◦ 时序差分算法

(1) MC Method:

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

其中 G_t 要完成一次采样才能知道.

(2) 时序差分:

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

$$V_\pi(s) = E_\pi[G_t | S_t = s]$$

$$= E_\pi[\sum \gamma^k R_{t+k} | S_t = s]$$

$$= E_\pi[R_t + \sum_{k=1}^{t=\infty} \gamma^k R_{t+k} | S_t = s]$$

$$= E_\pi[R_t + \gamma V_\pi(S_{t+1}) | S_t = s]$$

*: 单步更新, 完成了策略评估.

(3) 策略提升: 估计 $Q(s,a)$, 然后取 max.

Sarsa Algorithm: $s - a - r - s' - a'$

for $e=1 \to E$:

初始状态 s.

ε-greedy → 选择 a.

for $t=1 \to T$:

得到 r' 与 s' (环境反馈)

$$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma Q(s',a') - Q(s,a)]$$

$$s \leftarrow s',\ a \leftarrow a'$$

*: 主要维护一个 Q-table.

(4) 多步时序差分:

$$G_t = r_t + \gamma r_{t+1} + \cdots \gamma^n Q(S_{t+n}, a_{t+n})$$

(5) Q-Learning:

$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha [R_t + \gamma \max Q(S_{t+1}, a) - Q(S_t$

实际上, Q-learning 直接估计 Q^* (off-policy)

Sarsa 在估计当前 ε-greedy 下的 $Q(s, a)$ (on-policy)

(6) 采样策略: π_{sample}

目标策略: π_{target}

on-policy: $\pi_{sample} = \pi_{target}$

off-policy: $\pi_{sample} \neq \pi_{target}$