the following n-step returns for $n = 1, 2, \infty$:

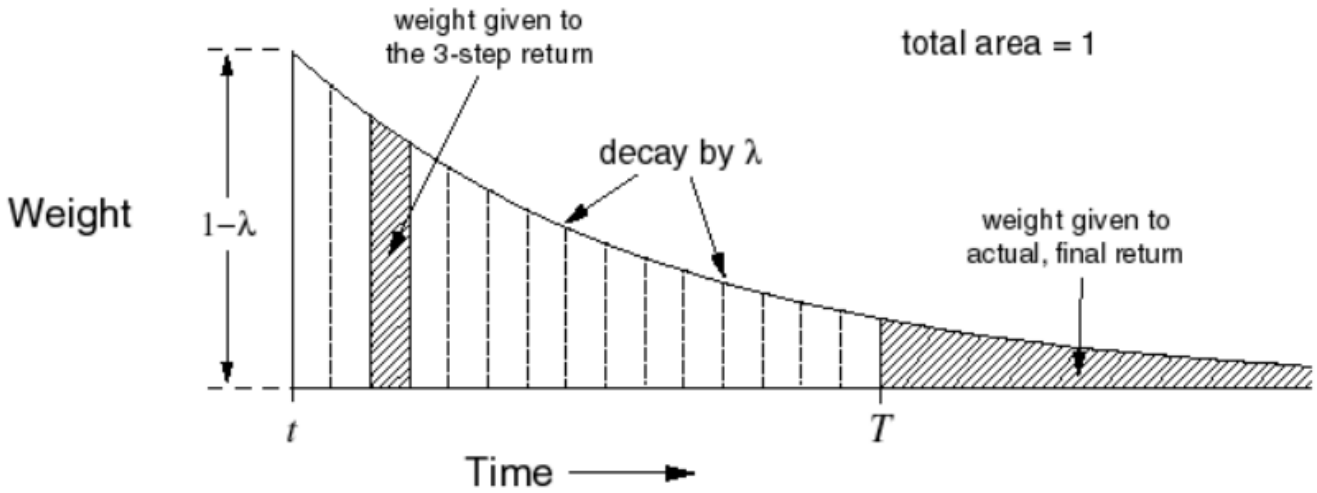
$$
\begin{array}{lll}
n = 1 & (TD) & G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\
n = 2 & & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\
\vdots & & \vdots \\
n = \infty & (MC) & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T
\end{array}
$$

- Define the n-step return

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

Forward-view TD(λ)

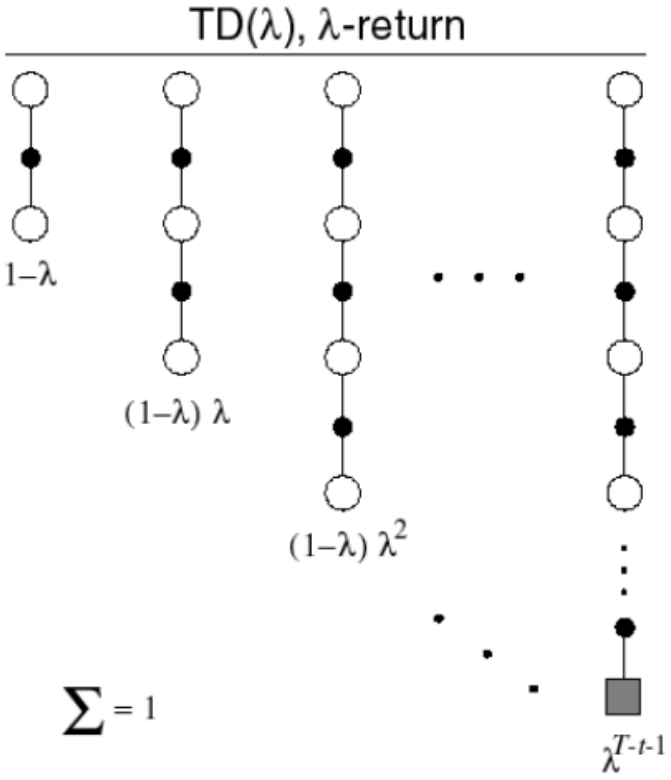


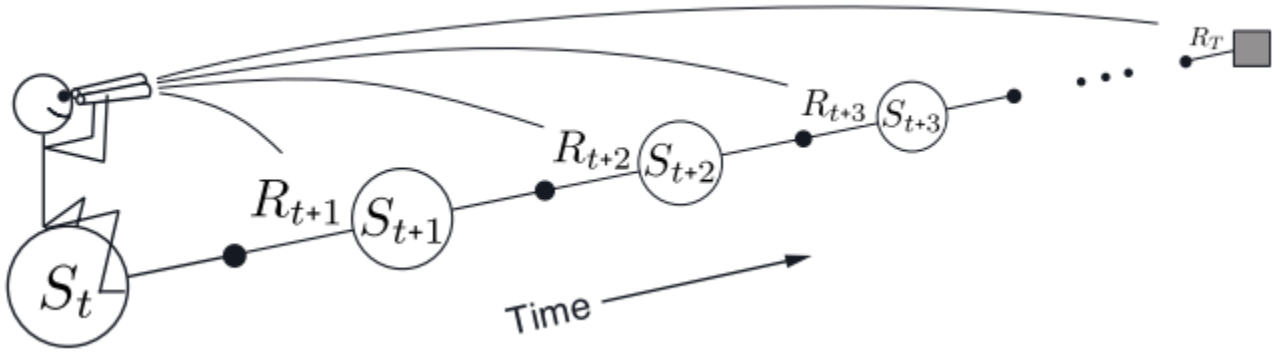


$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_{t:t+n} + \lambda^{T-t-1}G_t,$$

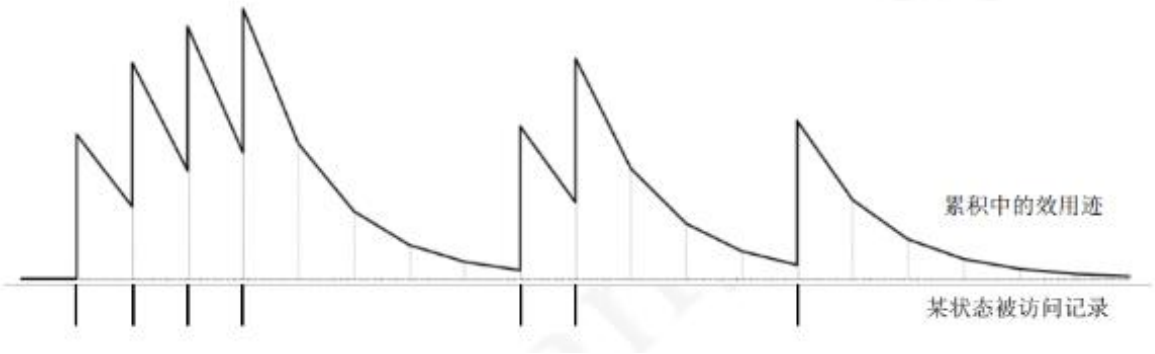
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^\lambda - V(S_t)\right)$$

Forward-view TD(λ)





Backward-view TD(λ)



Consider an episode where s is visited once at time-step k, TD(λ) eligibility trace discounts time since visit,

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$
$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases}$$

- When $\lambda = 0$, only current state is updated

$$E_t(s) = \mathbf{1}(S_t = s)$$
$$V(s) \leftarrow V(s) + \alpha\delta_t E_t(s)$$

- This is exactly equivalent to TD(0) update

$$V(S_t) \leftarrow V(S_t) + \alpha\delta_t$$

Backward-view TD(λ)

Consider an episode where s is visited once at time-step k, TD(λ) eligibility trace discounts time since visit,

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$
$$= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases}$$

When λ == 1

$$E_t(s) = \gamma E_{t-1}(s) + \mathbf{1}(S_t = s)$$
$$= \begin{cases} 0 & \text{if } t < k \\ \gamma^{t-k} & \text{if } t \geq k \end{cases}$$

Online-update

$$V(s) \leftarrow V(s) + \alpha\delta_t E_t(s)$$

Offline-update

$$\sum_{t=1}^{T-1} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T-1} \gamma^{t-k}\delta_t =$$

$$\delta_k + \gamma\delta_{k+1} + \gamma^2\delta_{k+2} + \ldots + \gamma^{T-1-k}\delta_{T-1}$$

Backward-view TD(λ)

$$\delta_k + \gamma\delta_{k+1} + \gamma^2\delta_{k+2} + \ldots + \gamma^{T-1-k}\delta_{T-1}$$

$$=R_{k+1} + \gamma V(S_{k+1}) - V(S_k)$$
$$+ \gamma R_{k+2} + \gamma_2 V(S_{k+2}) - \gamma V(S_{k+1})$$
$$+ \gamma^2 R_{k+3} + \gamma^3 V(S_{k+3}) - \gamma^2 V(S_{k+2})$$
$$.$$
$$.$$
$$+\gamma^{T-1-k} R_t + \gamma^{T-k} V(S_T) - \gamma^{T-1-k} V(_{T-1})$$
$$=R_{k+1} + \gamma R_{k+2} + \gamma^2 R_{k+3} \ldots + \gamma^{T-1-k} R_T - V(S_k)$$
$$=G_k - V(S_k)$$

$$\delta_t E_t(s) = \alpha \sum_{t=k}^{T-1} \gamma^{t-k}\delta_t = \alpha (G_k - V(S_k))$$

Backward-view TD(λ)

$$\sum_{t=1}^{T} \alpha \delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k} \delta_t = \alpha \left(G_k^\lambda - V(S_k) \right)$$

Backward-TD(λ) ==? Forward-TD(λ)

$$\begin{aligned}
G_t^\lambda - V(S_t) &= -V(S_t) \quad + \quad (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma V(S_{t+1})\right) \\
&\quad + \quad (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})\right) \\
&\quad + \quad (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})\right) \\
&\quad + \quad \ldots \\
&= -V(S_t) \quad + \quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma V(S_{t+1}) - \gamma\lambda V(S_{t+1})\right) \\
&\quad + \quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma V(S_{t+2}) - \gamma\lambda V(S_{t+2})\right) \\
&\quad + \quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma V(S_{t+3}) - \gamma\lambda V(S_{t+3})\right) \\
&\quad + \quad \ldots \\
&= \quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right) \\
&\quad + \quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma V(S_{t+2}) - V(S_{t+1})\right) \\
&\quad + \quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma V(S_{t+3}) - V(S_{t+2})\right) \\
&\quad + \quad \ldots \\
&= \delta_t + \gamma\lambda\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \ldots
\end{aligned}$$

Sarsa(λ)

$$E_0(s) = 0$$
$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

Eligibility trace in TD(λ)

$$E_0(s,a) = 0$$
$$E_t(s,a) = \gamma\lambda E_{t-1}(s,a) + \mathbf{1}(S_t = s, A_t = a)$$

Eligibility trace in Sarsa(λ)

TD target: $y_t = r_t + \lambda \cdot Q_\pi(s_{t+1}, a_{t+1})$

TD error: $\delta_t = y_t - Q_\pi(s_t, a_t)$

Update: $Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \cdot \delta_t$

Update in Sarsa

TD target: $y_t = r_t + \lambda \cdot Q_\pi(s_{t+1}, a_{t+1})$

TD error: $\delta_t = y_t - Q_\pi(s_t, a_t)$

Update: $Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \cdot \delta_t \cdot E_t(s,a)$

Update in Sarsa(λ)

Sarsa(λ)

Initialize $Q(s,a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$
Repeat (for each episode):
  $E(s,a) = 0$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$
  Initialize S, A
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)
    $\delta \leftarrow R + \gamma Q(S', A') - Q(S, A)$
    $E(S, A) \leftarrow E(S, A) + 1$
    For all $s \in \mathcal{S}, a \in \mathcal{A}(s)$:
      $Q(s,a) \leftarrow Q(s,a) + \alpha \delta E(s,a)$
      $E(s,a) \leftarrow \gamma \lambda E(s,a)$
    $S \leftarrow S'$; $A \leftarrow A'$
  until S is terminal

THANK YOU