Temporal-difference Learning

(MC ideas + DP ideas)

MC ideas ↑ learn from experience

DP ideas ↑ update estimates based in part on other learned estimates, without waiting for a final outcome

prediction: estimate value function V_π for a given policy π

control: find an optimal policy

MC / constant-α MC:

Full episode

$$V(S_t) \leftarrow V(S_t) + \alpha [G_t - V(S_t)]$$

TD(0) / one-step TD:

$$V(S_t) \leftarrow V(S_t) + \alpha [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

sample

current estimate

next time step

$$V_\pi(s) \doteq \mathbb{E}_\pi [G_t \mid S_t = s] \quad \text{(MC)}$$

$$= \mathbb{E}_\pi [R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$= \mathbb{E}_\pi [R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s] \quad \text{(TD(0))}$$

"sample updates"

Looking ahead to a sample successor state (or state-action pair)

TD error :

$$V(S_t) \longleftrightarrow R_{t+1} + \gamma V(S_{t+1})$$

$$\delta_t \doteq R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

MC error :

$$G_t - V(S_t) = \sum_{k=t}^{T-1} \gamma^{k-t} \delta_k$$

TD vs DP

DON'T require a model

TD vs MC ← wait until the end of an episode

consider transitions from state-action pair to state-action pair, and learn the values of state-action pairs.

Sarsa: On-policy TD control

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

$(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$

对比:

TD: $V(S_t) \leftarrow V(S_t) + \alpha\left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right]$

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: Off-policy TD control

Watkins, 1989

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

Expected Sarsa

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \mathbb{E}_\pi\left[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}\right] - Q(S_t, A_t)\right]$$

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

Double Learning

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q_2(S_{t+1}, \arg\max_a Q_1(S_{t+1}, a)) - Q(S_t, A_t)\right]$$