STUDY PRESENTATION

SUSC Summer 2023

강화학습 스터디 내용 소개

동아대 AI학과 김지선

STUDY PRESENTATION

01

MDP(Markov Decision Model)

Remind Notation

- 기본 용어
  - 상태: s(x), 행동: a(u)
  - 다음 상태: s'(x'), 행동: a'(u')
  - 보상함수 r(a|s)
  - 보상의 합(반환값) Gt
    - 감마(r)는 Discount Factor로 0<=r<=1의 값으로 현재에 얼마나 더 가중치 줄지를 결정하고,
  - 정책: π(u|x)
  - 상태천이 확률 밀도 함수P(x|x,u)
  - 기댓값 E[]

- 벨만 방정식
  - 행동 가치 함수 Q(s,a)
  - 상태 가치 함수 V(s)

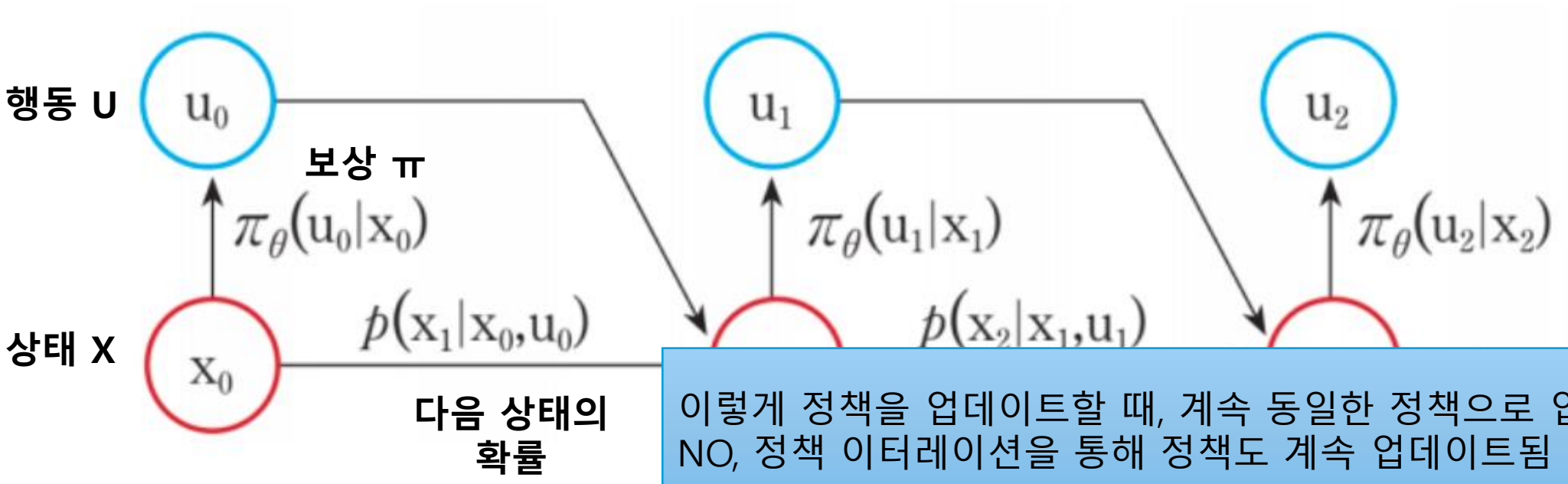
MDP(Markov Decision Model)

- 순차적 행동 결정 문제를 풀기 위해 문제를 수학적으로 정의한 것.
- 용어
  - 상태: 에이전트의 정보, 움직이는 속도와 같은 동적 요소도 상태로 표현 가능
  - 행동: 에이전트가 취할 수 있는 행동
  - 정책: 순차적 결정 문제에서 구해야할 답,모든 상태에 대해 어떤 행동을 할지 알아야함.
    - 모든 상태에 대해 에이전트가 어떤 행동을 할지 정해놓는 것.
  - 상태 변환 확률: 상태가 다른 상태로 변할 확률을 수학적으로 나타낸 것.
  - 보상: 에이전트가 학습할 수 있는 유일한 정보, 보상을 통해 행동에 대한 평가 받음.
    - r(discount factor, 할인률): 0<=r<=1, 얼마나 더 가까운 미래를 더 중요하게 가중치 줄지 결정

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

MDP(Markov Decision Process) 전개

- 어떤 정책에 의해 u가 확률적으로 선택되면, 상태천이 확률에 의해 상태변수 x1으로 이동하고 보상이 주어진다.
- **상태변수, 행동, 보상의 순서가 반복되는 것**



이렇게 정책을 업데이트할 때, 계속 동일한 정책으로 업데이트 되는가?
NO, 정책 이터레이션을 통해 정책도 계속 업데이트됨

강화학습의 목표

- 마지막 시점에 <u>가장 많은 보상을 얻을 수 있는 최적의 정책을 얻는 것</u>.
- 원하는 정책을 얻기 위해서는?
  - 정책 평가 (가치함수를 기반으로 현재 정책 평가)
  - 정책 발전 (평가 완료 후에, 최대의 보상을 얻을 수 있도록 정책 수정)
- 정책 평가 처음에는 이 방정식의 양변이 일치하지 않음

(1) 정책 포함 $$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

(2) 정책 미포함 $$v_\pi(s) = \sum_{a \in A} \pi(a|s)\left(r(s,a) + \gamma \sum_{s' \in S} P^a_{ss'} v_\pi(s')\right)$$

[그림1. 벨만 기대방정식]

⇒ 정책 평가 처음에는 가치함수가 최적의 가치함수가 아니기 때문.

강화학습 방법

- 최적의 정책(Policy)를 산출하는게 목표
- 강화학습은 공통적으로 Step1~3 반복
  - Step1) 정책을 실행해 샘플(데이터) 생성
  - Step2) 정책에 따른 모델/가치 함수를 추정
  - Step3) 정책 개선

    => 정책을 한 번 실행해서 가치를 계산해보고, 정책을 개선한다.

    => 정책은 계속 업데이트된다.

STUDY PRESENTATION

01

벨만 방정식

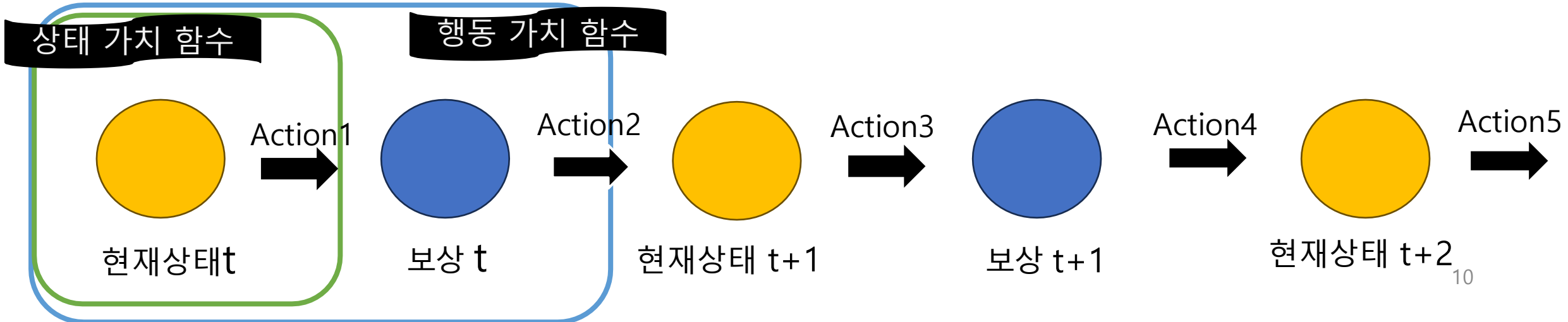
벨만 기대 방정식, 벨만 최적 방정식

벨만 기대 방정식

- 현재 상태 가치함수와 다음 상태 가치함수의 관계식이 벨만 방정식.
- 벨만 기대 방정식은 특정 정책을 따라갔을 때 기대함수 사이의 관계식
- 벨만방정식은 결국, **어떤 상태에서, 어떤 행동을 할 때, 어떤 가치가 있는지를 미리 예측할 수 있는 것이 벨만 기대 방정식**
- 가치함수
  - 특정 상태의 가치는 반환값에 대한 기댓값으로 특정 상태의 가치를 판단할 수 있음.

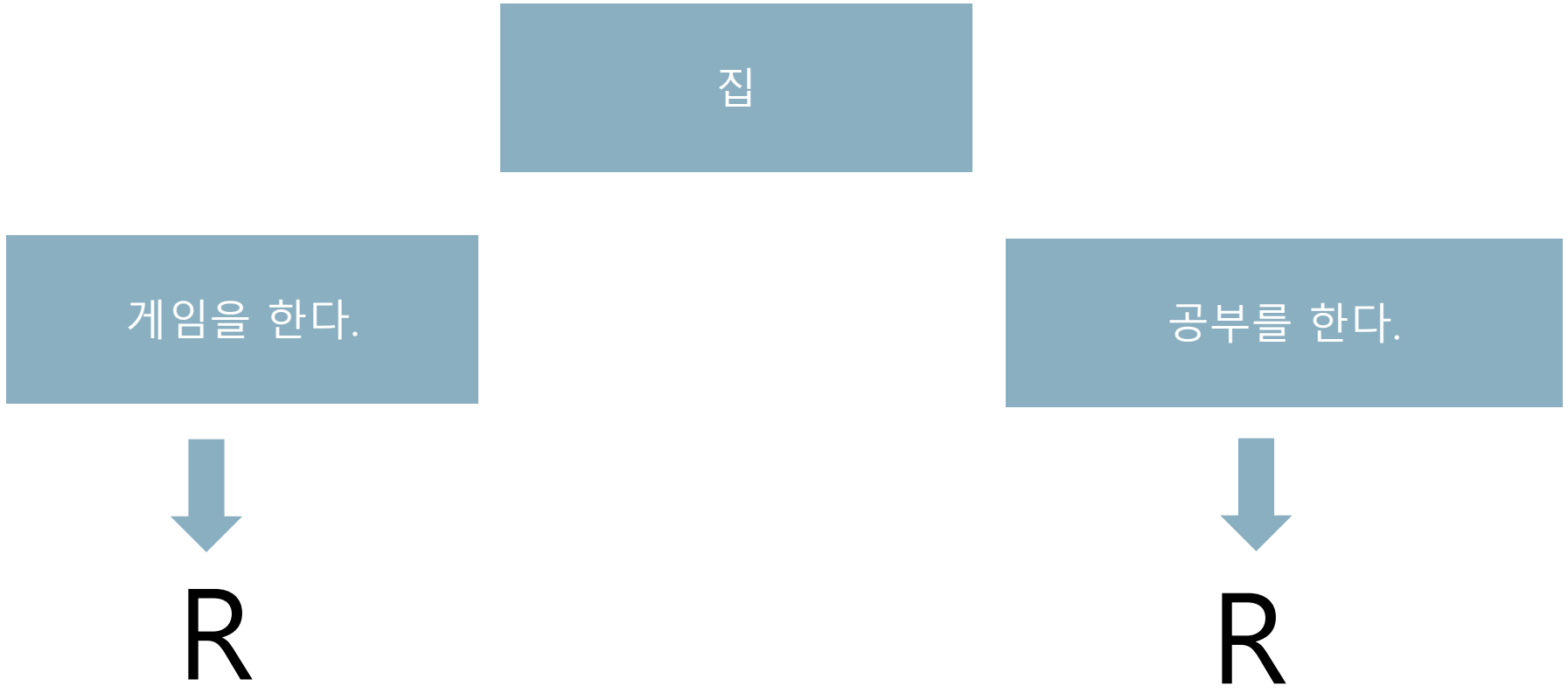
벨만 에러 방정식 활용 예시

- 결국 벨만 방정식은 연속적으로 행동을 결정해야하는 강화학습 문제에서 유용
  - 현재 시점 (t)를 기준으로, 다음 시점(t+1), 다다음 시점(t+2) 시점의 미래 보상을 계산 가능
  - 이때 행동을 하기 전에 내가 선택할 행동으로 받는 미래의 보상을 계산할 수 있음
  - 상태 가치 함수/행동 가치 함수를 통해 각 시점에서 미래 보상을 계산할 수 있는 것은 매우 유
- 벨만 방정식
  - 이때 상태 가치와 행동 가치간의 관계를 계산할 수 있어서 강화학습의 핵심 이론으로 쓰임.

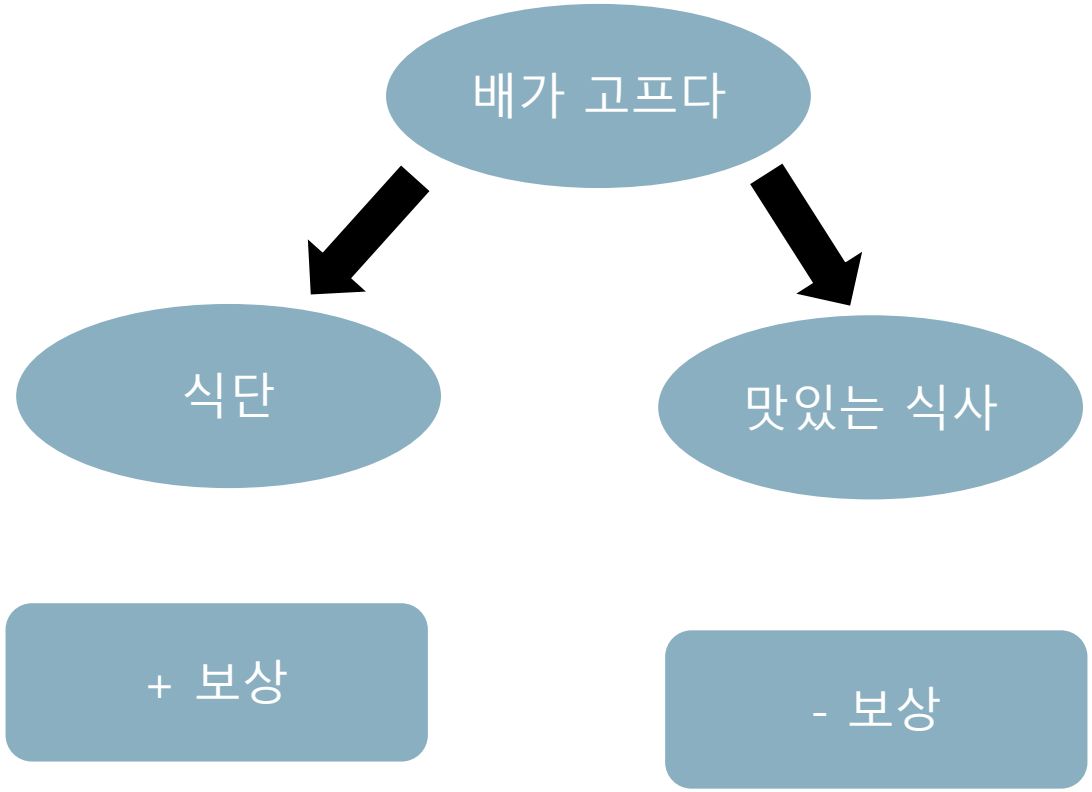


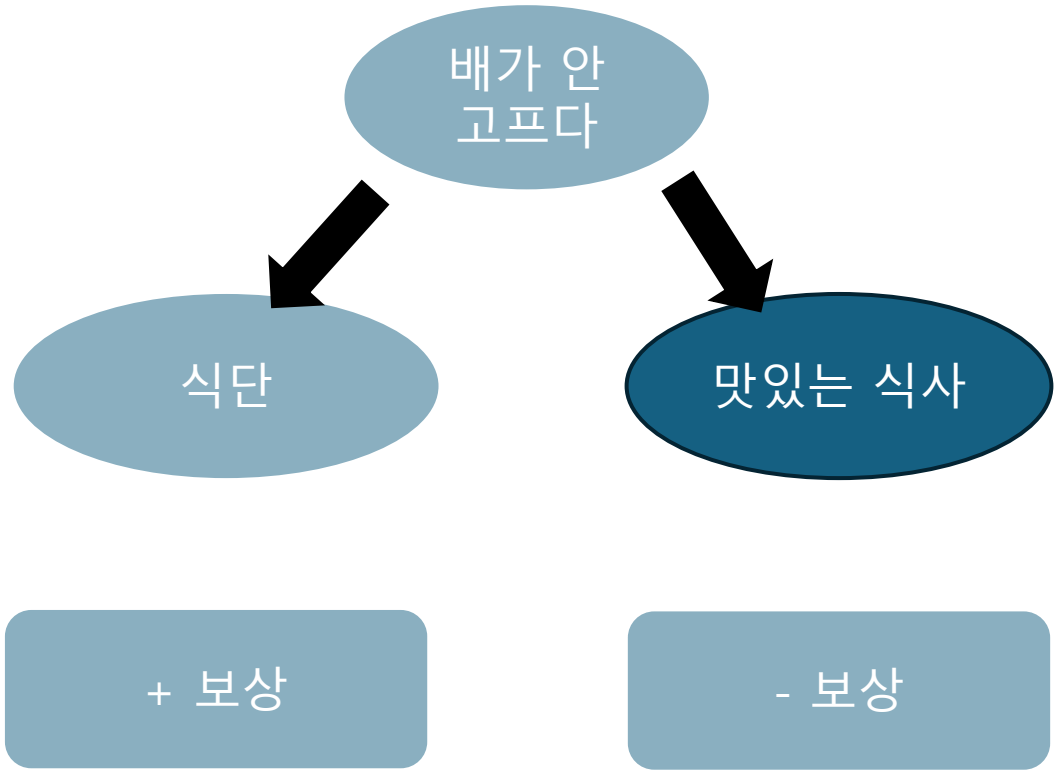
현재 시점에서 다음 행동을 했을 때 보상이란?

현실에서 여러가지 선택지가 있을 때, 기준이 있는 것이 중요하듯, 강화 학습에서도 정책(Policy)를 통해, 기준을 잘 설정해 주는 것이 매우 중요



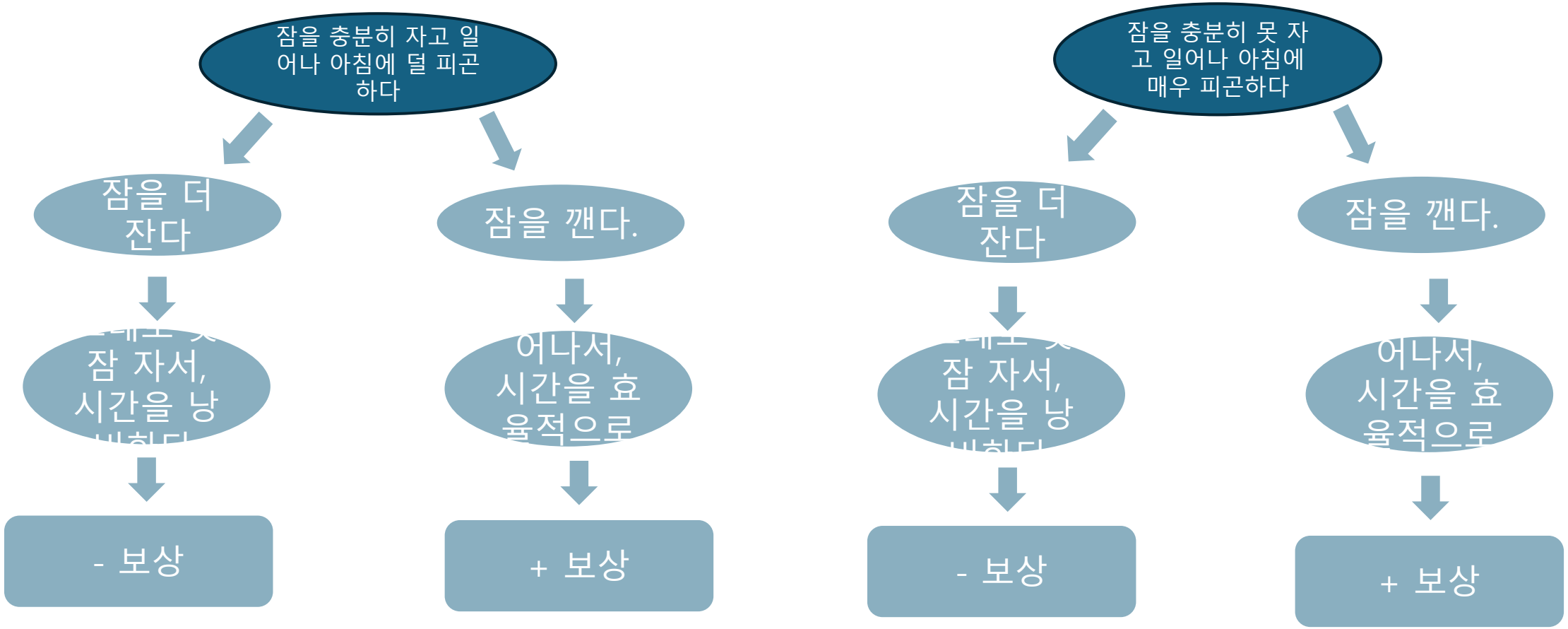
현재 상태에 따라, 행동의 보상이 바뀐다





⇒ **최적의 상태에서 최선의 선택을 할 확률이 더 높다.**
⇒ **즉, 좋은 상태(최적의 상태)를 만드는 것이 더 유리하다. (최적 상태 가치 함수)**

일상생활에서도 다 선택의 연속



이러한 실제 Agent의 보상을 최대화하기 위해, 실제 강화학습에서는 행동 기준인 정책(Policy)를 설계할 때, -reward를 주는 방향으로도 학습한다.

벨만 에러 방정식 종류

- 벨만 기대 방정식: 가장 기본적인 상태
  - 가치함수
  - 행동 가치함수
- 벨만 최적 방정식:
  - 최적 상태 가치함수
  - 최적 행동 가치함수를 구하는 것.

벨만 방정식 종류

- 벨만 기대 방정식
  - **상태 가치 함수:** 그 상태로 갈 경우에 앞으로 받을 보상의 합에 대한 **기댓값.(E)**
    - G(t+1)은 앞으로 받을 것이라 예상되는 보상이여서 그 가치함수로 표현 가능.
    - 가치 이터레이션
    - 이 가치함수 수식을 그대로, 계산은 가능하지만 상태가 많아질수록 많은 상태에 대해 고려해야 해서 비효율적, 바로 다음 시점까지만 계산하고 반복하는 다이나믹 프로그래밍 방식 사용하고, 이를 가치 이터레이션이라고 함.
  - **행동 가치 함수**
  - 에이전트는 행동 가치 함수를 기준으로 가장 좋은 행동을 선택함.
- 벨만 최적 방정식
  - **최적 상태가치함수, 최적 행동가치 함수**

벨만 방정식 수식

- 벨만 기대 방정식
  - 상태 가치 함수: 한 상태로 앞으로 받을 보상의 합에 대한 기댓값.
  - 행동 가치 함수: 한 상태에서 행동했을 때 받을 수 있는 보상 합의 기댓값.

$$v(s) = E[G_t \mid S_t = s]$$

수식 2.21 가치함수

$$v(s) = E[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} \cdots) \mid S_t = s]$$

$$v(s) = E[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

수식 2.23 반환값으로 나타내는 가치함수

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t$$

수식 2.27 큐함수의 정의

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

수식 2.25 정책을 고려한 가치함수의 표현

벨만 최적 방정식 수식

- 어떤 행동을 할 확률(정책)과 그 행동을 했을 때 가치(보상)을 가지는지를 곱해서도 가치함수를 정의할 수 있음.

$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) q_\pi(s,a)$$

수식 2.26 가치함수와 큐함수 사이의 관계식

$$v_*(s) = \max_a [q_*(s,a) \mid S_t = s, A_t = a]$$

수식 2.39 큐함수 중 최대를 선택하는 최적 가치함수

9에서 큐함수를 가치함수로 고쳐서 표현하면 수식 2.40과 같습니

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

수식 2.25 정책을 고려한 가치함수의 표현

$$v_*(s) = \max_a E[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$

수식 2.40 벨만 최적 방정식

벨만 방정식

- 현재 상태변수의 가치와, 다음 상태 변수 가치와의 관계 나타냄.

$$v(s) = E\,[G_t|S_t = s]$$

$$= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots |S_t = s]$$

$$= E\,[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} \cdots)|S_t = s]$$

↓ t+1에 대한 반환값으로 표현

$$= E\,[R_{t+1} + \gamma G_{t+1}|S_t = s]$$

↓ G_{t+1} 또한 확률 변수로 정해져 있는 특정 값이 아님 (기댓값 사용)

$$= E\,[R_{t+1} + \gamma v(S_{t+1})|S_t = s]$$

$$\boxed{v(s) = E\,[R_{t+1} + \gamma v(S_{t+1})|S_t = s]}$$

벨만 최적 방정식

- MDP 안에 존재하는 모든 정책(π) 중에서, 상태 가치함수의 값을 가장 높게 하는 정책을 선택해서 계산한 밸류가 최적 밸류

  => 최적 밸류를 찾아야할 때 벨만 최적방정식을 사용한다.

  - 상태가치 값을 최대로 만드는 정책을 적용시 상태 가치함수, 행동 가치함수를 최적 상태/행동가치함수라고 함.

(1) $v_*(s) = \max_\pi v_\pi(s)$ → **최적 상태가치 함수**

(2) $q_*(s,a) = \max_\pi q_\pi(s,a)$ → **최적 행동가치 함수**

*증명: https://myetc.tistory.com/36

STUDY PRESENTATION

THANK YOU

발표를 마치겠습니다

궁금한 점은 바로 질문해 주세요!