벨만 방정식(Bellman Equation)

π, s $\xrightarrow{\text{Estimate (recursive)}}$ V

- Recap

– 마르코프 결정 프로세스(Markov Decision Process)

$$\mathbf{MDP} \equiv (\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}, \boldsymbol{\gamma})$$

– 전이 확률 행렬 (Transmission Matrix)

$$P_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

– 보상함수(Reward Function)

$$R_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

– 정책 함수(Policy Function)

$$\boldsymbol{\pi}(\mathbf{a}|\mathbf{s}) = \mathbb{P}[A_t = a | S_t = s]$$

상태 s에서 액션 a를 선택할 확률

– 상태 가치 함수(State Value Function)

$$v_\pi(s) = \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s]$$
$$= \mathbb{E}_\pi[G_t | S_t = s]$$

s부터 끝까지 π를 따라서 움직일 때 얻는 리턴의 기댓값

– 액션 가치 함수(State-action Value Function)

$$\mathbf{q}_\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

s에서 a를 선택하고, 그 이후에는 π를 따라서 움직일 때 얻는 리턴의 기댓값

벨만 기대 방정식(Bellman Expectation Equation)

$$v_\pi(s_t) = \mathbb{E}_\pi[r_{t+1} + \gamma v_\pi(s_{t+1})]$$

$$q_\pi(s_t, a_t) = \mathbb{E}_\pi[r_{t+1} + \gamma q_\pi(s_{t+1}, a_{t+1})]$$

: current state의 value = 1-step(immediate) reward + next state의 value

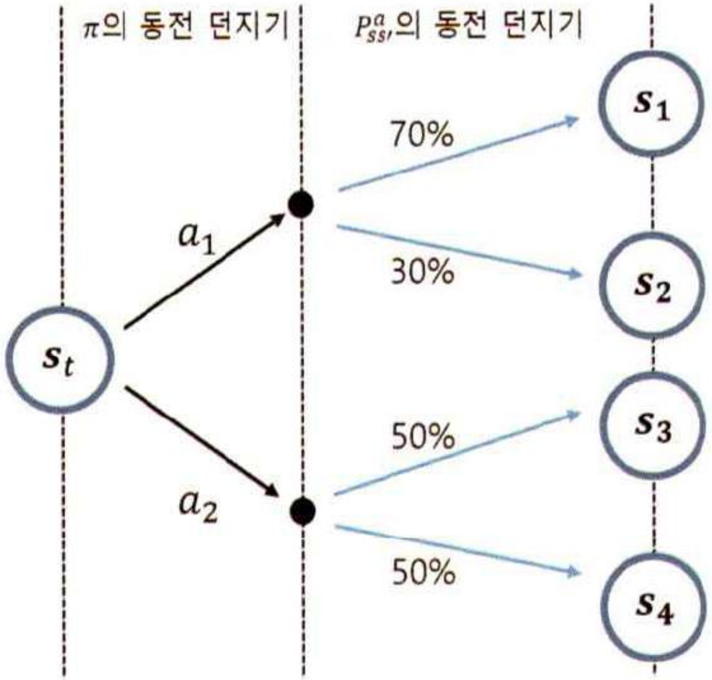
$$\begin{aligned} v_\pi(s_t) &= \mathbb{E}_\pi[G_t] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma(r_{t+2} + \gamma r_{t+3} + \cdots)] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma G_{t+1}] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma v_\pi(s_{t+1})] \end{aligned}$$

벨만 기대 방정식

OX 퀴즈

❶ $v_\pi(s_t) = r_{t+1} + \gamma v_\pi(s_{t+1})$가 성립한다.

❷ $v_\pi(s_t) = \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+1} + \gamma^2 v_\pi(s_{t+2})]$가 성립한다.



Policy가
action을 선택
(by agent)

Transmission Matrix가
next state를 선택
(by environment)

[MDP의 특징]
current state는 deterministic
next state는 non-determintsic

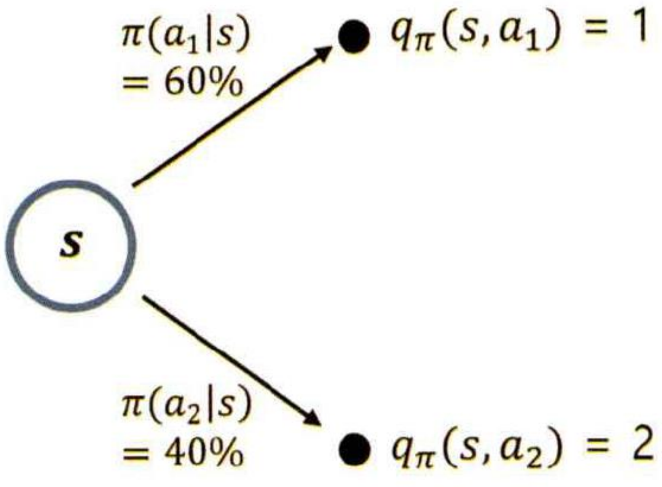
벨만 기대 방정식

$$\underbrace{v_\pi(s)}_{s\text{의 밸류}} = \sum_{a \in A} \underbrace{\pi(a|s)}_{s\text{에서 }a\text{를 실행할 확률}} \underbrace{q_\pi(s,a)}_{s\text{에서 }a\text{를 실행하는 것의 밸류}}$$

: current state에서 선택할 수 있는 모든 action들의 value -〉 current state의 value

Cf. stationary policy vs non-stationary policy

	stationary	**non-stationary**
dependency	s	s,t
case	infinite-horizon case	finite-horizon case



| 그림 3-3 | 액션 밸류로 상태 밸류 계산하기

$$\begin{aligned} v_\pi(s) &= \pi(a_1|s) * q_\pi(s,a_1) + \pi(a_2|s) * q_\pi(s,a_2) \\ &= 0.6 * 1 + 0.4 * 2 \\ &= 1.4 \end{aligned}$$

벨만 기대 방정식

$$q_\pi(s,a) = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \, v_\pi(s')$$

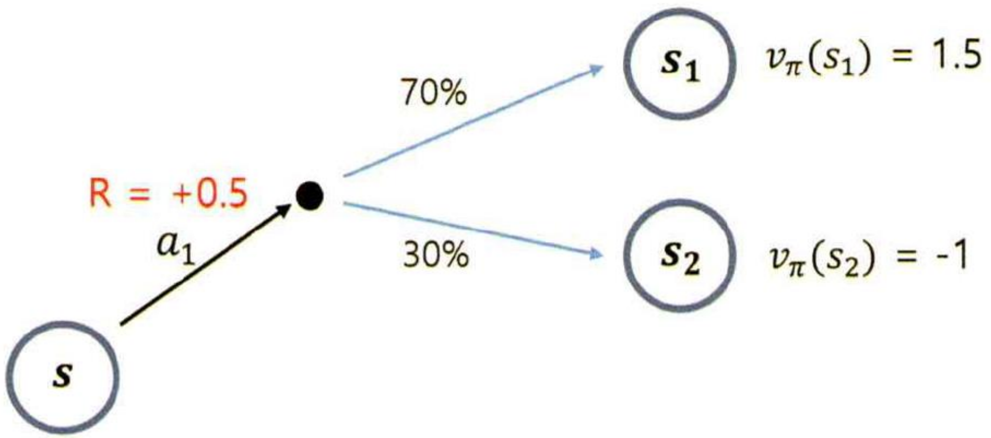
s에서 a를 실행하는 것의 밸류

즉시 얻는 보상

s에서 a를 실행하면 s'에 도착할 확률

s'의 밸류

: next state의 value -〉 current state의 action들의 value
(마지막 step의 value=immediate reward -〉 1step씩 거슬러 올라가면서 계산)



| 그림 3-4 | 상태 밸류로 액션 밸류 평가하기

$$\begin{aligned} q_\pi(s,a_1) &= r_s^{a_1} + P_{ss_1}^{a_1} * v_\pi(s_1) + P_{ss_2}^{a_1} * v_\pi(s_2) \\ &= 0.5 + 0.7 * 1.5 + 0.3 * (-1) \\ &= 1.25 \end{aligned}$$

벨만 기대 방정식

$$v_\pi(s) = \sum_{a\in A} \pi(a|s)\, \underline{q_\pi(s,a)} = \sum_{a\in A} \pi(a|s) \left(r_s^a + \gamma \sum_{s'\in S} P_{ss'}^a\, v_\pi(s') \right)$$

대입

$$q_\pi(s,a) = r_s^a + \gamma \sum_{s'\in S} P_{ss'}^a\, v_\pi(s')$$

$$q_\pi(s,a) = r_s^a + \gamma \sum_{s'\in S} P_{ss'}^a\, \underline{v_\pi(s')} = r_s^a + \gamma \sum_{s'\in S} P_{ss'}^a \sum_{a'\in A} \pi(a'|s')\, q_\pi(s',a')$$

대입

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

벨만 기대 방정식

$$v_\pi(s) = \mathbb{E}_\pi[r' + \gamma v_\pi(s')]$$: MDP를 모를 때(model-free) 접근법 - experience를 통한 학습(ex. sampling 이용)

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \left(r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \, v_\pi(s') \right)$$: MDP를 알 때(model-based/planning) 접근법

"MDP를 안다" - 보상 함수&전이행렬을 안다

- **보상 함수** r_s^a : 각 상태에서 액션을 선택하면 얻는 보상
- **전이 확률** $P_{ss'}^a$: 각 상태에서 액션을 선택하면 다음 상태가 어디가 될지에 관한 확률 분포

벨만 최적 방정식(Bellman Optimal Equation)

- Optimal Value & Optimal Policy

$$v_*(s) = \max_\pi v_\pi(s)$$

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

: MDP안에 모든 policy 중 가장 좋은(value가 가장 큰) policy의 value

모든 상태 s에 대해, $v_{\pi_1}(s) > v_{\pi_2}(s)$ 이면 $\pi_1 > \pi_2$ 이다. (partial ordering)

MDP 내의 모든 π에 대해 $\pi_* > \pi$를 만족하는 π_*가 반드시 존재한다.

: 모든 starting state에서 optimal한 stationary policy가 존재한다.
Cf. non-stationary의 경우 t에 따라 policy가 변함
 -〉 모든 step에서의 optimal policy & optimal value를 알아야 starting state를 평가할 수 있음

R.A. Howard, Dynamic Programming and Markov Processes, MIT Press, Cambridge, MA, 1960.

벨만 최적 방정식

- Bellman optimal equation

- 최적의 정책 : π_*
- 최적의 밸류 : $v_*(s) = v_{\pi_*}(s)$ (π_*를 따랐을 때의 밸류)
- 최적의 액션 밸류 : $q_*(s,a) = q_{\pi_*}(s,a)$ (π_*를 따랐을 때의 액션 밸류)

: 각 state에서의 action이 optimal policy를 따르므로 (deterministic)
agent에서 policy가 action을 선택하는 과정의 확률적인 계산이 사라짐

$$v_*(s_t) = \max_a \mathbb{E}[r_{t+1} + \gamma v_*(s_{t+1})]$$

$$q_*(s_t, a_t) = \mathbb{E}[r_{t+1} + \gamma \max_{a'} q_*(s_{t+1}, a')]$$

$$v_*(s) = \max_a q_*(s,a)$$

$$q_*(s,a) = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

$$v_*(s) = \max_a \left[r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s') \right]$$

$$q_*(s,a) = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} q_*(s',a')$$

벨만 최적 방정식

- Bellman optimal equation

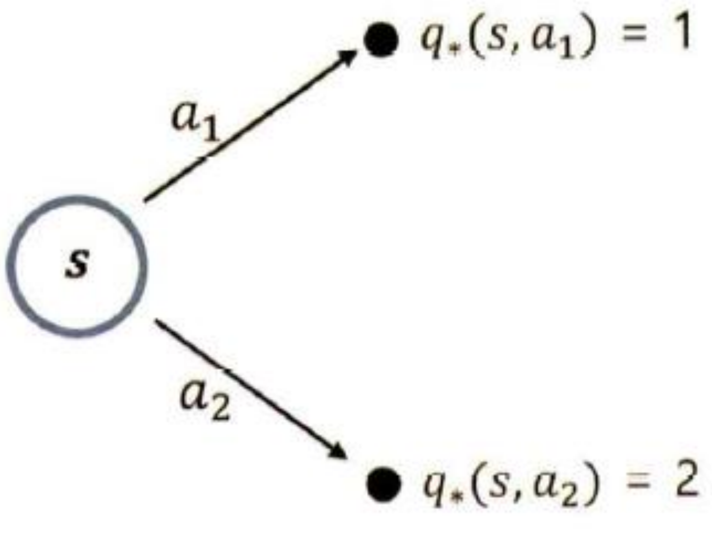
$$v_*(s) = \max_a \mathbb{E}[r + \gamma v_*(s') \mid s_t = s, a_t = a]$$: $E_\pi \rightarrow E$

$$q_*(s,a) = \mathbb{E}[r + \gamma \max_a q_*(s', a') \mid s_t = s, a_t = a]$$: Q-Learning에 이용

$$v_*(s) = \max_a q_*(s,a)$$

$$q_*(s,a) = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

$$\begin{aligned} v_*(s) &= \max_a q_*(s,a) \\ &= \max(q_*(s,a_1), q_*(s,a_2)) \\ &= \max(1,2) = 2 \end{aligned}$$



| 그림 3-5 | 두 개의 액션이 선택 가능한 상태 s

벨만 최적 방정식

- Bellman optimal equation

$$v_*(s) = \max_a q_*(s,a) = \max_a \left[r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s') \right]$$

대입

$$q_*(s,a) = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

$$q_*(s,a) = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s') = r_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} q_*(s',a')$$

대입

$$v_*(s') = \max_{a'} q_*(s',a')$$

Summary

given π,s
(if π is optimal)

$\xrightarrow{\text{Bellman equation}}$

estimate π
(values are maximum)

- **Reference**

- 노승은, 바닥부터 배우는 강화학습, 영진닷컴, 2020.
- Leslie Pack Kaelbling, Planning and acting in partially observable stochastic domains, 1998.
- David Silver, UCL Course on RL, https://www.davidsilver.uk/teaching/