2. 마르코프 의사결정 프로세스 (Markov Decision Process)

순천향대학교 컴퓨터공학과

이 상 정

학습 내용

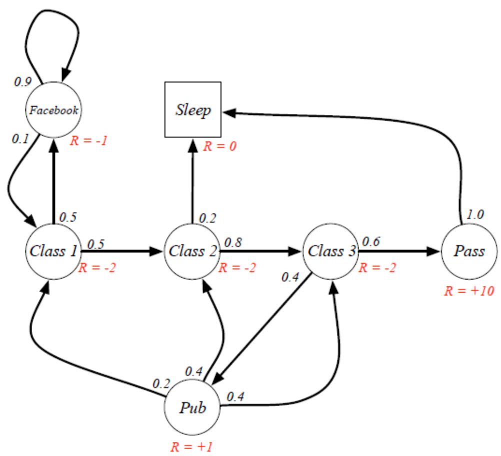
1. 마르코프 프로세스 (Markov Process)
2. 마르코프 보상 프로세스 (Markov Reward Processes)
3. 마르코프 의사결정 프로세스 (Markov Decision Processes)

1. 마르코프 프로세스 (Markov Process)

마르코프 의사결정 프로세스

- **마르코프 의사결정 프로세스 (Markov Decision Process, MDP)은 강화학습에서 환경을 기술하는 수학 프레임워크**
  - **에이전트가 환경에서 발생하는 모든 정보를 볼 수 있다고 가정 (완전한 관찰, *fully observable*)**
  - **현재 상태는 진행이 되고 있는 프로세스 중에서 특정 시점이고, 모든 환경을 다 볼 수 있는 완전한 특성을 갖춤**

Andrey Markov
(1856-1922)



마르코프 프로퍼티 (Markov Property)

❑ 마르코프 프로퍼티 (Markov Property) 정의

- 미래는 현재 상태에만 의존하고, 과거와는 독립적

Definition

A state S_t is *Markov* if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

- 현재의 상태는 과거의 모든 관련 정보를 포함
- 다가오는 미래를 예측하는데 충분한 정보를 포함
- 강화학습은 현재의 시점에서 미래 가치를 예측하여 의사결정

상태 전이 행렬 (State Transition Matrix)

❑ 현재의 상태인 s 와 연속된 다음의 상태를 s' 라고 했을때 상태가 s에서 s'로 변경될 상태 전이 확률 (State Transition Probability)의 정의

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

❑ 상태 전이 행렬 (State Transition Matrix) P는 모든 상태 s 의 다음의 상태 s' 로 변경될 확률을 정의

- 각 행의 합은 1

$$\mathcal{P} \;\; = \begin{array}{cc} & \textit{to} \\ \textit{from} & \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix} \end{array}$$

마르코프 프로세스 (Markov Process)

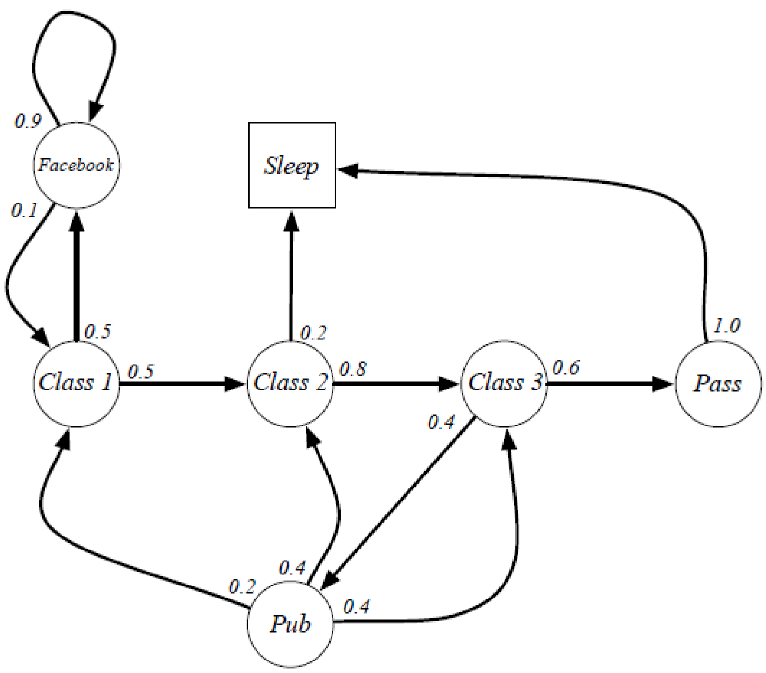
- 마르코프 프로세스 (Markov Process)는 과거를 기억하지 않는 마르코프 프로퍼티를 갖는 랜덤한 상태들의 시퀀스와 상태가 변경(전이)될 확률로 표현되는 랜덤 프로세스 (random process)
  - 마르코프 체인 (Markov Chain)이라고도함

Definition

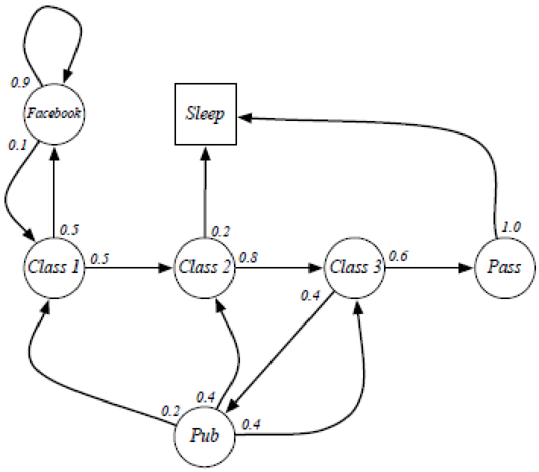
A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$

마르코프 프로세스 예: 학생 마르코프 체인



학생 마르코프 체인 예: 에피소드 예

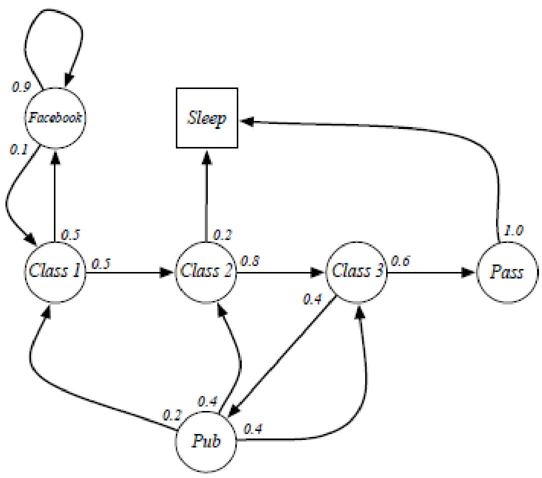


Sample episodes for Student Markov Chain starting from $S_1 = C1$

$$S_1, S_2, ..., S_T$$

- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

학생 마르코프 체인 예: 상태 전이 행렬



$$\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array} \begin{array}{c} \begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\ \left[\begin{array}{ccccccc} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ & & & & & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{array}\right] \end{array}$$

2. 마르코프 보상 프로세스 (Markov Reward Process)

마르코프 보상 프로세스 (Markov Reward Process)

- 마르코프 보상 프로세스 (Markov Reward Process, MRP)는 마르코프 체인에 가치(value) 개념을 추가
  - 현재 상태에서 다음 상태로 변경 시 받게 될 보상 (reward)
  - 0과 1사이의 값을 갖는 할인율 (discount factor)
    - 미래에 받게 될 보상은 할인율을 적용하여 현재 즉시 받게 될 보상과 다른 가치를 적용

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}\left[R_{t+1} \mid S_t = s\right]$
- γ is a discount factor, $\gamma \in [0, 1]$

할인율 (Discount Factor)

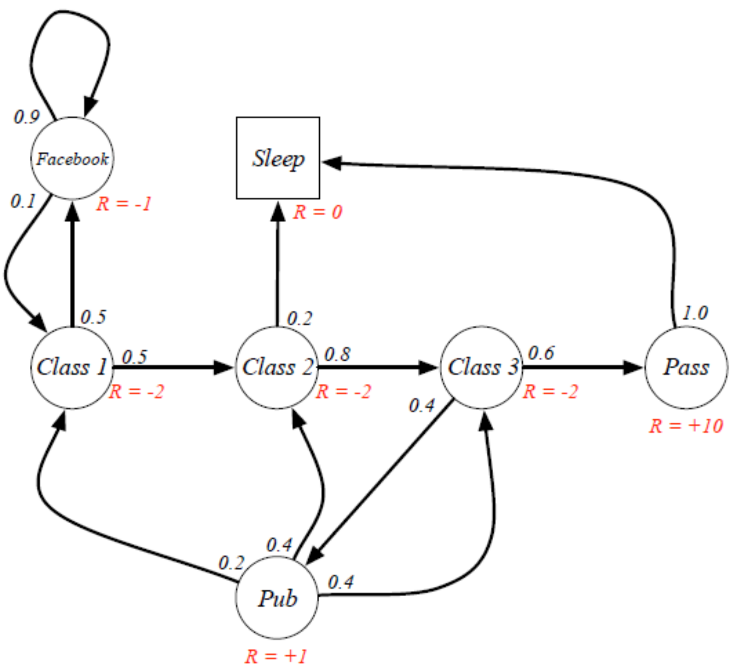
- 현재 얻게 되는 보상이 미래에 얻게 될 보상보다 얼마나 더 중요한지를 나타내는 값으로 0과 1사이의 값

$$\gamma \in [0, 1]$$

  - 스텝 t에서 미래를 포함한 전체 보상

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

학생 마르코프 체인 예: MRP



리턴 (Return)

- 리턴 (Return)은 현재 시점에서 미래에 받게 될 보상까지 고려

Definition

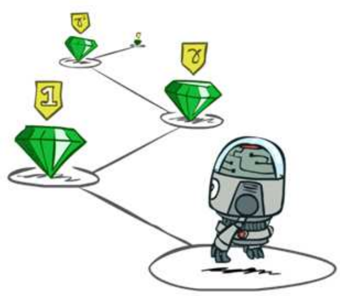
The *return* G_t is the total discounted reward from time-step t.

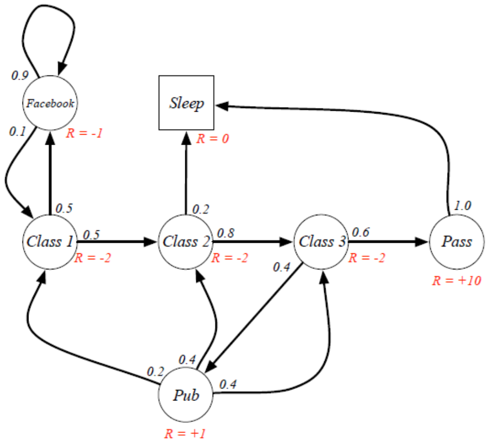
$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- 할인율 $\gamma \in [0,1]$ 은 미래의 보상을 현재의 가치로 환산
- k+1 시간 스텝 후의 보상 R의 가치는 $\gamma^k R$.
- 즉시 받는 보상을 미래의 지연된 보상보다 높게 평가
  - 할인율이 0에 가까우면 근시안적인 평가
  - 할인율이 1가까우면 원시적안적인 평가

할인율 적용 이유

- 대부분의 강화학습에서는 다음과 같은 이유로 할인율을 적용
  - 수학적으로 계산이 편리
  - 사이클릭 마르코프 프로세스에서 무한대의 리턴 값을 방지
  - 미래의 불확실성에 대해 할인
  - 보상이 재정적인 경우 즉시 받는 보상이 지연된 보상보다 더 많은 이자 수익 유발
  - 동물/인간의 행동은 즉시 받는 보상을 선호





학생 MRP 예: 리턴 예

Sample returns for Student MRP:
Starting from $S_1 = \text{C1}$ with $\gamma = \frac{1}{2}$

$$G_1 = R_2 + \gamma R_3 + ... + \gamma^{T-2} R_T$$

C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	=	-2.25
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	=	-3.125
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	=	-3.41
C1 FB FB C1 C2 C3 Pub C1 ...	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	=	-3.20
FB FB FB C1 C2 C3 Pub C2 Sleep			

마르코프 의사결정 프로세스

가치함수 (Value Function)

- **가치함수 (Value Function)는 현재 상태에서 미래의 모든 기대하는 보상들을 표현**
  - **상태만을 고려한 상태-가치함수(state-value function)**
  - **미래의 가치를 표현**

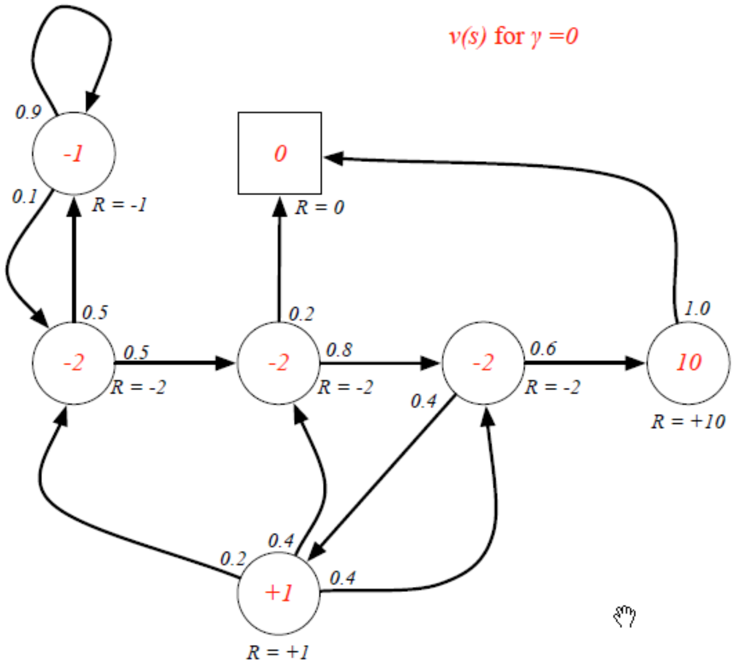
Definition

The *state value function* $v(s)$ of an MRP is the expected return starting from state s

$$v(s) = \mathbb{E}\left[G_t \mid S_t = s\right]$$

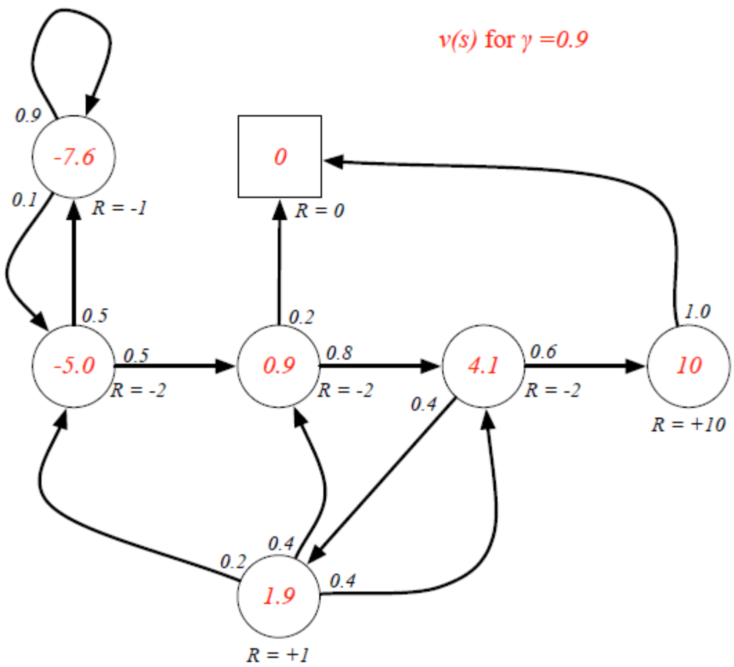
- **강화학습에서는 가치함수를 정확하게 표현하는 것이 핵심**
  - **미래 가치가 가장 큰 의사결정을 하고 행동하는 것이 최종 목표**

학생 MRP 예: 가치함수 예, 할인율 = 0



마르코프 의사결정 프로세스

학생 MRP 예: 가치함수 예, 할인율 = 0.9

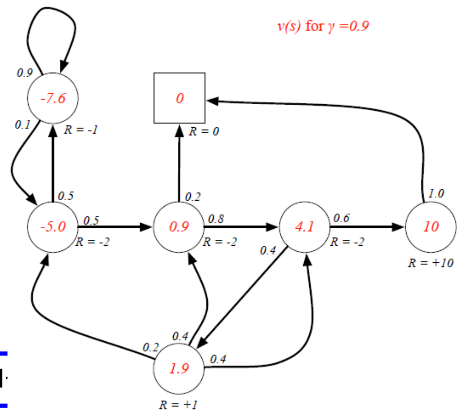


학생 MRP 예: 가치함수 계산 예 (참고)

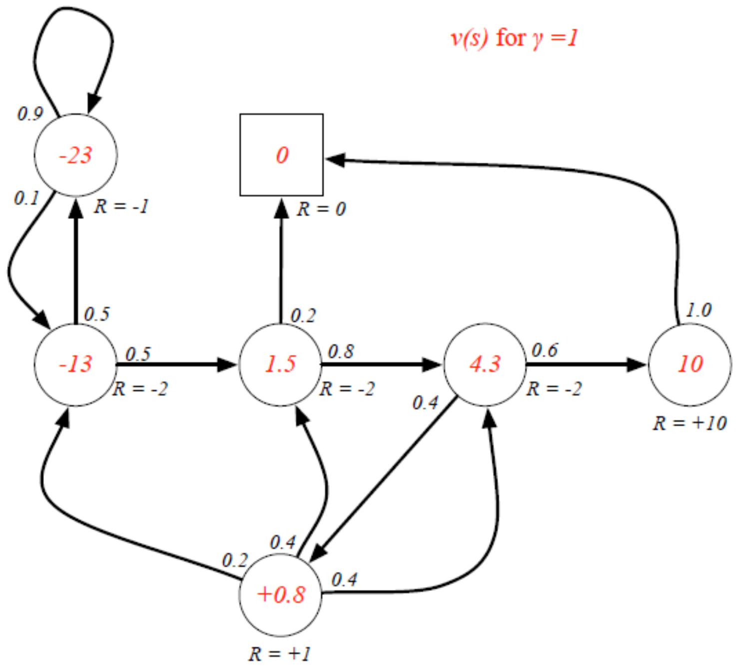
- ***각 상태의 가치 함수의 값은 이 후 소개하는 벨만 방정식으로 계산***

$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

- ***C3의 가치함수 값 계산 예 (r=0.9)***
  - ***v(c3) = R_{c3} + r * (0.6 * v(pass) + 0.4 * v(pub))***
    = -2 + 0.9 * (0.6 * 10 + 0.4 * 1.9)
    = 4.084



학생 MRP 예: 가치함수 예, 할인율 = 1



MRP의 벨만 방정식 (Bellman Equation) (1)

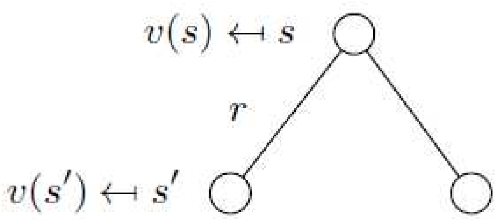
- **벨만 방정식(Bellman Equation)은 동적 계획법(dynamic programming)에서의 사용된 방정식**

- **가치함수는 아래의 두 요소로 구성**
  - 즉시 받는 보상 R_{t+1}
  - 할인된 다음 상태의 가치함수 $\gamma v(S_{t+1})$
    - 재귀적인 형태로 미래의 가치들이 현재의 가치에 영향

$$
\begin{aligned}
v(s) &= \mathbb{E}[G_t \mid S_t = s] \\
&= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s\right] \\
&= \mathbb{E}[R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \dots) \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]
\end{aligned}
$$

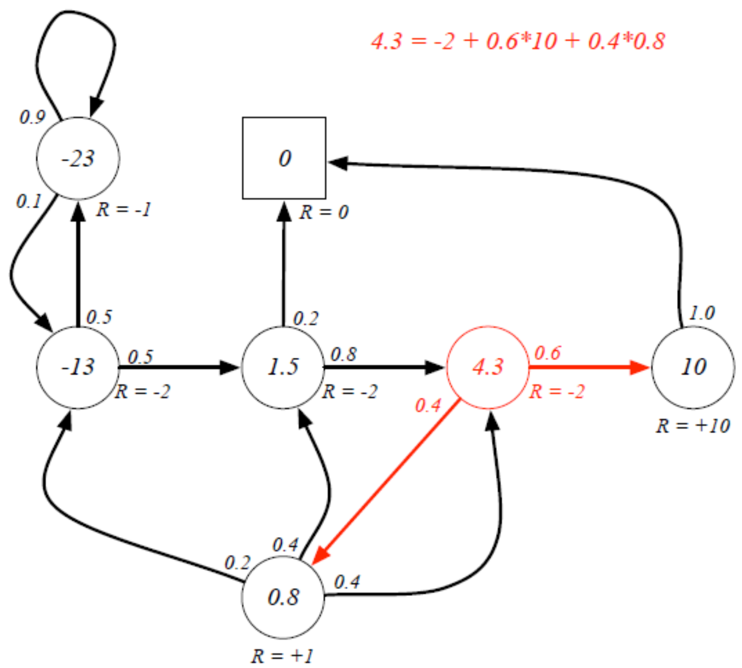
MRP의 벨만 방정식 (Bellman Equation) (2)

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



$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

학생 MRP 예: 벨만 방정식



벨만 방정식의 행렬 표현

- 모든 상태가 포함된 벨만 방정식은 행렬 (matrix) 형태로 표현

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

  - v는 각 상태 당 하나의 엔트리를 갖는 열 벡터(column vector)

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{11} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

벨만 방정식의 해 (Solution)

- 벨만 방정식은 선형 방정식(linear equation)
- 아래와 같이 직접 해를 구할 수도 있음

$$
\begin{aligned}
v &= \mathcal{R} + \gamma \mathcal{P} v \\
(I - \gamma \mathcal{P}) v &= \mathcal{R} \\
v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R}
\end{aligned}
$$

  - n개의 상태에 대해 $O(n^3)$ 계산 복잡도
  - 작은 크기의 MRP에 대해서만 직접 해를 구할 수 있음
- 일반적으로 큰 MRP는 아래와 같은 반복적인 방식(iterative method) 사용하여 해를 구함
  - 동적 계획법 (Dynamic Programming)
  - 몬테카를로 방법 (Monte-Carlo Method)
  - 시간차 학습 (Temporal-Difference Learning)

3. 마르코프 의사결정 프로세스 (Markov Decision Process)

마르코프 결정 프로세스 (Markov Decision Process, MDP)

❑ **마르코프 의사결정 프로세스 (Markov Decision Process, MDP)**는 MRP(Markov Reward Process)에 의사결정의 개념을 추가

- **행동(action)**이 추가

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'}^{a} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$
- γ is a discount factor $\gamma \in [0, 1]$.

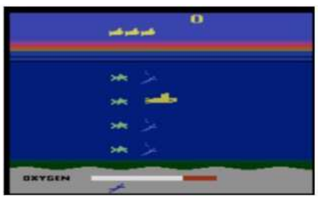
MDP 상태 (State)와 행동 (Action)

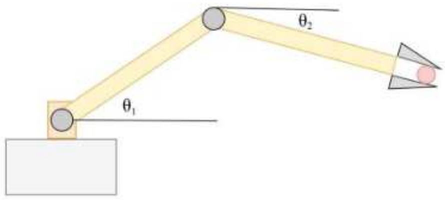
❑ **MDP의 상태(status)**는 에이전트가 인식하고 관찰하는 상태

- 상태 예
  - 게임에서는 게임 이미지 자체(픽셀)가 상태
  - 로봇 제어에서는 센서가 측정한 조인트 각도, 속도 등

❑ 환경의 특정 상태에서 에이전트가 지시하는 **행동(action)**

- 행동 예
  - 아타리 게임에서 조이스틱 움직임
  - 로봇 팔의 이동

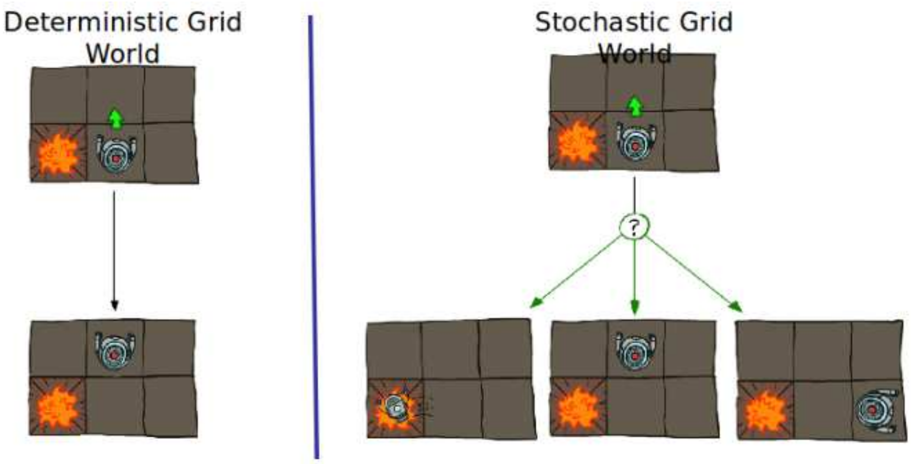




상태 전이 확률 행렬 (State Transition Probability Matrix)

- 에이전트가 특정 상태에서 특정 행동을 취할 때 전이될 상태의 확률

$$\mathcal{P}_{ss'}^{a} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$$

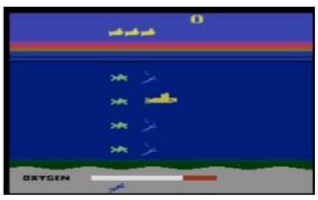


보상 (Reward)

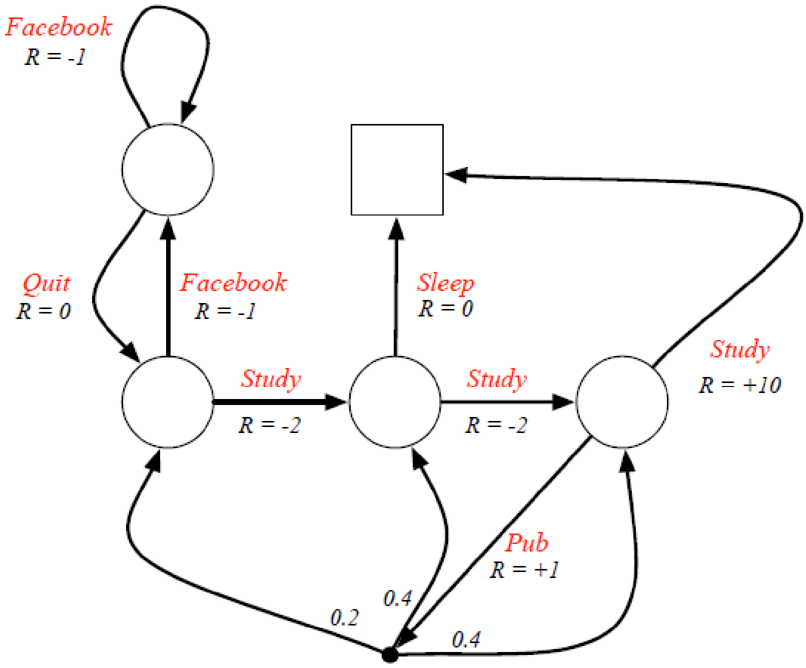
- 에이전트가 취한 행동에 따라 환경이 알려주는 보상 (reward)

$$\mathcal{R}_{s}^{a} = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$$

- 보상 예
  - 게임에서는 점수
  - 바둑에서는 승패
  - 궤도 제어에서는 의도한 궤도에 얼마나 가깝게 움직였는지 여부



학생 MDP 예



정책 (Policy) (1)

- **정책 (policy)는 현재 상태에서 에이전트가 어떤 행동 (action)을 취할 확률**

Definition

A *policy* π is a distribution over actions given states,

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

- MDP의 정책은 **현재의 상태**만 고려하고, 과거의 정보는 고려하지 않고 행동
- **확률적**으로 행동을 결정
- 정책은 시간 스텝의 변화와 무관하게 독립적

$$A_t \sim \pi(\cdot|S_t), \forall t > 0$$

정책 (Policy) (2)

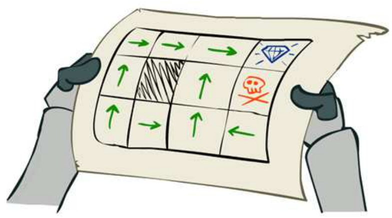
- MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 와 정책 ∏ 가 주어진 경우 상태 전이 확률과 보상에 따라 정책의 행동을 취할 확률 고려

- 상태 시퀀스 $S_1, S_2, ...$ 는 마르코프 프로세스 $\langle \mathcal{S}, \mathcal{P}^{\pi} \rangle$
- 상태와 보상 시퀀스 $S_1, R_2, S_2, ...$ 는 마르코프 보상 프로세스(MRP) $\langle \mathcal{S}, \mathcal{P}^{\pi}, \mathcal{R}^{\pi}, \gamma \rangle$

여기서,

$$\mathcal{P}^{\pi}_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^{a}_{ss'}$$

$$\mathcal{R}^{\pi}_{s} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^{a}_{s}$$



가치함수 (Value Function)

- 상태-가치함수 (state-value function) v는 현재 상태에서 정책의 기대되는 미래의 모든 보상의 합(리턴)
  - 현재 상태에서 모든 행동을 고려한 가치

Definition

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right]$$

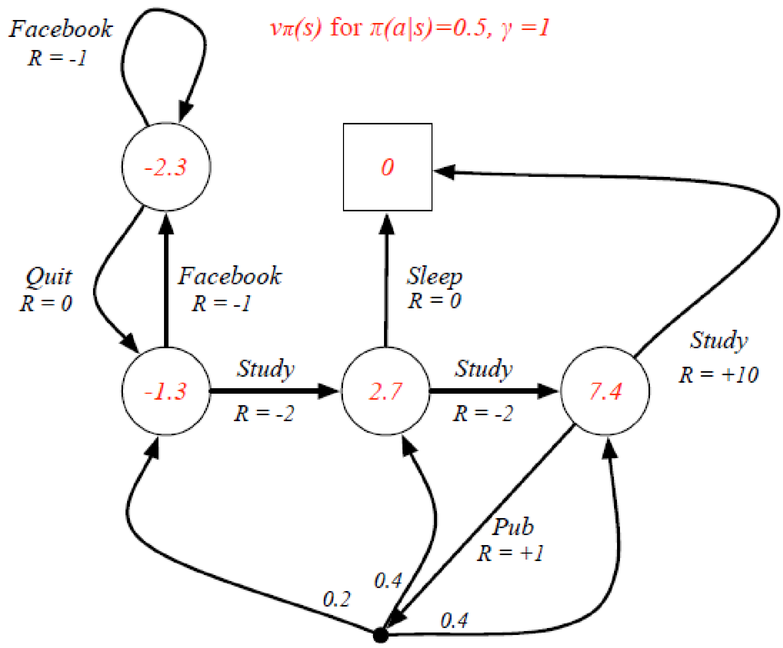
- 행동-가치함수 (action-value function) q은 현재 상태에서 특정 행동을 취하는 조건에서 정책의 기대되는 미래의 모든 보상의 합(리턴)
  - 현재 상태에서 특정 행동만 고려한 가치

Definition

The *action-value function* $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π

$$q_\pi(s, a) = \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$$

학생 MDP 예: 상태-가치함수



벨만 기대 방정식 (Bellman Expectation Equation)

- 상태/행동 가치함수도 벨만 방정식으로 표현
- 상태-가치함수는 현재 상태에서 정책을 따르는 즉시 받는 보상과 할인율을 적용한 다음 상태의 가치로 분리 표현

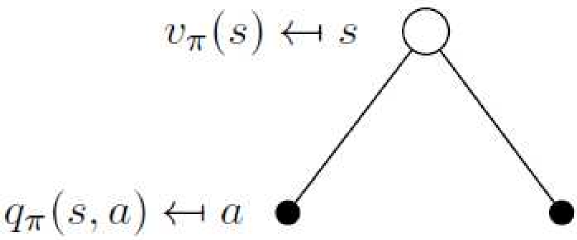
$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right]$$

- 행동-가치함수는 현재 상태에서 특정 행동을 취했을 때 즉시 받는 보상과 할인율을 적용한 다음 상태의 행동-가치로 분리 표현

$$q_\pi(s, a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a \right]$$

벨만 기대 방정식: 상태-가치함수 V^π (1)

- **상태-가치함수**는 현재 상태에서 정책에 따라 취할 수 있는 **모든 행동들의 행동-가치함수의 합**

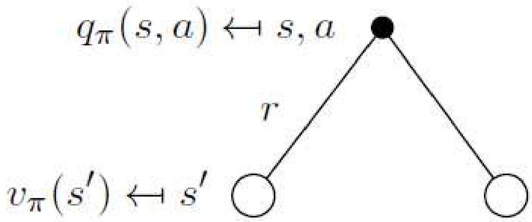


$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

마르코프 의사결정 프로세스

벨만 기대 방정식: 행동-가치함수 Q^π(1)

- **행동-가치함수**는 현재 상태에서 정책의 **특정 행동을 취했을 때 받는 보상**과 **전이될 수 있는 모든 다음 상태들의 상태-가치함수의 합**

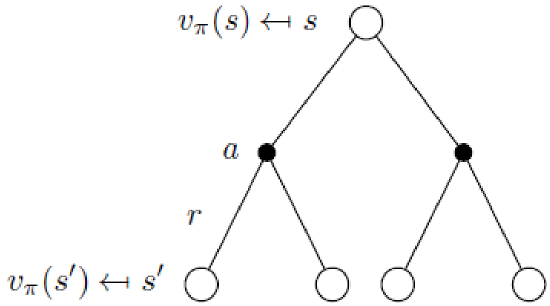


$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

벨만 기대 방정식: 상태-가치함수 V$^\pi$ (2)

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

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$



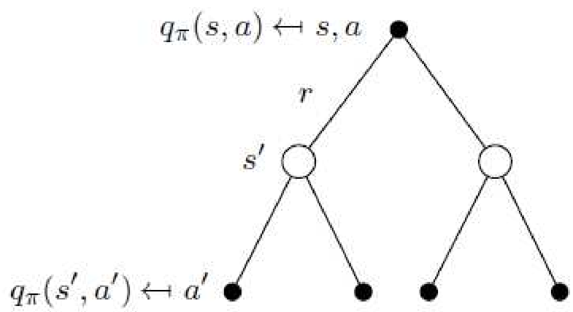
$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

마르코프 의사결정 프로세스

벨만 기대 방정식: 행동-가치함수 Q$^\pi$ (2)

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

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$



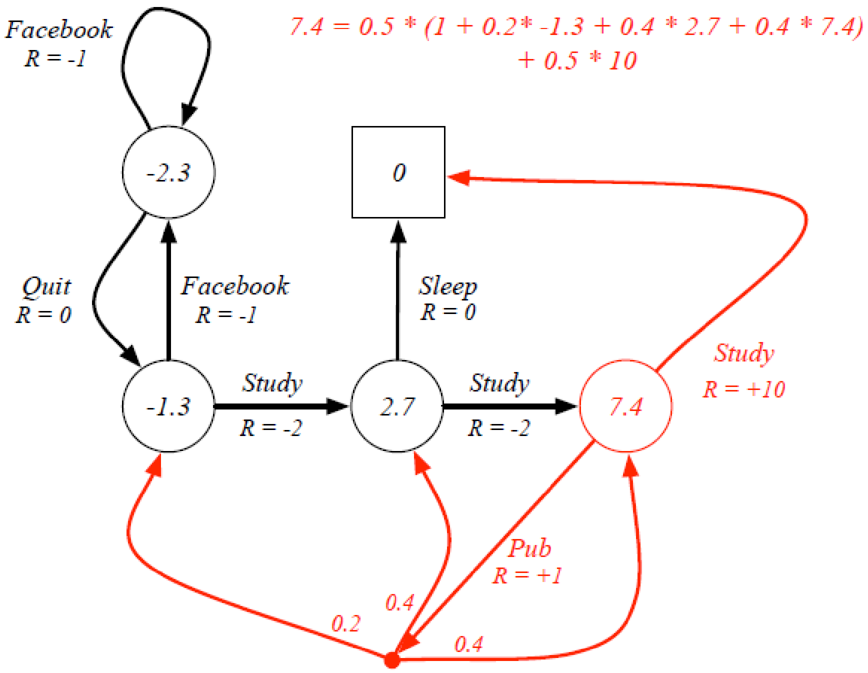
$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s',a')$$

학생 MDP 예: 벨만 기대 방정식 (상태-가치함수 V) 예

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

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

$\gamma = 1$



벨만 기대 방정식의 행렬 표현

- **벨만 기대 방정식**은 MRP와 유사하게 **행렬 (matrix)** 형태로
  표현

$$v_\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v_\pi$$

- **직접 해 (direct solution)**

$$v_\pi = (I - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}^\pi$$

최적화된 가치함수 (Optimal Value Function)

❑ 가치함수의 최적화

- MDP의 정책 중에서 최대의 가치를 갖는 정책의 가치함수
  - 최적화된 상태-가치함수, 최적화된 행동-가치함수
- MDP의 해(solution)은 최적화된 가치함수를 구하는 것

Definition

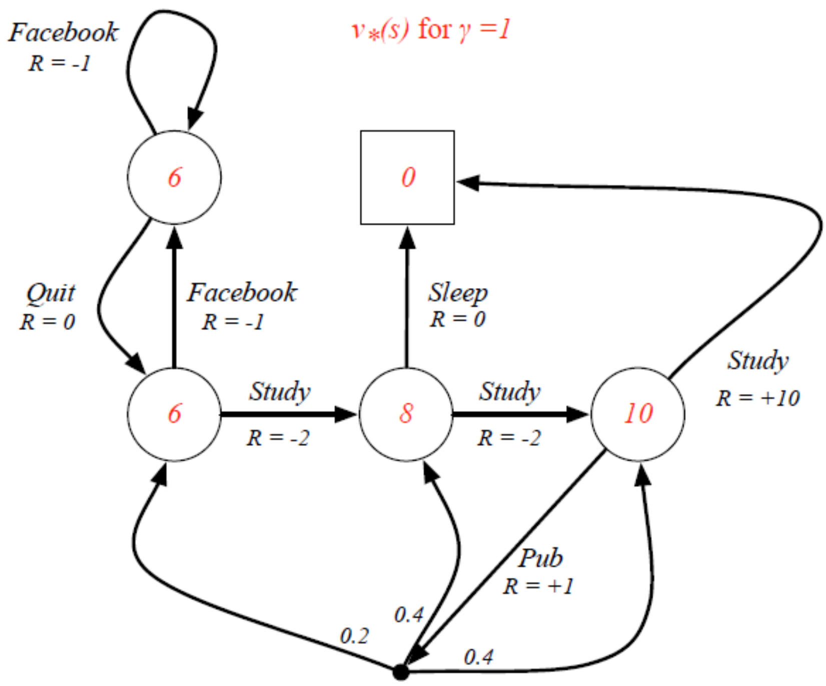
The *optimal state-value function* $v_*(s)$ is the maximum value function over all policies

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

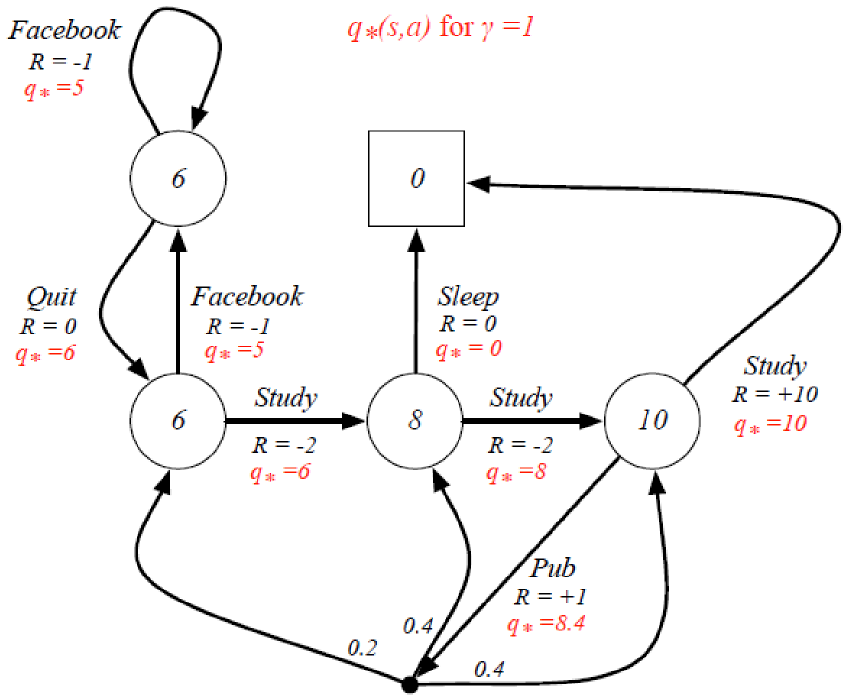
The *optimal action-value function* $q_*(s, a)$ is the maximum action-value function over all policies

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

학생 MDP 예: 최적화된 (상태-)가치함수



학생 MDP 예: 최적화된 행동-가치함수 예



최적화된 정책 (Optimal Policy)

- 모든 상태에서 한 정책의 가치함수가 다른 정책의 가치함수보다 크거나 같으면 정책이 크다고 정의(좋은 결과의 정책)

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \forall s$$

- 모든 상황에서 가장 최적이 되는 정책이 하나 이상 존재
- 최적화된 정책은 최적화된 가치함수와 행동-가치함수를 달성

Theorem

For any Markov Decision Process

- *There exists an optimal policy π_* that is better than or equal to all other policies, $\pi_* \geq \pi, \forall \pi$*
- *All optimal policies achieve the optimal value function, $v_{\pi_*}(s) = v_*(s)$*
- *All optimal policies achieve the optimal action-value function, $q_{\pi_*}(s,a) = q_*(s,a)$*

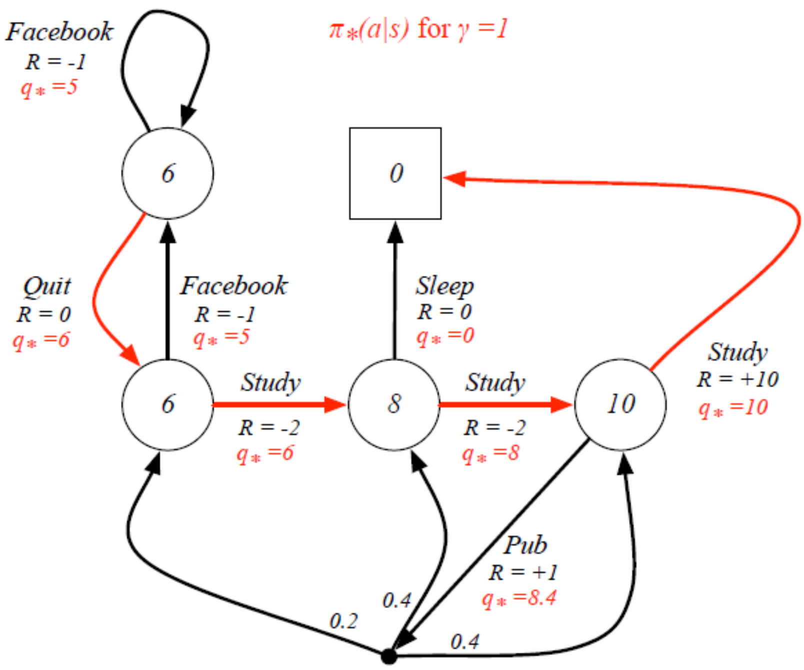
최적화된 정책 발견

- 최적화된 정책은 각 상태에서 최적화된 행동-가치함수 q_* 를 최대화하는 것

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\text{argmax}}\ q_*(s,a) \\ 0 & otherwise \end{cases}$$

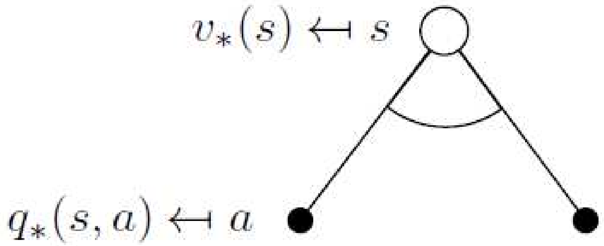
- q_*가 최대값이되는 행동을 선택하면 되므로, q*(s,a)를 알면 최적화된 정책을 알 수 있음
- 결정론적 최적화 정책 (deterministic optimal policy)

학생 MDP 예: 최적화된 정책



벨만 최적화 방정식 (Bellman Optimality Equation): 최적화된 가치함수 V* (1)

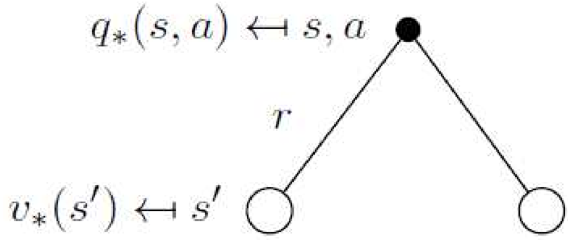
- 최적화된 가치함수 v_*는 최대값을 갖는 q_*를 선택



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

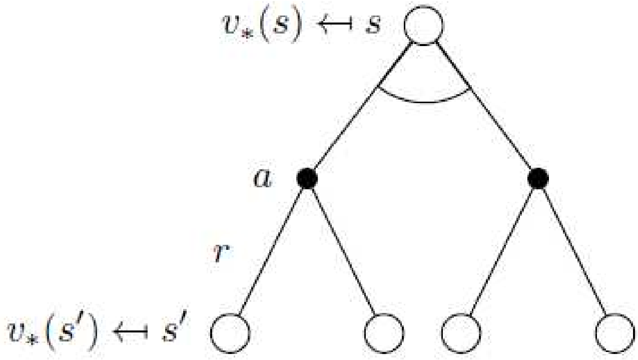
벨만 최적화 방정식: 최적화된 행동-가치함수 Q* (1)

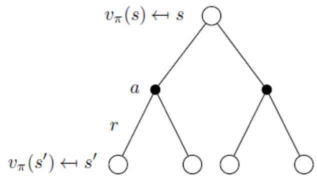
- 최적화된 행동-가치함수 q_*는 다음 상태의 최적화된 가치 함수를 반영하여 표현



$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

벨만 최적화 방정식: 최적화된 가치함수 V* (2)

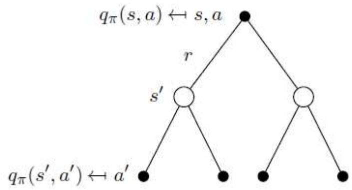




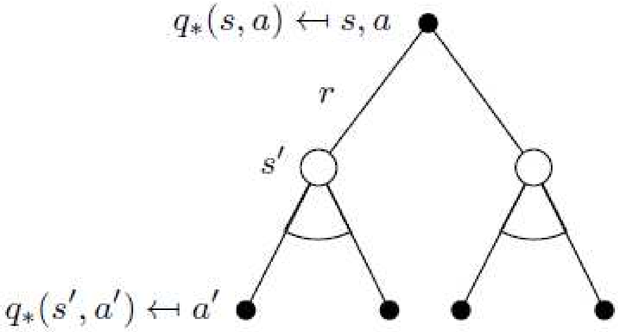
$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

$$v_*(s) = \max_a \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

벨만 최적화 방정식: 최적화된 행동-가치함수 Q* (2)

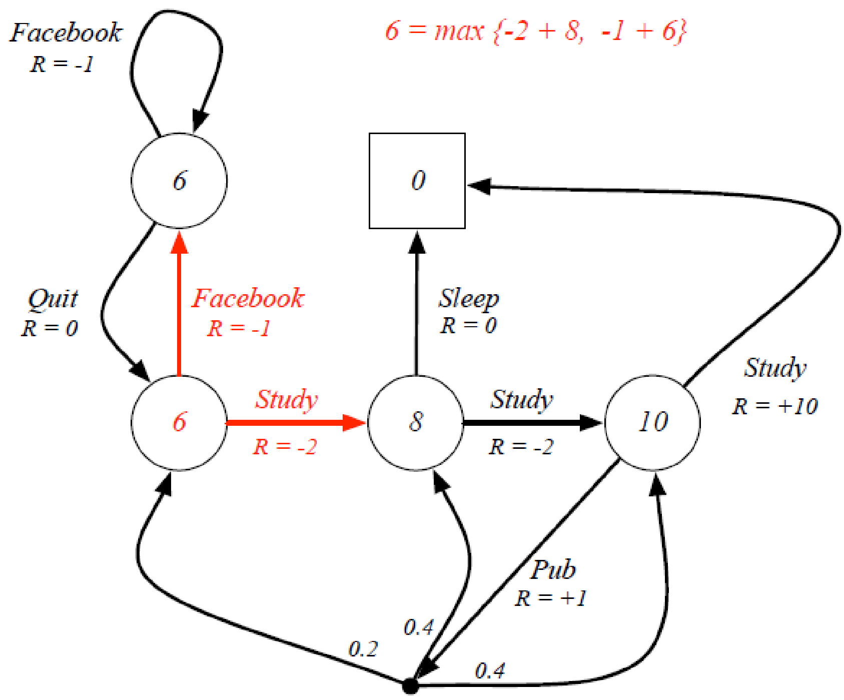


$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s',a')$$



$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} q_*(s',a')$$

학생 MDP 예: 최적화된 가치함수 V*



벨만 최적화 방정식 해

- 벨만 최적화 방정식은 비선형 함수
- 일반적인 해는 없음

- 많은 반복 해 방식 (iterative solution method)
  - 가치 반복 (Value Iteration)
  - 정책 반복 (Policy Iteration)
  - Q-러닝 (Q-learning)
  - Sarsa

참고 자료

❑ David Silver - UCL Course on RL, 2015

- http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
- Lecture 2: Markov Decision Processes

❑ RL (강화학습) 기초

- http://daeson.tistory.com/m/category/Reinforcement%20Learning
- 3. Markov Decision Processes (1)
- 4. Markov Decision Processes (2)