# 강화학습 기초

2019305050 이제희

# 목차

# MDP(Markov Decision Process)

순차적으로 행동을 결정해야 하는 문제를 풀기 위한 수학 모델
누적 보상을 최대로 하기위한 최적의 정책을 찾는 것이 목표

## MDP의 5가지 요소

- 상태(state)
- 행동(action)
- 보상(reward)
- 상태변환확률(state transition probability)
- 감가율(discounting factor)

# 정책, 반환값

## 정책

각 상태에서 에이전트가 할 행동에 대한 정보

$$\pi(a|s) = P[A_t = a|S_t = s]$$

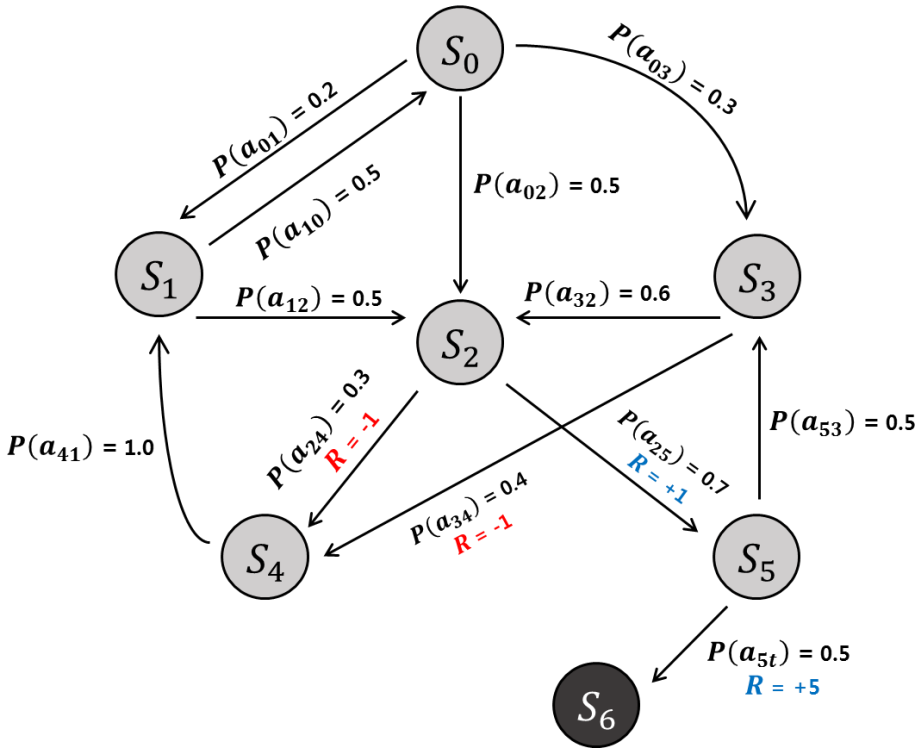
## 반환값

현재 시간 t를 기준으로 에피소드가 끝날 때까지 받을 보상을 감가해서
현재가치를 변환한 보상들의 합

$$G_t = R_{t+1} + \gamma R_{t+2}$$

# 벨만 기대 방정식(상태가치함수)

현재 상태에 대한 가치를 보여주는 방정식




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

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

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

$$v(s) = \sum_{a\in A} \pi(a\,|\,s)(R_{t+1} + \gamma \sum_{s'\in S} P_{ss'}^{a} v(s'))$$

# Q함수(상태-행동가치함수)

상태와 행동 두 가지를 동시에 고려해 가치를 보여주는 방정식

$$\boldsymbol{q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a]}$$

$$\mathrm{q}_\pi(s,a) = E_\pi[R_{t+1} + \gamma \mathrm{q}_\pi(S_{t+1}, A_{t+1}) + |S_t = s, A_t = a]$$

큐 함수로 표현한 가치함수

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

# 최적정책, 벨만 최적 방정식

## 최적정책

큐함수가 최대가 되는 행동을 반환

$$\pi^*(s,a) = \begin{Bmatrix} 1\ if\ a = argmax_{a \in A} q^*(s,a) \\ 0 \quad otherwise \end{Bmatrix}, \text{최적 정책}$$

## 벨만 최적 방정식

정책에 의한 확률적 요소를 배제하고 최적정책만 선택하는 것

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

$$q^*(s,a) = max_a E[R_{t+1} + \gamma q^*(S_{t+1}, a') | S_t = s, A_t = a]$$

# 정책 이터레이션

정책 평가와 정책 발전을 번갈아 수행하며 최적의 정책을 구하는 방법

정책 평가 : 이전 단계의 가치함수로부터 새로운 가치함수를 업데이트
이를 여러 번 반복하게 되면 참 가치함수를 알 수 있음

$$v_{k+1}(s) = \sum_{a \in A} \pi(a \mid s)(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_k(s'))$$

정책 발전 : 정책 평가를 바탕으로 정책을 더 좋은 방향으로 업데이트
현재 상태에서 선택가능한 가장 큰 큐함수를 가지는 행동을 수행함

# 가치 이터레이션

정책 평가만 수행해 가장 큰 값을 가지는 값으로 업데이트하는 방법

정책평가 : 벨만 최적 방정식을 사용

$$v_{k+1}(s) \equiv max_a E[R_{t+1} + \gamma v_k(s_{t+1}) | S_t = s, A_t = a]$$
$$= max_a \sum_{s'} P^a_{ss'} [R_{t+1} + \gamma v_k(s')]$$