model-based RL: algorithms can directly call $p()$

model-free RL: only have access to $p()$ via trajectories

- on policy (SARSA) : we need trajectories using (approx) the current agent
- off policy (Q-learning) : we can use any samples for learning

same rule for choosing actions based on state
π, Q, ε-greedy Q

Model-free methods $\rightarrow$ only access trajectories

Temporal-difference TD(0) (S&B 6.1) $\pi \rightarrow V$

$(s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, \ldots\ldots) \leftarrow$ trajectory

$(s_t, a_t, r_{t+1}, s_{t+1}) \leftarrow$ piece of trajectory

$$V_\pi(s_t) = E_{a_t \sim \pi(\cdot|s_t)} E_{s_{t+1} \sim p(\cdot|s_t, a_t)} [r_{t+1} + \gamma V_\pi(s_{t+1})]$$

model-based method, $E \rightarrow \sum_{a_t}, \sum_{s_{t+1}}$

model-free method, sample a_t, s_{t+1}

~~$V_\pi(s_t) = r_{t+1} + \gamma V_\pi(s_{t+1})$ where $(s_t, a_t, r_{t+1}, s_{t+1})$ is a sample~~

doesn't work, because one sample is not enough

$$V_\pi(s_t) \sim \underbrace{r_{t+1} + \gamma V_\pi(s_{t+1})}_{\text{target}}$$

want to average these

– use incremental average

$$V_{avg} = \frac{V_n + (1-\alpha) V_{n-1} + (1-\alpha)^2 V_{n-2} + \dots}{\sum_i (1-\alpha)^i}$$

$$V_\pi(s_t) = (1-\alpha) V_\pi(s_t) + \alpha \left(\underbrace{r_{t+1} + \gamma V_\pi(s_{t+1})}_{\text{target}} \right)$$

Very noisy improvement to V_π

α = learning rate $\in (0, 1)$

$$V_\pi(s_t) = V_\pi(s_t) + \alpha \underbrace{\left(r_{t+1} + \gamma V_\pi(s_{t+1}) - V_\pi(s_t) \right)}_{\delta_t \,=\, \text{incement}}$$

where $a_t \sim \pi(\cdot \mid s_t)$ and $s_{t+1} \sim p(\cdot \mid s_t, a_t)$

sampled from

SARSA (S & B 6.4)

$Q \to$ better Q

$\cancel{\pi \to Q}$

Use traj segments $\begin{array}{ccccc} S & A & R & S & A \\ (s_t, & a_t, & r_{t+1}, & s_{t+1}, & a_{t+1}) \end{array}$

$\ldots, s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, r_{t+2}, s_{t+2}, a_{t+2}, r_{t+3}, \ldots$; $s_{t+1} \to a'_{t+1}$

$$Q(s_t, a_t) = r_{t+1} + \gamma E_{s_{t+1} \sim p(\cdot|s_t, a_t)} \left[E_{a_{t+1} \sim \pi(\cdot|s_{t+1})} \left[Q(s_{t+1}, a_{t+1}) \right] \right]$$

simplified: $$Q(s_t, a_t) = r_{t+1} + \gamma E[Q(s_{t+1}, a_{t+1})]$$
$$= E[\underbrace{r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})}_{\text{target}}]$$

"on-policy" learning

targets should be sampled from E distribution

$$Q(s_t, a_t) \leftarrow (1-\alpha) Q(s_t, a_t) + \alpha (\underbrace{r_{t+1} + \gamma Q(s_{t+1}, a'_{t+1})}_{\text{target}})$$

$a'_{t+1} = \arg\max_a Q(s_{t+1}, a)$

$$= Q(s_t, a_t) + \alpha \underbrace{(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))}_{\delta_t = \text{increment}}$$

$(s_0, a_0, r_1, s_1, a_1)$

$(17, 3, 2, 2, 1)$

$$Q = \begin{bmatrix} \cdot & \\ & \cdot \end{bmatrix} \; 25, \quad 4$$

$0 \to 0.2$

Given $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$

Update $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta_t$

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

Choose actions:

greedy: $a_t = \arg\max_a Q(s_t, a)$ ← best choice based on current Q function (no exploration) (all exploitation)

ε-greedy: $a_t = \begin{cases} \text{random } a & \text{with probability } \varepsilon \\ \arg\max_a Q(s_t, a) & \text{otherwise} \end{cases}$

exploration/exploitation tradeoff param.

ε-greedy is hopefully close to optimal when Q is close to optimal

Q-learning (S&B 6.5)

Use $(s_t, a_t, r_{t+1}, s_{t+1})$

Won't assume that are chosen from Q

off policy

$$Q(s_t, a_t) = r_{t+1} + \gamma \mathbb{E}_{s_{t+1} \sim p(\cdot \mid s_t, a_t)} \left[\mathbb{E}_{a_{t+1} \sim \pi(\cdot \mid s_{t+1})} \left[Q(s_{t+1}, a_{t+1}) \right] \right]$$

$$a_{t+1} = \arg\max_a Q(s_{t+1}, a)$$

$$= r_{t+1} + \gamma \mathbb{E}_{s_{t+1} \sim p(\cdot \mid s_t, a_t)} \left[\max_a Q(s_{t+1}, a) \right]$$

Given a sample s_{t+1}:

$$Q(s_t, a_t) \approx r_{t+1} + \gamma \max_a Q(s_{t+1}, a)$$

$$Q(s_t, a_t) \leftarrow (1-\alpha) Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \right)$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big(\underbrace{r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)}_{\delta_t \text{ increment}}\Big)$$

SARSA

$$Q(s_t, a_t) \leftarrow (1-\alpha)\, Q(s_t, a_t) + \alpha\big(\underbrace{r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})}_{\text{target}}\big)$$

Q-learning

$$Q(s_t, a_t) \leftarrow (1-\alpha)\, Q(s_t, a_t) + \alpha\Big(r_{t+1} + \gamma \max_a Q(s_{t+1}, a)\Big)$$

don't need a_{t+1}

Q will improve no matter how we choose actions

(so long as we have sufficient exploration)

In practice, use ε-greedy actions to concentrate exploration in high-reward states.