# Model-Free Control

→ 기본이 되는 Monte-Carlo Control ⇒ ε-greedy Exploration

$$\pi(a|s) = \begin{cases} \epsilon/m + 1-\epsilon & \text{if } a^* = \underset{a\in A}{\operatorname{argmax}}\, Q(s,a) \\ \epsilon/m & \text{otherwise} \end{cases}$$

(m actions)

→ $1-\epsilon$ 최적 (활용)
$\epsilon$ 확률로 랜덤선택 (탐색)
↳ 모든 행동 탐색가능

< Policy evaluation : Monte-Carlo policy evaluation $Q \approx q_\pi$
Policy Improvement : ε-greedy policy Improvement

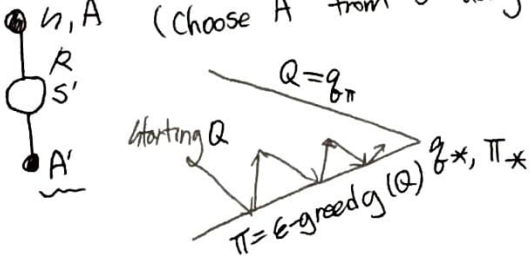
* Off-policy

학습 잘되는데, episode 마다 적용 learns only from the tails of episodes.
↳ learning will be slow

* SARSA : on-policy on TD Control V에서 Q로 적용.

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

(Choose $A'$ from $S'$ using policy derived from $Q$ (e.g. ε-greedy))




n-Step SARSA

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(q_t^{(n)} - Q(S_t, A_t))$$

Forward-view SARSA(λ)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(q_t^{\lambda} - Q(S_t, A_t))$$

단점) 끝날때까지 끝난게 아니다 → 추가) n-step off-policy TD

Backward-view SARSA(λ)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\, \delta_t E_t(S_t, A_t)$$

$$\delta_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$$

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

decay 추가

$$E_0(s,a) = 0$$

* **Qlearning** : off-policy on TD control.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big(\underbrace{R_{t+1} + \gamma \max_a Q(S_{t+1}, a)}_{\text{target}} - Q(S_t, A_t)\big)$$ (error)

< behavior : $\epsilon$-greedy
target : greedy $\quad \pi(S_{t+1}) = \underset{a'}{\operatorname{argmax}}\, Q(S_{t+1}, a')$

$\Rightarrow R_{t+1} + \max_{a'} \gamma Q(S_{t+1}, a')$.

($S_{t+1}$에서 환경과 interaction X, SARSA는 환경과 interaction O)
↳ Q-value iteration

왜 importance sampling이 되어야지 않은가? $\alpha$에서 처리됨.

★ Expected SARSA : off-policy로 작용, Q-learning보다 효과적. (expected value로 인해)

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \bar{V}_{t+n-1}(S_{t+n})$$

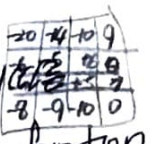
$$\left(\bar{V}_{t+n-1}(S_{t+n}) = \sum_a \pi(a|S_{t+n}) Q_{t+n-1}(S_{t+n}, a),\ t+n < T\right)$$

Q-learning의 max와 동일하게 적용

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big(R_{t+1} + \gamma E[Q(S_{t+1}, a) \mid S_{t+1}] - Q(S_t, A_t)\big)$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big(R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\big)$$

# Prediction & Control in DP




state에 대한 true value function 구하기

- Prediction: 현재 진행하는 policy에 따라 → policy evaluation 하기
- Control: policy를 optimal 하게 만듦 Q로 (policy를 바꾸는게 좋은지 나쁜지 판단 후 update) → policy improvement 하기
  - ↳ greedy

DP
- Policy Iteration: Bellman Expectation Eqn ← (Policy Evaluation, Policy Improvement → greedy)
- Value Iteration: Bellman Optimality Eqn.
  - ↓ greedy

- Policy Improvement: state value를 바탕으로, action-value function을 이용해 policy에 좋은 action 선택하여 policy update.
  - Q의 max: greedy policy improvement.

- Value Iteration: 이동가능한 state $s'$에 대해 max를 취해 greedy하게 improve.

$$v_{k+1}(s) = \max_{a \in A} \left( R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_k(s') \right) \quad \text{max}$$

$$\leftrightarrow \text{Policy evaluation}: \quad v_{k+1}(s) = \sum \pi(a|s) \left( R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_k(s') \right) \quad \text{합}$$