Reinforcement Learning

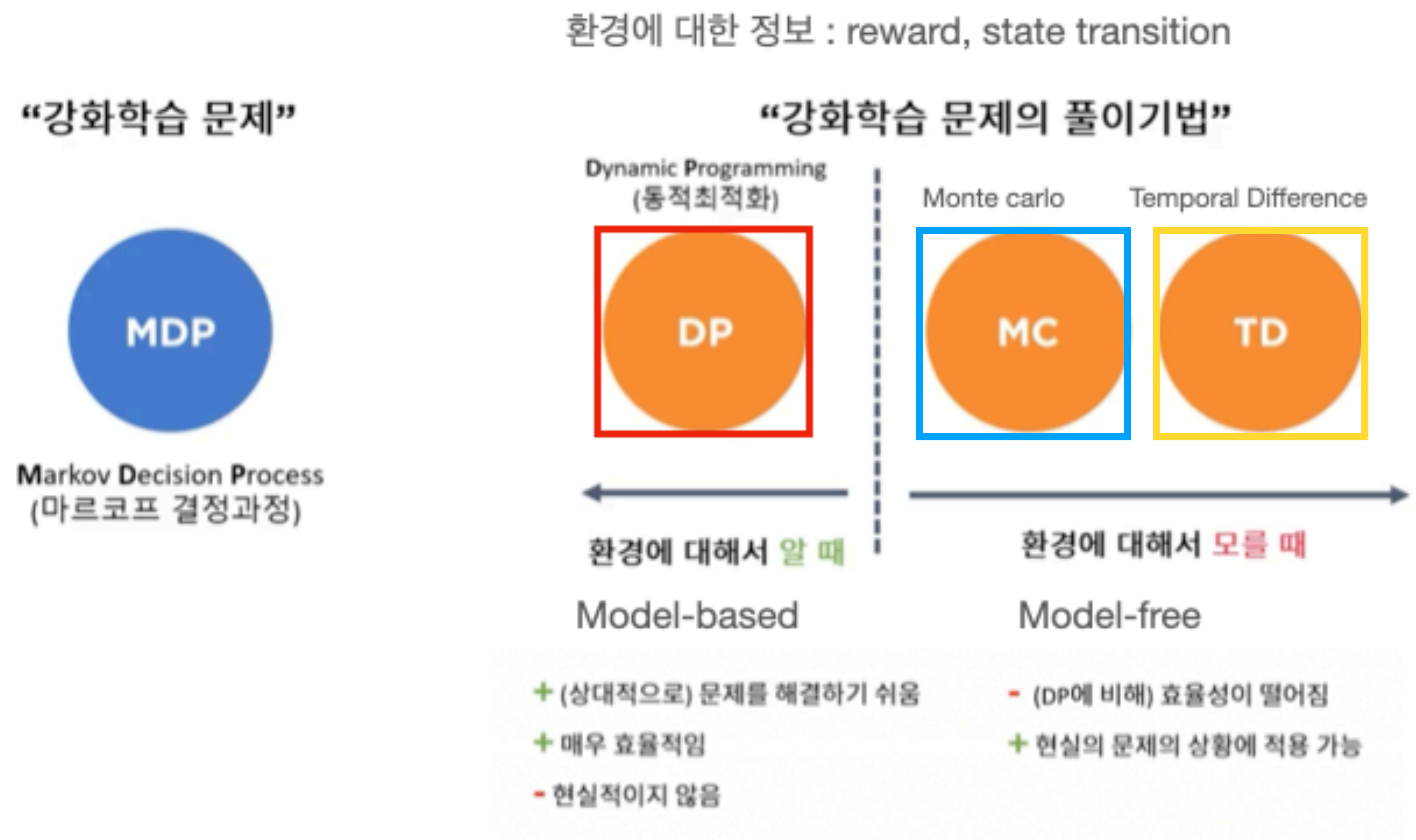
MC Control, TD SARSA

정규헌

Reinforcement Learning

강화학습 문제

- 최적 정책를 찾는 것!
- **Model based : DP, Asynchronous DP**
  - **알고 있는 값**들을 활용해 특정 공식을 수렴할 때까지 반복해 최적해를 찾음
  - 알고있는 값 - 환경모델이 필요
- **Model free : MC, TD**
  - **MC** : 데이터를 활용해서 최적해를 찾음(불편추정량)
    - 각 state와 action 사이의 관계에 대한 정보는 전혀 활용되지 않음
  - **TD** : 알고있는 일부 경험과 함께 데이터를 활용
    - 각 state와 action 사이의 관계를 활용
    - 불편추정량이 아니기 때문에 오차가 발생할 수 있음



Temporal Difference

TD vs MC

TD 기법

- Episode가 종결되지 않아도 사용이 가능
- 편향이 존재
  - 편향으로 인해 시행횟수와 무관하게 오류가 생길 수 있음
  - 추정치의 분산이 낮아 적은 시행에도 좋은 추정치를 얻을 수 있음
- Markov Discision Process (MDP) 특성을 활용
  - MDP환경이 아니면 정확도가 떨어짐

MC 기법

- Episode가 종결되어야만 사용가능
- 편향 존재하지 않음 (불편추정량)
  - 분산이 높아 많은 시행이 필요
  - 시행횟수만 충분하다면 참 가치함수를 찾을 수 있음
  - 충분한 시행횟수를 얻기 위해 더 많은 시간이 필요
- Markov Discision Process (MDP) 특성 활용하지 않음
  - MDP 환경이 아니어도 정확한 추정이 가능

Monte Carlo

Incremental MC policy evalutation

Vanila	MC policy evaluation	$V(s) \leftarrow \frac{S(s)}{N(s)}$	$S(s)$: 상태 s에 대한 G_t 들의 합 $N(s)$: 상태 s를 (처음) 방문한 횟수
	Incremental MC policy evaluation	상태 s (처음) 방문때마다, $N(s) \leftarrow N(s) + 1$ $V(s) \leftarrow V(s) + \boxed{\frac{1}{N(s)}}(G_t - V(s))$	Gt를 추산 N(s)는 counter로 구해야함
	현실에서는, $N(s)$ 을 세는 것조차 어려움	$V(s) \leftarrow V(s) + \boxed{\alpha}(G_t - V(s))$ Learning rate	Gt를 추산 N(s)를 세는것도 어려움 - s가 실수인경우, 종류가 모두 알려지지 않은 경우 등.. 적당히 작은 값으로 counter 역할

Temporal Difference

TD : n-step TD

$$V(s) \leftarrow V(s) + \alpha\left(G_t^{(n)} - V(s)\right)$$

Aka TD(0)

1-step TD : $G_t^{(1)} \stackrel{\text{def}}{=} R_{t+1} + \gamma V(S_{t+1})$: 1 스텝까지의 보상 (새로운 정보) + 가치함수 (알고 있는 정보)

2-step TD :$G_t^{(2)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$: 2 스텝까지의 보상 (새로운 정보) + 가치함수 (알고 있는 정보)

3-step TD :$G_t^{(3)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})$

...

∞-step TD :$G_t^{(\infty)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-1} R_T$: 몬테카를로와 동일

$$G_t^{(n)} \stackrel{\text{def}}{=} \boxed{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{n-1} R_{t+n}} + \gamma^n V(S_{t+n})$$

현재부터 n step까지 새로 알게된 정보

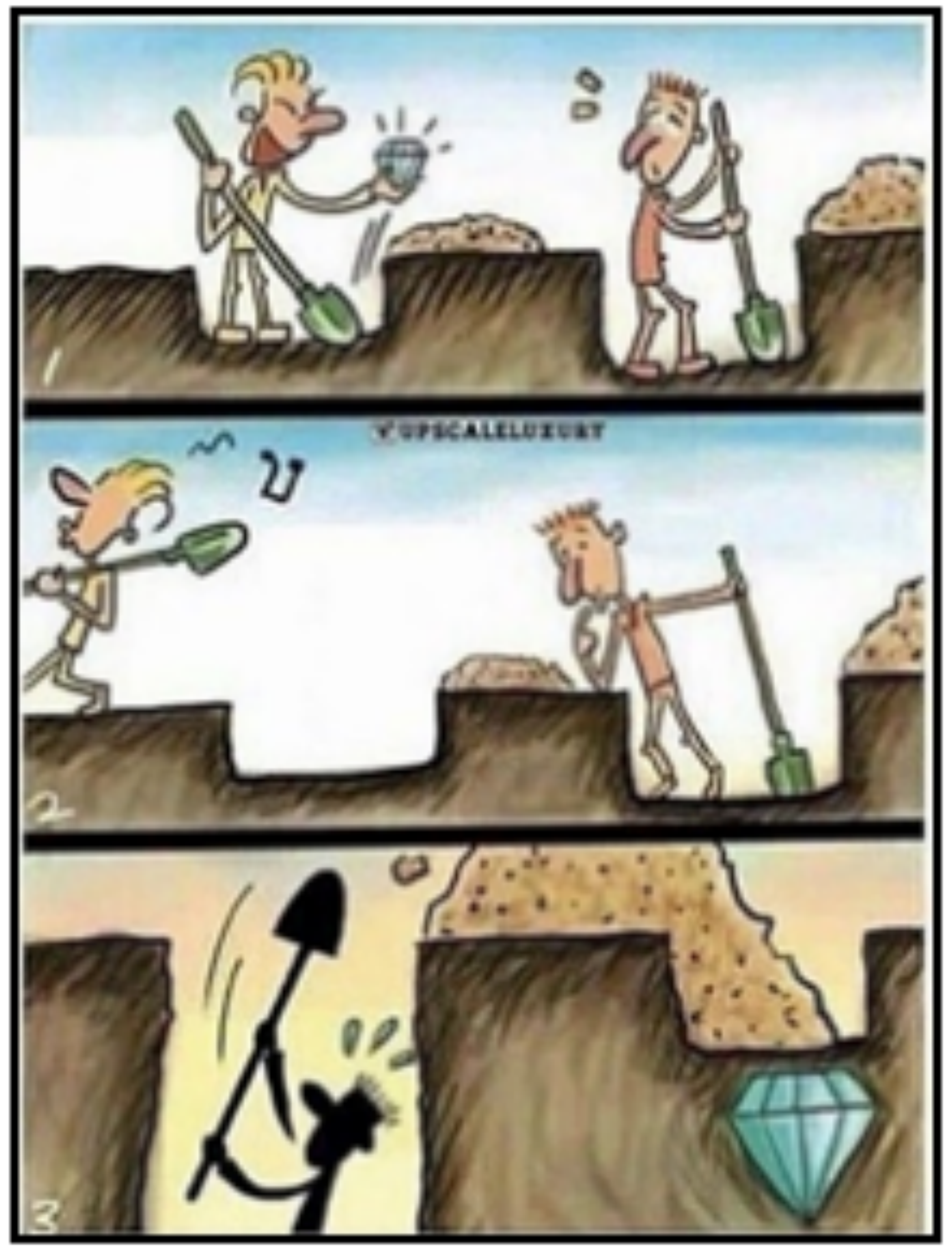
원래 알고있던 정보

Finding Policy

TD든 MC든 구한 것은 V, Q

결국 V(s)를 구했으면 Q(s,a)를 구해야하고 그 값을 통해 policy π를 구해야함

- **Greedy action** : 점수가 가장 큰 쪽으로 움직임
    - 미래를 생각하지 않고 각 단계에서 가장 최선의 선택 (Exploitation)
    - Exploration이 충분하지 않아 최상의 결과가 나오기 힘듬
- **Epsilon-greedy** : 점수가 가장 큰 쪽으로 움직이되, **epsilon 확률값 0.2** 만큼은 random
    - 부족한 Exploration을 보충
- **Decaying epsilon-greedy** : epsilon 값을 0에 가깝게 점점 줄여나감
    - Exploration + Exploitation



Greedy policy

Epsilon greedy policy

(s, a) (처음) 때마다,

$$N(s,a) \leftarrow N(s,a) + 1$$

$$Q(s,a) \leftarrow Q(s,a) + \frac{1}{N(s,a)}(G_t - Q(s,a))$$

GLIE 조건 : 모든 N(s, a)에 대해서 한 번씩 방문 할 수 있도록 epsilon을 학습진행에 따라 스케줄링

Greedy policy (탐욕적 정책) $\pi(a|s) = \begin{cases} 1, & if\ a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^{\pi}(s,a) \\ 0, & otherwise \end{cases}$

Learning rate 초기 값과 epsilon값을 어떻게 설정하느냐에 따라 성능에 영향을 크게 미침

ϵ-Greedy policy $\pi(a|s) = \begin{cases} \frac{\epsilon}{|\mathcal{A}|} + 1 - \epsilon, & if\ a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^{\pi}(s,a) \\ \epsilon/|\mathcal{A}|, & otherwise \end{cases}$ $|\mathcal{A}|$: 가능한 action 갯수

GLIE epsilon-greedy policy

$$\epsilon \leftarrow \frac{1}{k}$$

$$\pi \leftarrow \epsilon - greedy(Q)$$

MC Control

Generalized Policy Iteration

정책 반복 (Policy iteration)

입력: 임의의 정책 정책 π

출력: 개선된 정책 π'

1. 정책 평가 (PE) 를 적용해 $V^{\pi}(s)$ 계산
2. 정책 개선 (PI) 를 적용해 π' 계산

알고있는 값들을 활용, 수렴할때 까지 DP를 활용해

$$V^{\pi}(s) \xrightarrow{P,R} Q^{\pi}(s,a) \to \pi'$$

Greedy algorithm

$$\pi'(s) = \underset{a \in \mathcal{A}(s)}{\operatorname{argmax}} Q^{\pi}(s,a)$$

일반화된 정책 반복 (Generalized Policy iteration)

입력: 임의의 정책 정책 π

출력: 개선된 정책 π'

어떠한 알고리즘을 활용해서 가치함수 V,Q 계산

정책평가
1. 임의의 방식을 활용해 적용해 $V^{\pi}(s)$ 계산
   ex. MC policy evaluation

정책개선
2. 임의의 방식을 활용해 적용해 π' 계산 ($\pi' \geq \pi$ 를 만족)
   ex. greedy

Monte Carlo를 활용해 Value function은 추산했고, 정책은?

$$V^{\pi}(s) \xrightarrow{P,R} \boxed{\quad ? \quad}$$

policy improvement를 위해서는 action value function Q를 구해야함

MC Control

MC는 Q function 구하는 것이 가능

데이터 (정책 π 을 따라서 생성): $s_t \in \mathcal{S} = \{s^1, s^2, s^3\}, a_t \in \mathcal{A} = \{a^1, a^2, a^3\}$

Episode 1: $s_1, a_1, r_1; s_2, a_2, r_2; s_3, a_3, r_3; \ldots; s_T, a_T, r_T$
Episode 2: $s_1, a_1, r_1; s_2, a_2, r_2; s_3, a_3, r_3; \ldots; s_T, a_T, r_T$
Episode 3: $s_1, a_1, r_1; s_2, a_2, r_2; s_3, a_3, r_3; \ldots; s_T, a_T, r_T$

(상태, 행동, 보상) 정책 π 을 따라서 생성

First-visit method

Episode 1 : $\boxed{(s^1, a^2,} 1); (s^3, a^1, 5); (s^2, a^3, 3), (s^1, a^3, 10), (s^2, a^2, 2)$ $Q_\pi(s^1, a^2)$ = 1+5+3+10+2 = **21**
Episode 2 : $(s^3, a^1, 5); (s^2, a^2, 2); \boxed{(s^1, a^2,} 1); (s^2, a^3, 3), (s^1, a^3, 10)$ $Q_\pi(s^1, a^2)$ = 1+3+10 = **14**
Episode 3 : $(s^2, a^3, 3); \boxed{(s^1, a^2,} 1); (s^3, a^1, 5); (s^1, a^3, 10), (s^2, a^2, 2)$ $Q_\pi(s^1, a^2)$ = 1+5+10+2 = **18**

$Q_\pi(s^1, a^2)$ **= 모든 Episode에 대한 리턴의 산술평균 = (21+14+18) / 3 = 17.66**

위 방식 외에도 Every-visit, Incremental MC도 가능

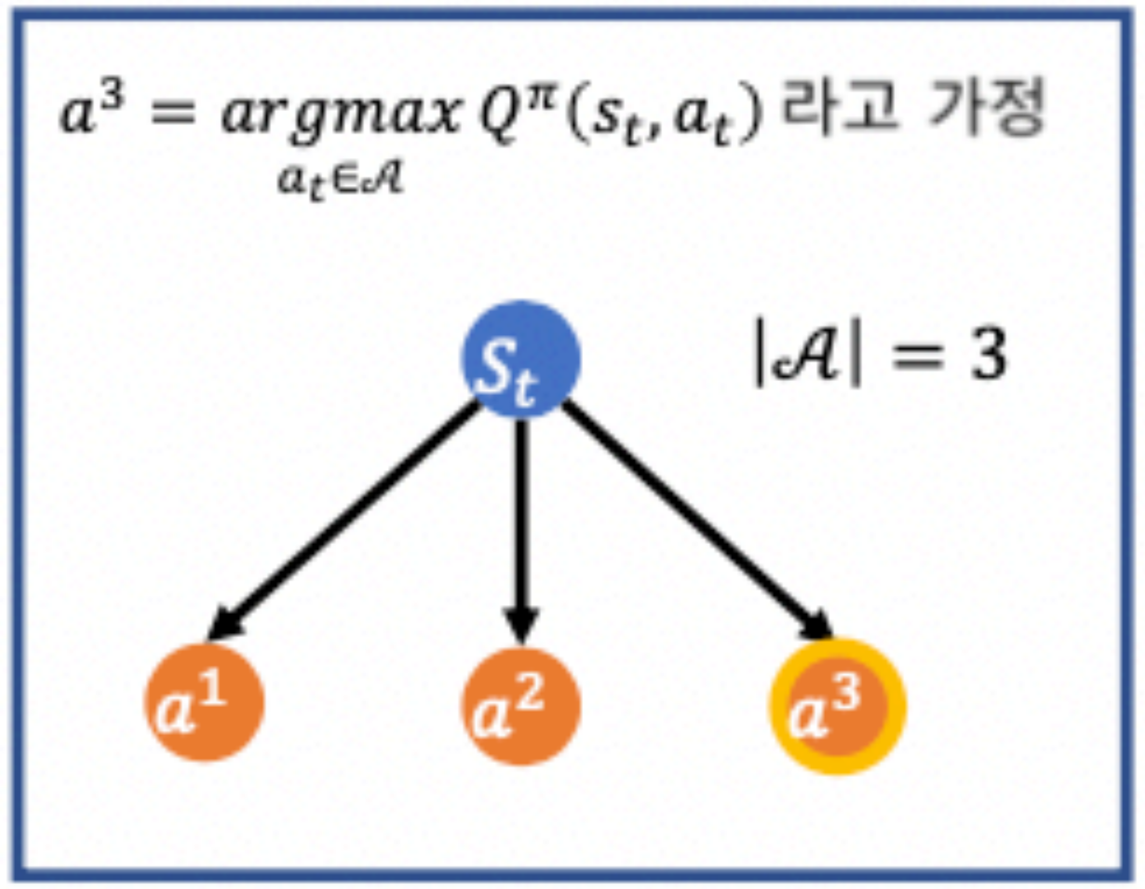
Q를 구했으면 정책에 맞는 π 구하면 됨!

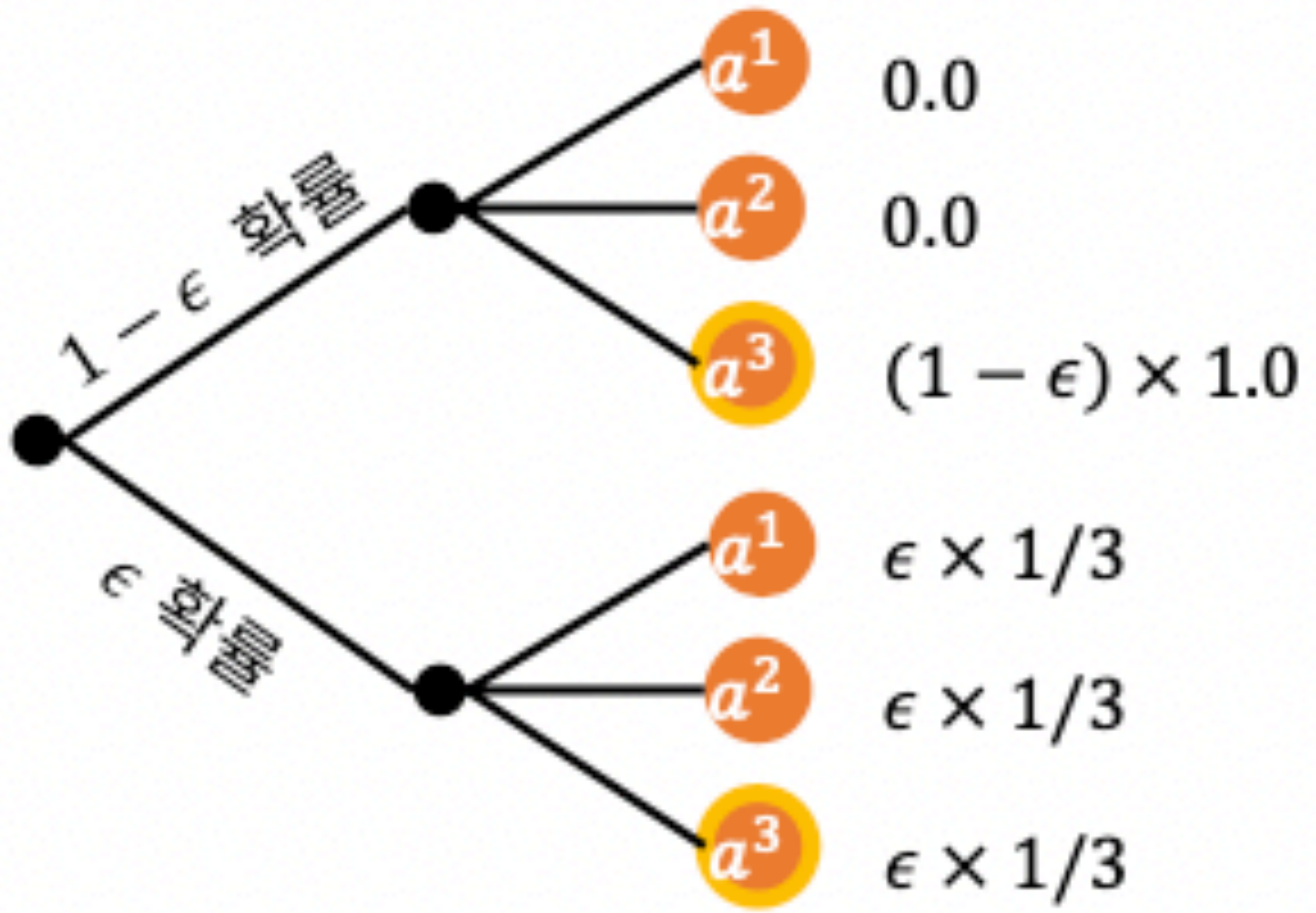
MC Control

e-greedy policy

ϵ-Greedy policy $\quad \pi(a|s) = \begin{cases} \dfrac{\epsilon}{|\mathcal{A}|} + 1 - \epsilon, & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \boxed{Q^{\pi}(s,a)} \\ \epsilon/|\mathcal{A}|, & otherwise \end{cases} \quad |\mathcal{A}|$: 가능한 action 갯수

- $1-\epsilon$ 의 확률로 "가장 좋은" 행동을 선택.
- ϵ 의 확률로 모든 가능한 행동 중 하나를 임의로 선택.





$$\pi(a|s_t) = \begin{cases} \epsilon/3, & a = a^1 \\ \epsilon/3, & a = a^2 \\ (1-\epsilon) + \epsilon/3, & a = a^3 \end{cases}$$

MC Control

MC policy evaluation + E-greedy

Monte-Carlo 정책 평가

Incremental MC policy evaluation

(s,a) (처음) 때마다,

$N(s,a) \leftarrow N(s,a) + 1$

$$Q(s,a) \leftarrow Q(s,a) + \frac{1}{N(s,a)}(G_t - Q(s,a))$$

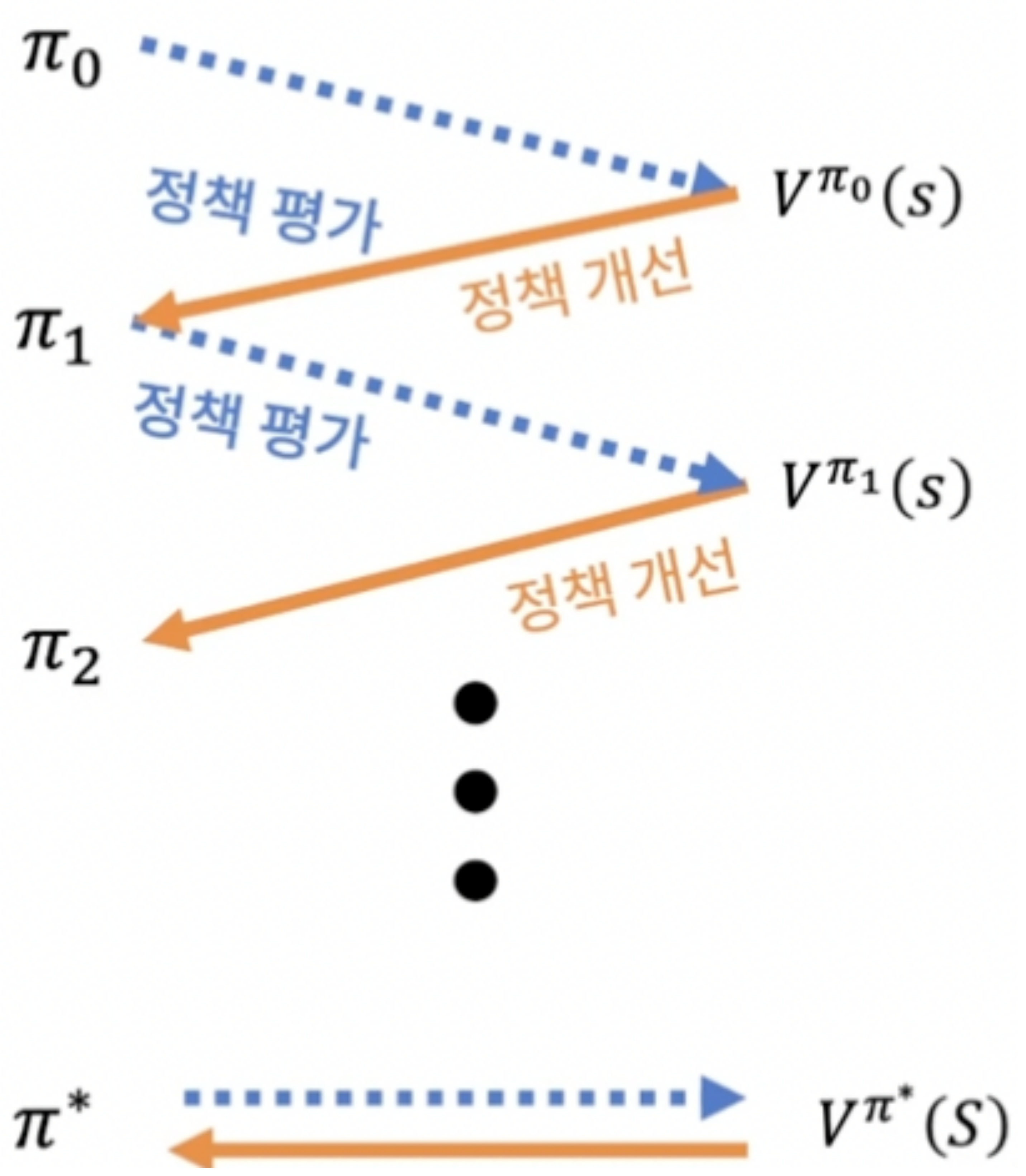
(현실에서는, $N(s)$ 을 세는 것조차 어려움)

$$Q(s,a) \leftarrow Q(s,a) + \alpha(G_t - Q(s,a))$$

GLIE ϵ-greedy 정책개선

$\epsilon \leftarrow \frac{1}{k}$ k가 증가할수록 epsilon 감소

$\pi \leftarrow \epsilon - greedy(Q)$



TD Control

TD에서는 V만 구했는데.. Q는?

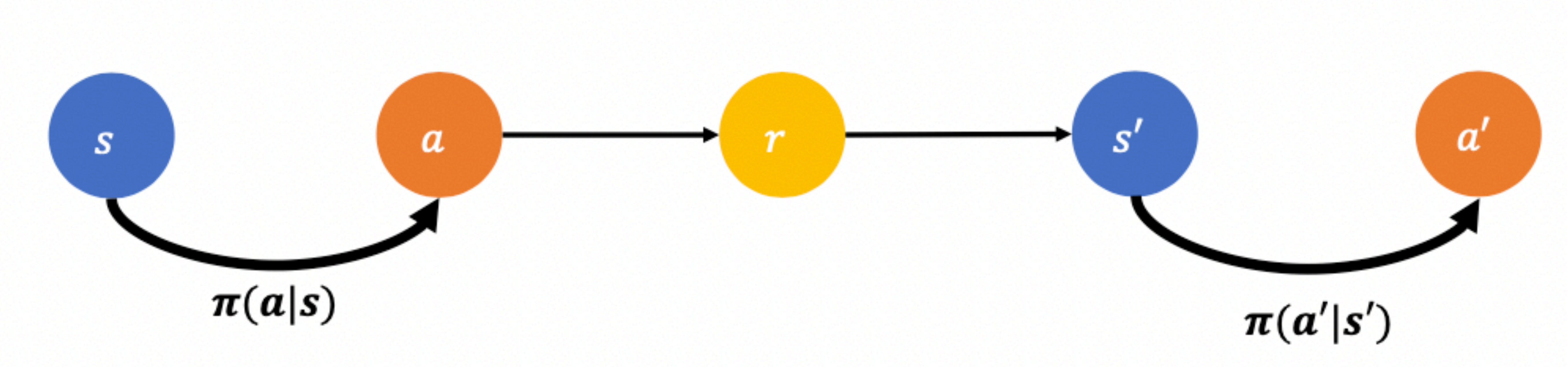
Temporal Difference

$$V(s) \leftarrow V(s) + \alpha\left(G_t - V(s)\right)$$

TD(0) 의 경우: $G_t \stackrel{\text{def}}{=} R_{t+1} + \gamma V(S_{t+1})$

TD Control

SARSA : TD를 활용한 Q 추산 - TD(0)



SARSA update:

$G_t \stackrel{\text{def}}{=} r + \gamma Q(s', a')$

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

현재 Evaluation 하는 정책 π를 따라서 a' 이 결정.

$$a' = \pi(s')$$

TD Control

SARSA : TD를 활용한 Q 추산

SARSA

초기화 $Q(s,a) \leftarrow 0$ 모든 $(s,a) \in \mathcal{S} \times \mathcal{A}$

반복 (에피소드 1, ...,):

초기 상태 s 관찰

$Q(s,a)$를 활용해서 a 결정 (ex. ϵ greedy 정책)

반복:

a 를 환경에 가한 후, r과 s' 관측.

s' 에서 $Q(s',a)$ 를 활용해 a' 결정 (ex. ϵ greedy 정책)

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

$s \leftarrow s'$; $a \leftarrow a'$

까지 s 는 종결상태

까지 $Q(s,a)$ 수렴.

TD Control

n-step TD

$$V(s) \leftarrow V(s) + \alpha\left(G_t^{(n)} - V(s)\right)$$

1-step TD : $G_t^{(1)} \stackrel{\text{def}}{=} R_{t+1} + \gamma V(S_{t+1})$: **1** 스텝까지의 보상 (새로운 정보) + 가치함수 (알고 있는 정보)

2-step TD :$G_t^{(2)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$: **2** 스텝까지의 보상 (새로운 정보) + 가치함수 (알고 있는 정보)

3-step TD :$G_t^{(3)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})$

…

∞-step TD :$G_t^{(\infty)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-1} R_T$: 몬테카를로와 동일

$$G_t^{(n)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

TD Control

n-step SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(q_t^{(n)} - Q(s_t, a_t)\right)$$

1-step TD : $q_t^{(1)} \stackrel{\text{def}}{=} R_{t+1} + \gamma Q(S_{t+1})$: **1** 스텝까지의 보상 (새로운 정보) + 가치함수 (알고 있는 정보)

2-step TD :$q_t^{(2)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 Q(S_{t+2})$: **2** 스텝까지의 보상 (새로운 정보) + 가치함수 (알고 있는 정보)

3-step TD :$q_t^{(3)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 Q(S_{t+3})$

...

∞-step TD :$q_t^{(\infty)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-1} R_T$: 몬테카를로와 동일

Q(S) : S시점에서 주어진 정책으로 action을 고른 후의 Q 값

$$q_t^{(n)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n Q(S_{t+n})$$

TD Control

SARSA (λ)

SARSA update: $G_t \stackrel{\text{def}}{=} r + \gamma Q(s', a')$

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

현재 Evaluation 하는 정책 π를 따라서 a' 이 결정.
$a' = \pi(s')$

$$q_t^{(n)} \stackrel{\text{def}}{=} R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ \dots + \gamma^{n-1} R_{t+n} + \gamma^n Q(S_{t+n})$$

$$q_t^{\lambda} \stackrel{\text{def}}{=} (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} q_t^{(n)}$$

$(0 \leq \lambda \leq 1)$

SARSA (λ) update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(q_t^{\lambda} - Q(s_t, a_t)\right)$$

TD Control

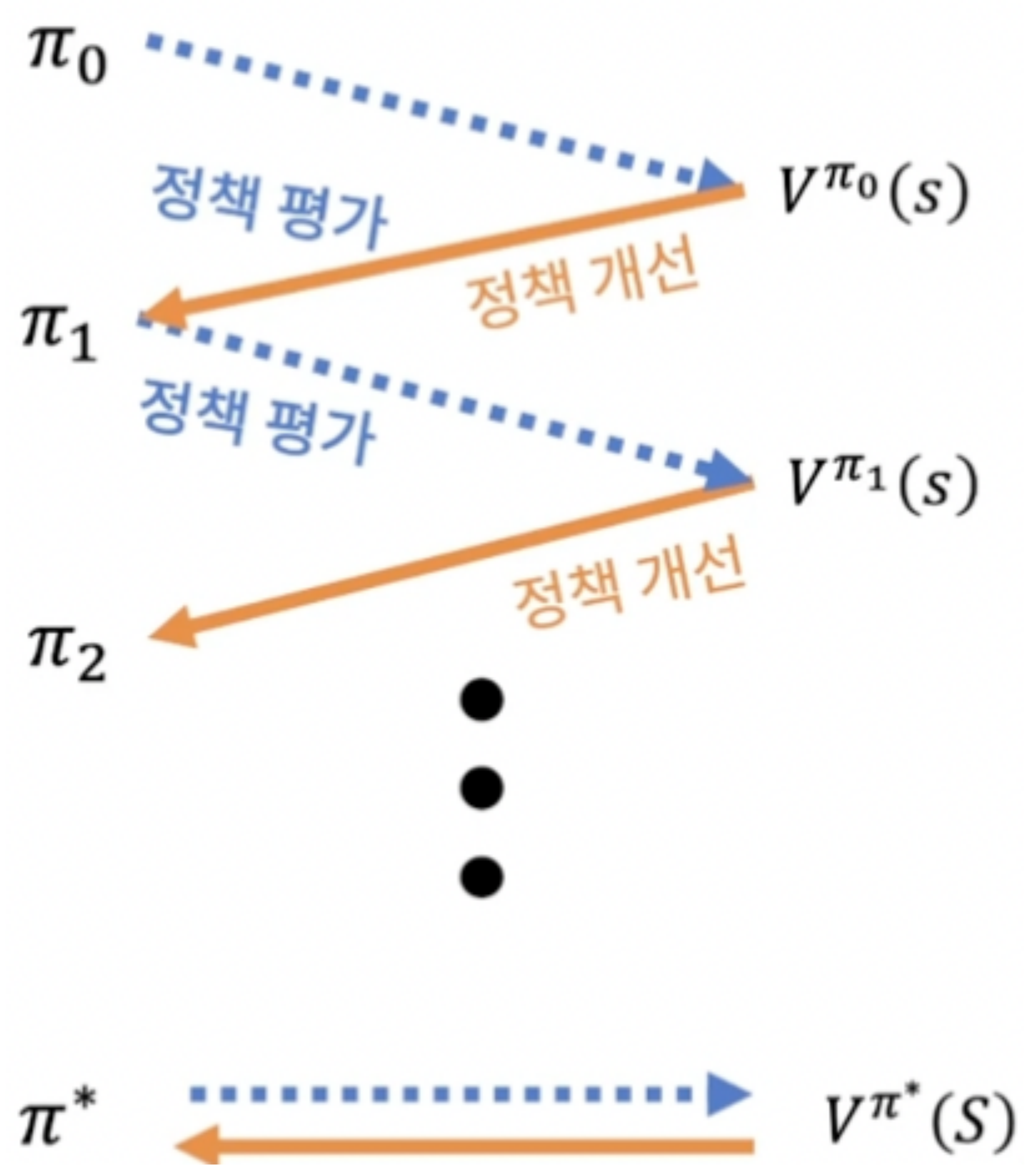
SARSA : TD를 활용한 Q 추산

정책평가

SARSA 를 활용해 $Q^{\pi}(s,a)$ 추산

정책개선

ϵ-탐욕적 정책 개선



감사합니다.