Temporal Difference Method

Similar to the MC method, temporal difference methods only require experiences, i.e., sample trajectories (sequence of states, actions, rewards).

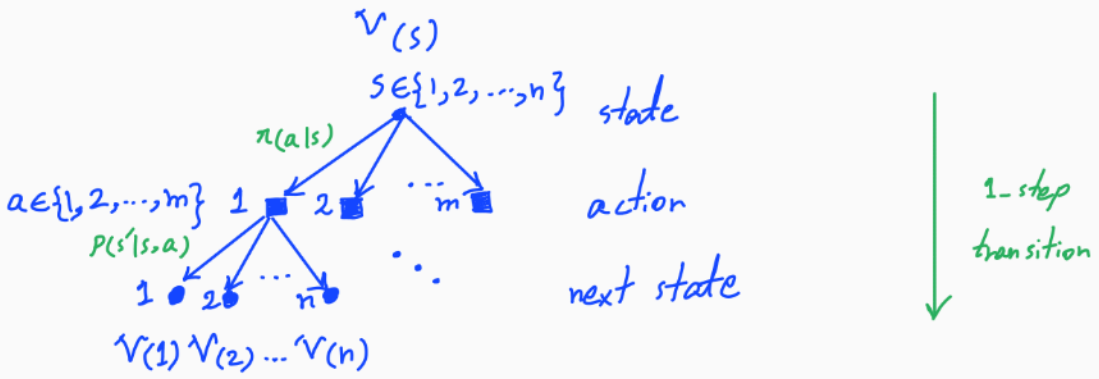
Model_free method

Temporal difference: Using dynamic programming over estimated values to learn online from incomplete episodes.

TD for policy evaluation: Given a stationary policy π, we want to compute

$$V^{\pi}(s) = \mathbb{E}_{\substack{A_t \sim \pi \\ S_{t+1} \sim P(\cdot \mid S_t, A_t)}} \left[\sum_{t=0}^{T} \gamma^{t} R(S_t, A_t) \mid S_0 = s \right]$$

Idea: Estimate the expected return using the sample immediate reward and the current estimate of the future expected return value function).



Let $\hat{v}^\pi$ be the current estimate of the value function of π

$$\longrightarrow v^\pi(s) = \mathbb{E}_{\substack{A_t\sim\pi \\ S_{t+1}\sim P(\cdot|S_t,A_t)}}\left[\sum_{t=0}^{T}\gamma^t R(S_t,A_t) \,\middle|\, S_0=s\right]$$

$$= \mathbb{E}_{\substack{A_t\sim\pi \\ S_{t+1}\sim P(\cdot|S_t,A_t)}}\left[R(S_0,A_0)+\gamma\sum_{t=1}^{T}\gamma^t R(S_t,A_t) \,\middle|\, S_0=s\right]$$

Bellman consistency equation

$$v^\pi(s) = \mathbb{E}_{a,s'}\left[R(s,a) + \gamma v^\pi(s')\right]$$

$$= \mathbb{E}_{A_0\sim\pi(S_0)}\left[R(S_0,A_0) + \gamma\,\mathbb{E}_{s'\sim P(\cdot|S_0,A_0)}\left[\overbrace{\mathbb{E}_{\substack{A_t\sim\pi \\ S_{t+1}\sim P(\cdot|S_t,A_t)}}\left[\sum_{t=0}^{T}\gamma^t R(S_t,A_t) \,\middle|\, S_0=s'\right]}^{v^\pi(s')}\right]\right]$$

$$\longrightarrow \hat{v}^\pi(s) = \frac{1}{\underset{\#\text{ samples}}{N}}\sum_{i=1}^{N}\left(R(s,\underset{\text{sample } i}{a^i}) + \gamma\hat{v}(\underset{\text{sample } i}{s'^i})\right)$$

cannot be done online for $N>1$

↓

incremental computation of mean

$$\hat{v}^\pi_{N(s)}(s) = \frac{1}{N(s)}\sum_{i=1}^{N(s)} G_i^s$$

$$\hat{v}^\pi_{N(s)+1}(s) = \hat{v}^\pi_{N(s)}(s) + \frac{1}{N(s)+1}\left(G^s_{N(s)+1} - \hat{v}^\pi_{N(s)}(s)\right)$$

equivalent for certain choice of $\alpha = \frac{1}{N(s)+1}$

Recall the MC update:

$$\underset{\text{for all states visited in the episode}}{\hat{v}^\pi(s)} \longleftarrow \hat{v}^\pi(s) + \underset{\text{Learning rate / step size}}{\alpha}\Big(\underbrace{\overset{\text{MC target}}{G} - \hat{v}^\pi(s)}_{\text{MC error}}\Big)$$

$\longrightarrow$

TD(0) update:

$$\underset{\text{for the state just visited}}{\hat{v}^\pi(s)} \longleftarrow \hat{v}^\pi(s) + \underset{\text{Learning rate / step size}}{\alpha}\Big(\underbrace{\overbrace{R(s,a)+\gamma\hat{v}^\pi(s')}^{\text{TD target}} - \hat{v}^\pi(s)}_{\text{TD error}}\Big)$$

TD (0) method

Iterative method:

```
• Initialize $\hat{v}^{\pi}$
• Repeat
    • Sample $s_0 \in S$
    • Repeat until the episode terminates
        • Take action a according to $\pi(a|s)$ at s
        • observe reward $R(s,a)$ and next state $s'$
        • $\hat{v}^{\pi}(s) \longleftarrow \hat{v}^{\pi}(s) + \alpha\left[R(s,a) + \gamma \hat{v}^{\pi}(s') - \hat{v}^{\pi}(s)\right]$
        • Update the current state    $s \longleftarrow s'$
```

$$\begin{bmatrix} \hat{v}^{\pi}(1) \\ \hat{v}^{\pi}(2) \\ \vdots \\ \hat{v}^{\pi}(n) \end{bmatrix}_{|S| \times 1}$$

$$s = 2 \xrightarrow{\pi} a = 1 \xrightarrow{p} s' = 4 \,,\; r = R(2,1)$$

$$\hat{v}^{\pi}(2) \longleftarrow \hat{v}^{\pi}(2) + \alpha\left(r + \gamma \hat{v}^{\pi}(4) - \hat{v}^{\pi}(2)\right)$$

TD for policy optimization: We want to find an (approximately) optimal policy π^* through iterative policy evaluation and policy improvement.

Idea:

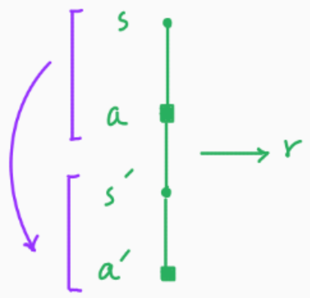
- Evaluate the policy by estimating the Q function using the TD method in an online fashion.
- Improve the policy using the estimated Q function in a greedy manner.

SARSA $\longrightarrow$ s, a, r, s', a'

Iterative method:

- Initialize $\hat{Q}$ s.t. $\hat{Q}(s,a)=0$ for terminal states
- Initialize π according to $\hat{Q}$ (e.g., ε_greedy)
- Repeat
    - Sample $s_0 \in \mathcal{S}$
    - Select action a according to $\pi(a|s)$ at s_0
    - Repeat until the episode terminates
        - Take action a
        - observe reward $R(s,a)$ and next state s'
        - Select action a' according to $\pi(a|s)$ at s'
        - policy evaluation • $\hat{Q}(s,a) \longleftarrow \hat{Q}(s,a) + \alpha \left[\underbrace{\overbrace{R(s,a) + \gamma \hat{Q}(s',a')}^{\text{TD target}} - \hat{Q}(s,a)}_{\text{TD error}}\right]$
        - policy improvement • Build $\pi(a|s)$ according to $\hat{Q}$ (e.g., ε_greedy)
        - Update the current state and the current action

$s \longleftarrow s'$ $\qquad$ $a \longleftarrow a'$



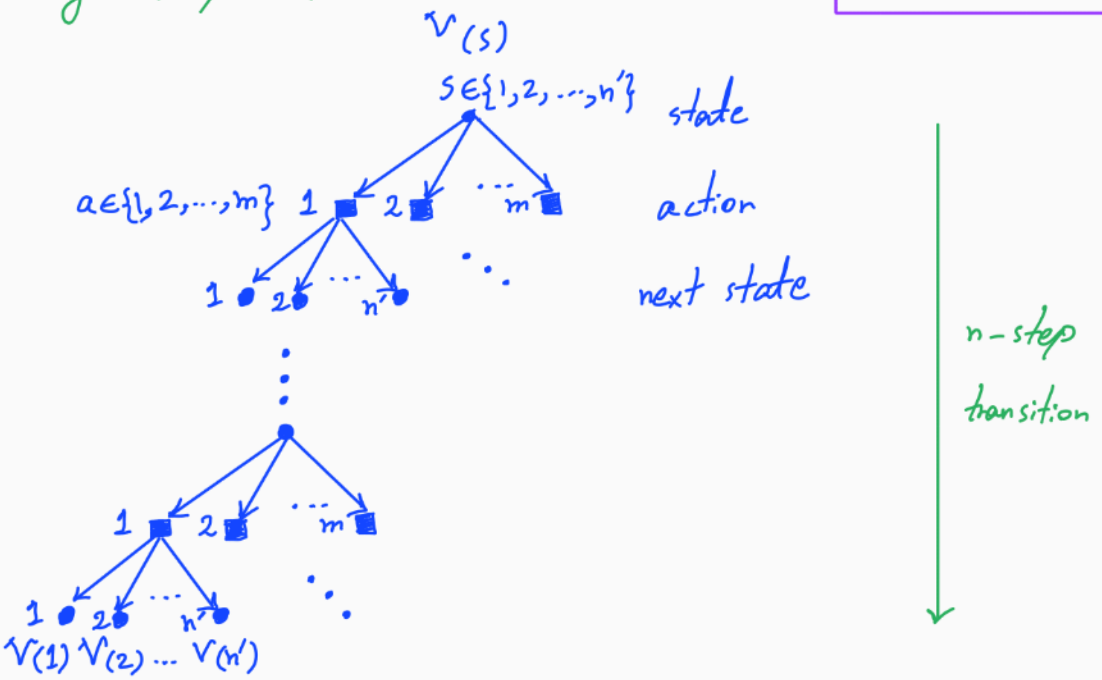
Note [n-step TD method]: Estimate the expected return using n sample immediate rewards and the current estimate of the future expected return (value function) after these n steps.

TD target → n-step return

$$\hat{V}^{\pi}(s_t) \longleftarrow \hat{V}^{\pi}(s_t) + \alpha\,(\underbrace{\overset{\text{TD target}}{G_t^{t+n}} - \hat{V}^{\pi}(s_t)}_{\text{TD error}})$$

Learning rate / step size

$$\boxed{G_t^{t+n} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n \hat{V}^{\pi}(s_{t+n})}$$



Note [TD(λ) method]: Updates the expected return using a weighted average of n-step returns over varying n.

TD target → weighted average of n-step returns

$$\hat{V}^{\pi}(s_t) \longleftarrow \hat{V}^{\pi}(s_t) + \alpha\,(\underbrace{\overset{\text{TD target}}{G_t^{\lambda}} - \hat{V}^{\pi}(s_t)}_{\text{TD error}})$$

Learning rate / step size

$$\boxed{G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{t+n} \qquad \lambda \in [0,1]}$$

* Backward view of TD(λ) using the idea of <u>eligibility traces</u> provides an incremental approach to approximate the forward view.