

# Chapter 1.1

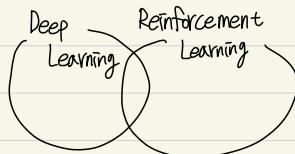
- Deep Mind : Alpha - Go, ...

RL : Action들의 연속 찾을 수 있나?

- Action → Action → ...

Goal : Maximize Reward.

활실한 목표 있어?



선택할 수 있는 방향 (동서남북) 범  
Reward 적기!

ex) 원쪽에 맛집 있을 때  
(R=1)

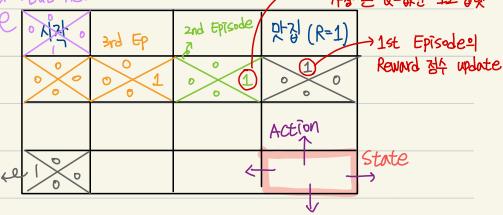
## 1.2 Q-Learning 기초편

- RL : 맛집 찾기 예시!

ex) Q-Learning

4th Episode  
But Action 고르자! ⇒ Q-Greedy.

랜덤하게 이 state에서 다른쪽 고름  
이 많은 원래 0인에 다른 행동의  
가장 큰 Q-값인 1로 업데이트



탐색적인

- Greedy Action

: 이동한 것에 대해 점수를 매기고, 가장 큰 값으로 이동.

단. 1st EP는 절수 존재하지 않아 랜덤하게 움직임.

-  $\epsilon$  - Greedy 진행

: Exploration을 하기 위해  $\epsilon$  - Greedy 진행 ( $0 < \epsilon < 1$ )

⇒  $\epsilon$  확률 만큼은 Greedy 하지 않고 랜덤하게 움직임.

$\epsilon$  이 너무 크면 학습이 진행되지 X

- Exploration & Exploitation

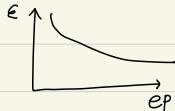
[Exploration : 더 좋은 길 탐험]  $\Rightarrow$  trade-off 관계  
[Exploitation : Q 값 이용한 활용]

• Exploration의 강점. ↗ 알기 있는 맛집 R=10

① 새로운 Path      ② 새로운 맛집 찾기

- (Decaying)  $\epsilon$  - Greedy

: 탐험 비율 줄이기 ex) 0.9  $\xrightarrow{p} 0$



시작 State 입장에서 보면  
Path 1과 Path 2의  
델 좋고/더 좋고 차이 X

$m_1$ : Path 1     $m_2$ : Path 2

- Discount factor

$\gamma_{(\text{Gamma})}$  : 0 ~ 1 사이의 값.

새로운 상태 Update 할 때, ( $\text{가장 큰 수} \times r$ )로 update

○ 사각형 1 <sup>2</sup>	△ 삼각형 2 <sup>2</sup>	○ 사각형 3 <sup>2</sup>	△ 삼각형 4 <sup>2</sup>	○ 사각형 5 <sup>2</sup>	△ 삼각형 6 <sup>2</sup>	○ 사각형 7 <sup>2</sup>	△ 삼각형 8 <sup>2</sup>	○ 사각형 9 <sup>2</sup>	△ 삼각형 10 <sup>2</sup>
○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △
○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △	○ ○ ○ ○ ○ ○ ○ ○ ○ ○	△ △ △ △ △ △ △ △ △ △

- 시작점에서 보면,  $r^2$ 으로 고르게 됨. (Greedy하게 고를 때)

$r \approx 1$  : 미래에 받을 reward의 집중 (Path 1과 Path 2 차이↑)

$\gamma \approx 0$  : 미래 reward 보다 현재 reward에 중실

→  $r=0.1$  이면 시작점에서 리워드  $0.1^2$  만큼 맛집 바로

## • 7의 장점

## ① 효율적 Path 찾기

② 현재 vs 미래 reward 중요도 조절 가능.

## - Q - Update

$$\Rightarrow Q(s_t, a_t) \xleftarrow[\text{State}]{\text{Action}}^{\text{update}} (-\alpha) Q(s_t, a_t) + \alpha \cdot \underbrace{(R_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}))}_{\begin{array}{l} \text{다음 } s_t \text{에서} \\ \text{가장 큰 Action } a_{t+1} \text{에 대한 } Q \end{array}}$$

Let,  $\alpha = 1$

① State 가 맛집 왼쪽 일 때:  $Q_{(S_t, a_t)} \leftarrow 0 + R_t$

② State가 시작 오른쪽 일 때 :  $Q(s_t, a_t) \leftarrow 0 + 1 \times (0 + 1 \cdot 1)$

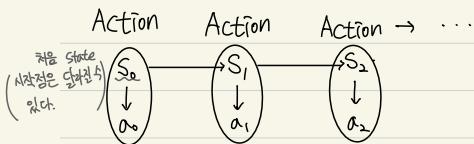
맛집 안내니까 Rt =

$$\therefore Q(s_t, a_t) \leftarrow (1-\alpha) \cdot Q(s_t, a_t) + \alpha \left( R_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \right)$$

~~$\alpha$ 가 작으면  
이부분은 커짐~~       $\alpha$ 가 클 때 이 부분 커짐

⇒  $\alpha$ : 새로운 것을 얼마나 많이 받아들이나?

## Chapter 2.1 Markov Decision Process



MDP의 중요한 성질

①  $P(a_1 | s_0, a_0, s_1)$

:  $s_0, a_0, s_1$ 이 주어졌을 때  $a_1$ 의 확률.

(State, Action은 모두 랜덤함.)

( $\Leftrightarrow$  distribution을 가지고 있고, 모두 Random Variable임)

$\rightarrow P(a_1 | s_0, a_0, s_1)$ 이 연속형 이면 pdf, 이산형이면 pmf.

$$\Rightarrow P(a_1 | s_0, a_0, s_1) = P(a_1 | \underbrace{s_1}_{\rightarrow s_1 \text{만 보면}}, a_0, s_1)$$

$\Rightarrow$  time 1에서 어떤 Action을 할 지에 대한 확률 분포

$\Leftrightarrow$  Policy : 어떤 행동을 할 까에 대한 정책.

②  $P(s_2 | s_0, a_0, s_1, a_1) = P(s_2 | s_1, a_1)$

$\Rightarrow$  transition (이동, 전이), Probability

Goal : Max Reward

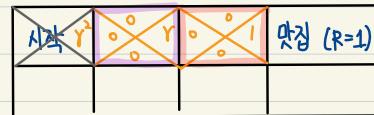
= Max Return

= Max Expected Return

at 할 때의 리워드

$$\text{Return } G_t \equiv R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots$$

Discounted Reward의 Sum



State에서 Reward = 1.

State에서 Reward =  $0 + \gamma \circ$  2번 이동했을 때  
1번 이동했을 때  
리워드 X

이 때, Action과 State 모두 Random Variable

$\rightarrow$  Reward도 Random Variable이 될

$\rightarrow$  Reward는 사실 Expected Reward!

즉, Reinforcement Learning은

어떤 distribution of Expected Return을 Maximize 할 거야?

$$P(a_t | s_t)$$

## Chapter 2.2 가치 상태 함수 (V) & 행동 가치 함수 (Q) & Optimal policy 개념.

Goal : Max Expected Return

이 때, Expected Return을 잘 표현하는 2가지 방법 존재  
지금부터 기대되는 (Expected) Return, 즉 현재 상태에 대한 가치를  
나타내는 상태 가치 함수 (State Value function).

지금 행동 (Action)으로부터 기대되는 (Expected) Return.

즉, 행동에 대한 가치를 나타내는 행동 가치 함수 (Action Value function).

$$\text{Return } G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots$$

$$* \text{ 평균값} : E[f(x)] = \int f(x) \cdot P(x) dx$$

### ① State Value function

: 지금부터의 Expected Return  $\rightarrow$   $a, s$  모두 random Variables.  
 $V(s_t) \equiv \int_{a_t: a_{\infty}} G_t \cdot P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) \cdot d a_{t+1} \dots d a_{\infty}$

→ 지금  $s_t$ 에서 할 수 있는 Action, State를 다 해보고,

그 Reward의 평균값이 State Value function

### ④ Optimal Policy

: State Value function을 Max 하는 것.

⇒ 과거는 주어진 것, 지금부터 기대되는 return을 Maximize하자!

### ② Action Value function

: 지금 행동으로부터 기대되는 Return

$Q(s_t, a_t) \equiv \int_{s_{t+1}: a_{\infty}} G_t \cdot P(s_{t+1}, a_{t+1}, s_{t+2}, a_{t+2}, \dots | s_t, a_t) ds_{t+1} \dots d a_{\infty}$

$s_t, a_t$  모두 주어짐.

### ③ Optimal Policy

: State Value function을 Max 하는

$P(a_t | s_t), P(a_{t+1} | s_{t+1}), \dots, P(a_{\infty} | s_{\infty})$ 가 optimal Policy.

f

c). 베이즈 정리를 통해  $P(a_t, s_{t+1}, a_{t+1}, \dots | s_t)$ 에서

$P(a_t | s_t), P(a_{t+1} | s_{t+1}), \dots, P(a_{\infty} | s_{\infty})$ 를 구할 수 o

⇒  $P(x, y) = P(x|y) \cdot P(y)$  이용!

$$P(a_t, s_{t+1}, a_{t+1}, \dots | s_t)$$

$$= P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) \cdot P(a_t | s_t)$$

엄밀히 말하면,

$$\text{베이즈 정리 } P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \text{ 아님.}$$

But, '표본공간의 변화'의 측면을 강조하기 위해

이 용어를 쓴 것 아닐까...?

## Chapter 2.3 벨만 방정식 (Bellman Equation)

### Bellman Equation

- Return  $G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots$
- $V(s_t) \equiv \int G_t \cdot P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) d_{a_t: a_\infty}$
- $Q(s_t, a_t) \equiv \int G_t \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) d_{s_{t+1}: a_\infty}$

⇒ 여기서,  $V(s_t)$ 를  $V(s_{t+1})$ 로 표현하기,

$Q(s_t, a_{t+1})$ 를  $Q(s_{t+1}, a_{t+1})$ 로 표현하는 방법

### ① $V(s_t)$ 를 $Q(s_t, a_t)$ 로 표현하기

$$P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) = P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) \cdot P(a_t | s_t) 이용$$

$$\begin{aligned} V(s_t) &\equiv \int G_t \cdot P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) d_{a_t: a_\infty} \\ &= \int G_t \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) \cdot P(a_t | s_t) d_{a_t: a_\infty} \\ &= \int_{a_t} \int_{s_{t+1}: a_\infty} G_t \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) d_{s_{t+1}: a_\infty} \cdot P(a_t | s_t) d_{a_t} \\ &= \int_{a_t} Q(s_t, a_t) \cdot P(a_t | s_t) d_{a_t} = Q(s_t, a_t) \end{aligned}$$

→  $s_t$ 에서의 모든 Action들에 대한 평균 리워드



상태  $s_t$ 에서 모든 Action에 따른 평균 리워드  
 $\Leftrightarrow a, b, c, d$ 의 평균.

### (장점) ① $V(s_t)$ 는 전부 할 변수 $a_t$ ( $a_t \sim a_\infty$ )

But, 별한식은 충분할 변수는  $a_t$  하나만 남음.

### ② $V(s_t)$ 를 $V(s_{t+1})$ 로 표현하기

$$(1) P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) = P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) \cdot P(a_t | s_t)$$

$$\therefore MDP (Markov Decision Process) \quad P(a_t, \dots | s_t, a_t, s_{t+1}) \cdot P(a_t, s_{t+1} | s_t)$$

$$\begin{aligned} (2) G_t &\equiv R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots \\ &= R_t + \gamma \underbrace{(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots)}_{G_{t+1}} \\ &= R_t + \gamma G_{t+1} \end{aligned}$$

→ (1)과 (2) 이용해  $V(s_t)$  정의

$$V(s_t) \equiv \int G_t \cdot P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) d_{a_t: a_\infty}$$

$$= \int_{a_t, s_{t+1}} \int_{a_{t+1}: a_\infty} (R_t + \gamma G_{t+1}) \cdot P(a_{t+1}, \dots | s_{t+1}) d_{a_{t+1}: a_\infty} \cdot P(a_t, s_{t+1} | s_t) d_{a_t, s_{t+1}}$$

$$\text{단. } V(s_{t+1}) = \int_{a_{t+1}: a_\infty} G_{t+1} \cdot P(a_{t+1}, \dots | s_{t+1}) d_{a_{t+1}: a_\infty}$$

$$\begin{aligned} &= \int_{a_t, s_{t+1}} (R_t + \gamma \cdot V(s_{t+1})) \cdot \underbrace{P(a_t, s_{t+1} | s_t)}_{= P(s_{t+1} | s_t, a_t) \cdot P(a_t | s_t)} d_{a_t, s_{t+1}} \\ &= P(s_{t+1} | s_t, a_t) \cdot P(a_t | s_t) \end{aligned}$$

transition / Probability  
 → 환경에서 주어지는 것.

Policy라고 함  
 → 찾고 싶은 것.

∴ ① or ② 식을 통해  $V(s_t)$ 를 최대화 하기 위해

그 식 내부에 있는  $P(a_t | s_t)$ 의 최대를 찾는 게

Optimal policy.

- Return  $G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots$

-  $V(s_t) \equiv \int G_t \cdot P(a_t, s_{t+1}, a_{t+1}, \dots | s_t) d a_t : a_\infty$

-  $Q(s_t, a_t) \equiv \int G_t \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) d s_{t+1} : a_\infty$

①  $Q(s_t, a_t) \stackrel{?}{=} V(s_t)$ 로 표현하기

$$P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) = P(a_{t+1}, \dots | s_t, a_t, s_{t+1}) \cdot P(s_{t+1} | s_t, a_t)$$

$$\text{MDP. } \hookrightarrow = P(a_{t+1}, \dots | s_{t+1}) \cdot P(s_{t+1} | s_t, a_t) \text{ 이다}$$

$$Q(s_t, a_t) \equiv \int G_t \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) d s_{t+1} : a_\infty$$

$$= \int (R_t + \gamma G_{t+1}) \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) d s_{t+1} : a_\infty$$

$$= \int_{s_{t+1}} \int_{a_{t+1}: a_\infty} (R_t + \gamma G_{t+1}) \cdot P(a_{t+1}, \dots | s_{t+1}) d a_{t+1} : a_\infty \cdot P(s_{t+1} | s_t, a_t) \cdot d s_{t+1}$$

$$\hookrightarrow V(s_{t+1}) = \int_{a_{t+1}: a_\infty} G_{t+1} \cdot P(s_{t+1}, a_{t+1}, \dots | s_{t+1}) d a_{t+1} : a_\infty$$

$$= \int_{s_{t+1}} (R_t + \gamma V(s_{t+1})) \cdot P(s_{t+1} | s_t, a_t) d s_{t+1}$$

②  $Q(s_t, a_t) \stackrel{?}{=} Q(s_{t+1}, a_{t+1})$ 로 표현하기.

$$P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) = P(a_{t+1}, \dots | s_t, a_t, s_{t+1}) \cdot P(s_{t+1} | s_t, a_t)$$

$$\text{MDP. } \hookrightarrow = P(s_{t+2}, \dots, a_\infty | s_t, a_t, s_{t+1}) \cdot P(s_{t+1}, a_{t+1} | s_t, a_t)$$

$$= P(s_{t+2}, \dots, a_\infty | s_{t+1}, a_{t+1}) \cdot P(s_{t+1}, a_{t+1} | s_t, a_t) \text{ 이다.}$$

$$Q(s_t, a_t) \equiv \int G_t \cdot P(s_{t+1}, a_{t+1}, \dots | s_t, a_t) d s_{t+1} : a_\infty$$

$$= \int (R_t + \gamma G_{t+1}) \cdot P(s_{t+2}, \dots, a_\infty | s_{t+1}, a_{t+1}) d s_{t+2} : a_\infty \cdot P(s_{t+1}, a_{t+1} | s_t, a_t) d s_{t+1} : a_\infty$$

$$\hookrightarrow Q(s_{t+1}, a_{t+1}) = \int G_{t+1} \cdot P(s_{t+2}, \dots, a_\infty | s_{t+1}, a_{t+1}) d s_{t+2} : a_\infty$$

$$= \int_{s_{t+1}, a_{t+1}} (R_t + \gamma Q(s_{t+1}, a_{t+1})) \cdot \underbrace{P(s_{t+1}, a_{t+1} | s_t, a_t)}_{\text{MDP.}} d s_{t+1}, a_{t+1}$$

$$\hookrightarrow = P(a_{t+1} | s_t, a_t, s_{t+1}) \cdot P(s_{t+1} | s_t, a_t)$$

$$\hookrightarrow = \underbrace{P(a_{t+1} | s_{t+1})}_{\text{Next Policy.}} \cdot \underbrace{P(s_{t+1} | s_t, a_t)}_{\text{transition.}}$$

## Chapter 3.1 Optimal Policy (derivation)

: State Value function 을 Max 하는 Policy.

지금부터 기대되는 return : 각각 Action 생각 X.

$$V(s_t) \equiv \sum_{a_t: a_\infty} G_t \cdot P(a_t, s_{t+1}, a_{t+1}, \dots, a_\infty | s_t) \cdot d a_t: a_\infty \\ = \sum_{a_t} Q(s_t, a_t) \cdot P(a_t | s_t) da_t$$

↓

Optimal Policy

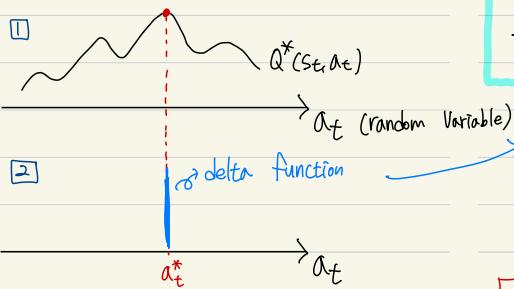
$$: \arg \max_{P(a_t | s_t)} \sum_{a_t} Q(s_t, a_t) \cdot P(a_t | s_t) da_t$$

→ 미래에 대한 optimal은 주어짐  $\Leftrightarrow$  given  $P^*(a_{t+1} | s_{t+1}), P^*(a_{t+2} | s_{t+2}) \dots$

$P^*(a_{t+1} | s_{t+1}), \dots$  를 이용해 optimal  $Q^*(s_t, a_t)$  를 구함.

→ 현재는 모르는데, 미래에 대한 상황 주어짐.

$$\Rightarrow \arg \max_{P(a_t | s_t)} \sum_{a_t} Q^*(s_t, a_t) \cdot P(a_t | s_t) \cdot da_t$$



② density function of  $a_t$ .

: 확률적으로 가장 큰  $Q^*$ 를 만드는  $a_t$ 만 추출하는 distribution이 최고.

$$\Rightarrow a_t^* \equiv \arg \max_{a_t} Q^*(s_t, a_t)$$

$\Rightarrow$  St에서의 optimal Policy는 다음과 같이 표현됨

$$P(a_t | s_t) = \delta(a_t - a_t^*)$$

$\Leftrightarrow$  결국,  $a_t^*$  를 고르세요.

$$cf. Q(s_t, a_t) = \sum_{s_{t+1}: a_\infty} G_t \cdot P(s_{t+1}, a_{t+1}, \dots, a_\infty | s_t, a_t) ds_{t+1}: a_\infty$$

이 식에서  $P(s_{t+1}, a_{t+1}, \dots, s_t, a_t)$  을  $P(a_{t+1} | s_{t+1}), P(a_{t+2} | s_{t+2}) \dots$  有.  
 $P(s_{t+1}, a_{t+1}, \dots, s_t, a_t)$

$$= P(a_{t+1}, s_{t+2}, \dots | s_t, a_t, s_{t+1}) P(s_{t+1} | s_t, a_t)$$

$$= P(a_{t+2}, s_{t+3}, \dots | s_{t+1}, a_{t+1}, s_{t+2}) \cdot P(s_{t+2} | s_{t+1}, a_{t+1}) \cdot P(a_{t+1} | s_{t+1}) \cdot P(s_{t+1} | s_t, a_t)$$

$$= P(s_{t+1}, \dots | s_{t+2}, a_{t+2}) \cdot P(a_{t+2} | s_{t+2}) \cdot P(s_{t+2} | s_{t+1}, a_{t+1})$$

$$\cdot P(a_{t+1} | s_{t+1}) \cdot P(s_{t+1} | s_t, a_t)$$

● : transition, 환경에서 주어진 것. 어찌할 수 X.

● : 내가 Maximise 할 것, Optimal Policy.

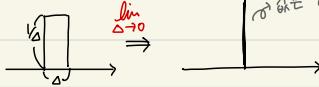
$\Rightarrow P^*(a_{t+2} | s_{t+2}), P^*(a_{t+1} | s_{t+1}) \dots$  을 넣었을 때의

$Q^*(s_t, a_t)$  가 주어짐.

cf. delta function (pdf)

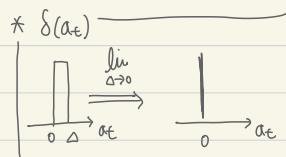
: 확률을 적분하면 1이 되는 함수이지만,

가질 수 있는 x값 한정임



즉, St에서의 optimal Policy는

$a_t^*$  가 주어졌을 때,  $Q^*$  를 Max하는  $a_t^*$  찾기



## Chapter 3.2 Monte Carlo 방법

How can we get  $Q^*$ ?

[ Monte Carlo  
Temporal difference ]

⇒ 알고리즘을 통해  $Q^*$ 에 근사시키자!

episode 진행하며  $Q^*$  update

⇒ (Decaying)  $\epsilon$ -Greedy 하게 구하기.

train 되면  $\epsilon=0$  으로 놓고 Action (Test)

### \* Monte - Carlo Method

대수의 법칙이 대표적인 Monte - Carlo Method.

⇒ 랜덤 표본을 뽑아 학수의 값을 확률적으로 계산하는 알고리즘  
복잡한 계산식일 경우, 그 값을 근사적으로 계산할 때 사용.

$$Q(S_t, a_t) \equiv \int_{S_{t+1}: a_\infty} G_t \cdot P(S_{t+1}: a_\infty | S_t, a_t) dS_{t+1}: a_\infty$$

$$\approx \frac{1}{N} \sum_{t=1}^N G_t^{(i)}$$

$$\text{단, } G_t^{(i)} = R_t^{(i)} + \gamma R_{t+1}^{(i)} + \dots, G_t^{(i)} \sim P(S_{t+1}: a_\infty | S_t, a_t)$$

행동(a)하고 그에 따른 transition 행(a') 및  
얻어지는 return의 sample들

↳  $\epsilon$ -Greedy Action 통해 선택.

### \* 대수의 법칙

$$E[X] = \int_X x \cdot P(x) dx \approx \frac{1}{N} \sum_{t=1}^N x_t$$

$P(x)$  따르는 x 뽑기

ex) 맷집 찾기 예시



→ : 지금 현재 상태  $S_t$ 에서

오른쪽으로 가는 행동 ( $a_t$ ) 시행.

→ : 맷집 나올 때 까지 모든 행동 해봄

$$G_t^{(i)} = R_t^{(i)} + \gamma R_{t+1}^{(i)} + \gamma^2 R_{t+2}^{(i)}, G_t^{(i)} \sim P(S_{t+1}: a_\infty | S_t, a_t)$$

이 때,  $P(S_{t+1}: a_\infty | S_t, a_t)$ 은  $\epsilon$ -Greedy Action

$$\rightarrow \frac{1}{N} \sum_{t=1}^N G_t^{(i)}$$

는 절차  $Q$ 값에 근사.

$\rightarrow Q$ 값을 Max 하게 하는  $P(S_{t+1}: a_\infty | S_t, a_t)$ 은  $\epsilon$ -greedy 하게 움직임

↳ Episode 지난 때마다 모든 절차 작아짐.

(Ep 1)

- 처음에 완전 랜덤하게 움직임  $\rightarrow P(S_{t+1}: a_\infty | S_t, a_t)$  낮음.

-  $P(S_{t+1}: a_\infty | S_t, a_t)$  바탕으로  $Q$  구함.

-  $Q$ 를 바탕으로  $\epsilon$ -greedy Action 시행.

-  $P(S_{t+1}: a_\infty | S_t, a_t)$  높아짐.

- 높아진  $P(S_{t+1}: a_\infty | S_t, a_t)$ 을 바탕으로 샘플  $G_t^{(i)}$  구함.

-  $G_t^{(i)}$ 를 바탕으로 더 높은  $Q$ 값 구함.

$$\Rightarrow \text{반복!!!} \Rightarrow [Q \approx Q^*]$$

$\downarrow$   
by.  $\epsilon$ -greedy.

## Chapter 3.3 Temporal difference (TD) & SARSA

### \* 1-Step TD

$$Q(S_t, A_t)$$

transition  
Policy.

$$= \int_{S_{t+1}, A_{t+1}} (R_t + \gamma \cdot Q(S_{t+1}, A_{t+1})) \cdot P(S_{t+1} | S_t, A_t) P(A_{t+1} | S_{t+1}) dS_{t+1} dA_{t+1}$$

$$\approx \frac{1}{N} \sum_{i=1}^N (R_t^{(i)} + \gamma \cdot Q(S_{t+1}^{(i)}, A_{t+1}^{(i)}))$$

random Variable

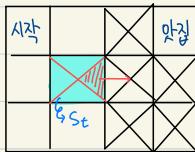
### Cf. Monte-Carlo Method

$$Q(S_t, A_t) = \int_{S_{t+1}: a_{t+1}} G_{t+1} : P(S_{t+1}: a_{t+1} | S_t, A_t) dS_{t+1} da_{t+1}$$

$$\approx \frac{1}{N} \sum_{i=1}^N G_t^{(i)}$$

단,  $G_t^{(i)} = R_t^{(i)} + \gamma R_{t+1}^{(i)} + \dots$

→  $G_t$  를 얻기 위해  $S_{t+1}$ 의 Reward가 날 때까지,  
끝까지 기약함. ( $\because G_t^{(i)} = R_t^{(i)} + \gamma R_{t+1}^{(i)} + \dots$ )



- MC 방법은  $S_t$ 에서 맛집에 갈 때까지 Sample을 얻어야 하지만,

TD 방법은  $S_{t+1}$  까지만 가는 샘플을 얻을 수 있음.  
즉, 위의 예에서,  $S_{t+1}^{(i)}$ 가 오른쪽일 때  
가능한  $A_{t+1}^{(i)} = \{위, 아래, 좌, 우\}$

$$\bar{Q}_N = \frac{1}{N} \sum_{i=1}^N (R_t^{(i)} + \gamma Q(S_{t+1}^{(i)}, A_{t+1}^{(i)}))$$

let,  $N=9$ ,  $\frac{1}{9} \sum_{i=1}^9 X_i (= \bar{Q}_9)$  추출

$$= \frac{1}{N} (\bar{Q}_{N-1} \cdot (N-1) + R_t^{(N)} + \gamma \cdot Q(S_{t+1}^{(N)}, A_{t+1}^{(N)}))$$

↑  $X_i$  를 하나씩 추출해  $N=10$  일 때  
 $\frac{1}{10} \sum_{i=1}^{10} X_i = \bar{Q}_9 + \frac{1}{10}(X_{10} - \bar{Q}_9)$

$$= \bar{Q}_{N-1} + \frac{1}{N} (R_t^{(N)} + \gamma \cdot Q(S_{t+1}^{(N)}, A_{t+1}^{(N)}) - \bar{Q}_{N-1})$$

⇒ Incremental MC update

(1-step TD) TD-target: 가장 최근의 샘플

$$\therefore \bar{Q}_N = \bar{Q}_{N-1} + \frac{1}{N} (R_t^{(N)} + \gamma \cdot Q(S_{t+1}^{(N)}, A_{t+1}^{(N)}) - \bar{Q}_{N-1})$$

$$= (1-\alpha) \bar{Q}_{N-1} + \alpha (R_t^{(N)} + \gamma \cdot Q(S_{t+1}^{(N)}, A_{t+1}^{(N)}))$$

$\frac{1}{N} = \alpha$

↳ Q-update 형태와 유사함.

### Cf. Q-update

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

## \* SARSA

- $S$  : 현재 state
- $A$  : 현재 Action
- $R$  : Reward
- $S'$  : Next State
- $A'$  : Next Action

$$\bar{Q}_N = (1-\alpha) \cdot \bar{Q}_{N-1} + \alpha \left( \underbrace{R_t^{(n)}}_{S,A} + \gamma \cdot \underbrace{Q(S_{t+1}^{(n)}, A_{t+1}^{(n)})}_{S,A} \right)$$

$$Q(S_t, A_t) = \sum_{S_{t+1}, A_{t+1}} \underbrace{(R_t + \gamma \cdot Q(S_{t+1}, A_{t+1}))}_{R} \cdot P(S_{t+1} | S_t, A_t) \cdot P(A_{t+1} | S_t, A_t)$$

→ 현재 state, Action에 대한 Q값을

Next State, Action, Reward를 이용해 샘플을 얻어 update함.

- Q를 바탕으로 Action을 정하기 때문에

Q가 업데이트되는 순간 P도 같이 업데이트

- Explore 필요해  $\epsilon$ -greedy 함.

(greedy 만 하면 탐험 못하고 충분히 다채로운 샘플 얻지 X  
 Q 업데이트 제대로 X)

→ (Decaying)  $\epsilon$ -greedy 하면서 Q 절실히 업데이트 해

greedy - policy 됨.

- MC, TD를 이용해  $Q^*$  구하면, greedy policy 최고.

## Converges to optimal

$P(A_{t+1}, S_{t+1})$  할 때, 즉 Action 할 때 (Decaying)  $\epsilon$ -greedy 하면서

점차 optimal policy로 다가가게 함

동시에 샘플 ( $R_t + \gamma \cdot Q(S_{t+1}, A_{t+1})$ )을 가지고 Q 업데이트 → optimal Q 얻어짐.

## Chapter 3.4 MC VS TD

### Review

Optimal Policy : state-value function 최대화

⇒ Action 하나 선택. 단  $Q^*$ 을 알아야 함.

How can we get  $Q^*$ ?

- [ Monte - Carlo ]
- [ Temporal difference ]

### \* Monte - Carlo Method

$$Q(s_t, a_t) \equiv \int_{s_{t+1}, a_{\infty}} G_t \cdot P(s_{t+1}: a_{\infty} | s_t, a_t) d s_{t+1}, a_{\infty}$$

$$\approx \frac{1}{N} \sum_{i=1}^N G_t^{(i)} \quad \text{단, } G_t^{(i)} = R_t^{(i)} + \gamma R_{t+1}^{(i)} + \dots$$

- 처음에  $P(s_{t+1}: a_{\infty} | s_t, a_t)$  값은 별로 좋지 않음.

샘플을 구해  $Q$  값 업데이트하고,  $\epsilon$ -greedy를 통해  $P$  값 업데이트 → 그럼  $Q$  값도 좋아짐.

-  $G_t^{(i)}$ 는 완벽히 좋은 sample

:  $P(s_{t+1}: a_{\infty} | s_t, a_t)$ 을 따르는  $G_t$ 의 Sample  $G_t^{(i)}$  추출

### (정점) Unbiased - Sample

(단점) Variance ↑ ↗  $t, t+1, t+2, \dots$  까지 Sample 진행!

(why? 여기저기 갈 곳 많아서 수렴 힘들)

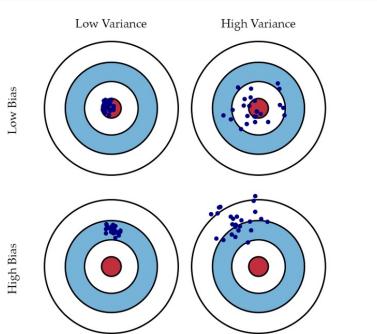


Fig. 1 Graphical illustration of bias and variance.

→ cf.  $E[X] = \int x \cdot P(x) dx \approx \frac{1}{N} \sum_{i=1}^N X_i$

: 확률 변수  $X$ 의 기댓값 구하는 방법은  $P(x)$ 를 따르는

$X$ 의 샘플  $X_i$ 를  $N$ 개 추출해 더하고  $N$ 으로 나눔

→  $X_i$ 는  $P(x)$ 를 따르는  $X$ 의 Sample!

### \* 1-Step TD

$$Q(s_t, a_t) \stackrel{\text{transition}}{=} \int_{s_{t+1}, a_{t+1}} (R_t + \gamma \cdot Q(s_{t+1}, a_{t+1})) \cdot P(s_{t+1} | s_t, a_t) \cdot P(a_{t+1} | s_t, a_t) d s_{t+1}, a_{t+1}$$

$$\approx \frac{1}{N} \sum_{i=1}^N (R_t^{(i)} + \gamma \cdot Q(s_{t+1}^{(i)}, a_{t+1}^{(i)}))$$

-  $\epsilon$ -greedy 하면서  $P(s_{t+1} | s_t, a_t) \cdot P(a_{t+1} | s_t, a_t)$  값 좋아지고,

Next  $Q$  값  $Q(s_{t+1}, a_{t+1})$ 은 Sample 통해 업데이트

→  $Q(s_{t+1}, a_{t+1})$ 도 샘플이기 때문에  $(R_t^{(i)} + \gamma \cdot Q(s_{t+1}^{(i)}, a_{t+1}^{(i)}))$ 는 완벽히 좋은 샘플은 아님.

why?  $Q(s_{t+1}, a_{t+1})$  조차도 업데이트 중. ⇒ Expected value.

↳ Expected Value가 완전히 수렴되지 않은 상태에서 샘플 가져옴

→  $Q(s_{t+1}, a_{t+1})$  값이 완벽하게 않기 때문에 bias 생김.

(정점) Variance ↓ (.. 다음 시간 ( $t+1$ )의 값에서만 sampling)

### (정점) Biased Sample