Markov Decision Processes

HaeJoong Lee

Sogang University, EE

2020.03.05

Markov Decision Processes

- ☐ **Markov Process**
- ☐ **Markov Reward Processes**
- ☐ **Markov Decision Processes**

Reward

☐ Rewards

- A **reward** R_t is a scalar vector feedback signal
- Indicates how well agent is dong at step t
- The agent's job is to maximise cumulative reward

Reinforcement learning is based on the reward hypothesis

Definition (Reward Hypothesis)

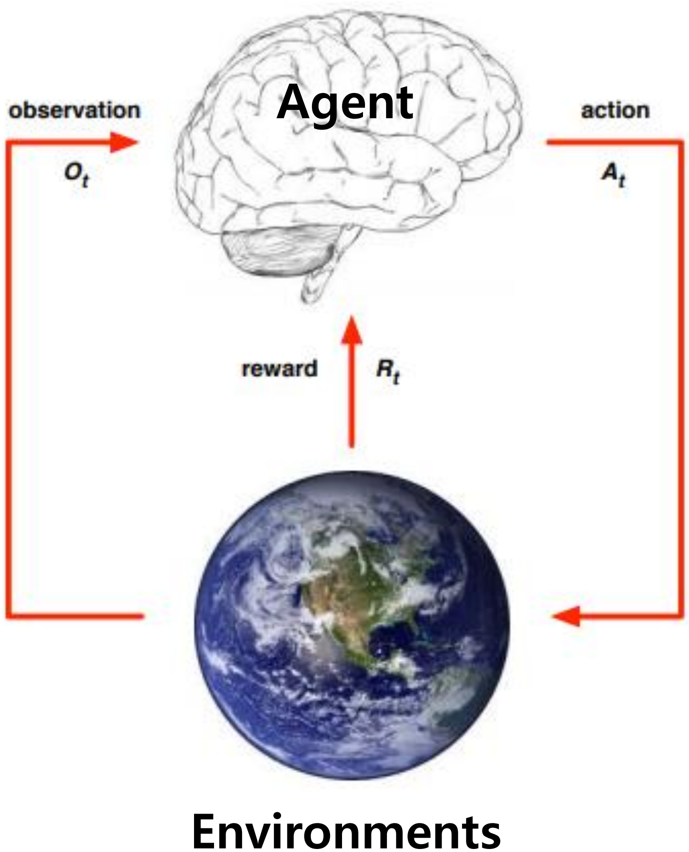
All goals can be described by the maximisation of expected cumulative reward

☐ Sequential Decision Making

- Goal : *select actions to maximise total future reward*
- Actions may have long term sequences
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward

Environments

☐ Agent and Environment



- At each step t **the agent**:
  - Executes actions A_t
  - Receives observation O_t
  - Receives scalar reward R_t

- **the environment**:
  - Receives actions A^t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}

- t increments at env. step

State

☐ History and State

- The history is the sequence of observations, actions, rewards

$$H_t = O_1, R_1, A_1, O_2, R_2, A_2, \dots, A_{t-1}, O_t, R_t$$

- **i.e.** all observable variables up to time t
- What happens next depends on the history:

  The agent selects actions
  The environment selects observations/rewards

- **State** is the information used to determine what happens next
- Formally, state is a function of the history:

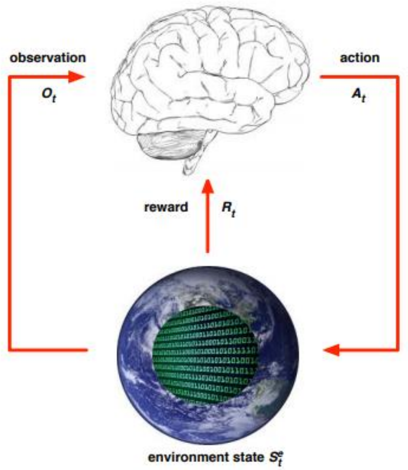
$$S_t = f(H_t)$$

NICELAB
Networking for Intelligence Communications and Energy

서강대학교 SOGANG UNIVERSITY

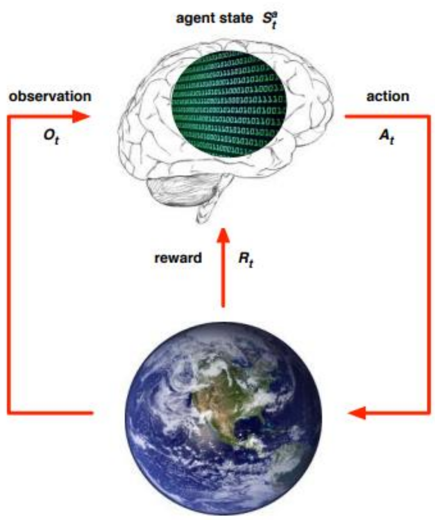
State

☐ Environment State



- The environment state S_t^e is the environment's private representation
- whatever data the environment uses to pick the next observation/reward
- The environment state is not usually visible to the agent
- Even if is S_t^e visible, it may contain irrelevant information

☐ Agent State



- The agent state S_t^a is the agent's internal representation
- whatever information the agent uses to pick the next action
- it is the information used by reinforcement learning algorithms
- It can be any function of history:

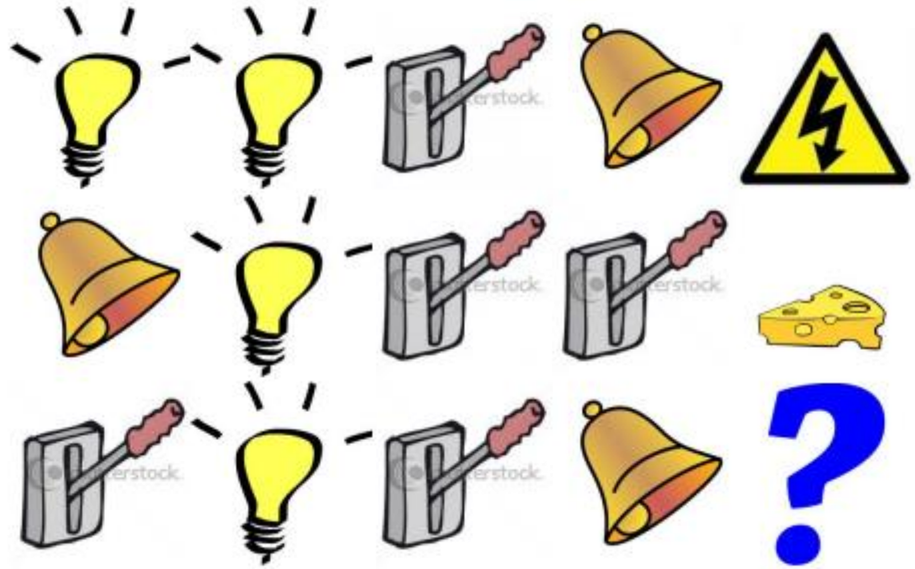
$$\boldsymbol{S_t^a = f(H_t)}$$

NICELAB
Networking for Intelligence Communications and Energy

서강대학교 SOGANG UNIVERSITY

State

Rat Example



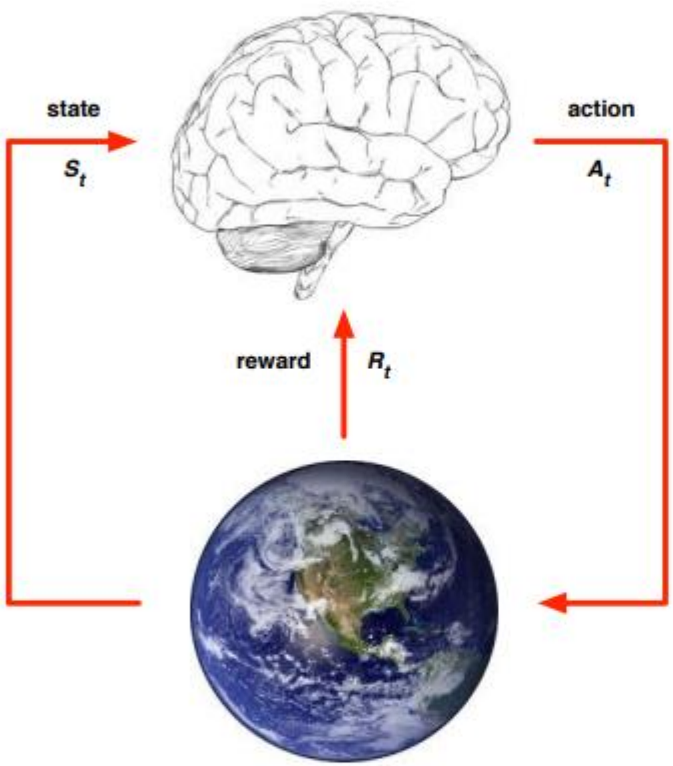
What if agent state = last 3 items in sequence?

What if agent state = counts for lights, bells and levers?

What if agent state = complete sequence?

State

☐ Fully Observable Environments



- Full observability: agent directly observes environment state

$$O_t = S_t^a = S_t^e$$

- Agent state = environment state = information state
- Formally, this is a **Markov decision process** (MDP)

NICELAB Networking for Intelligence Communications and Energy

서강대학교 Sogang University

Major Components of an RL Agent

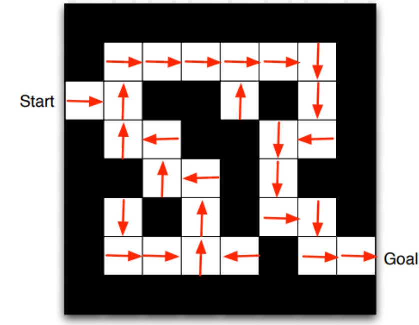
☐ **An RL agent may include one or more of these components:**

- **Policy** : agent's behaviour function
- **Value function** : how good is each state and/or action
- **Model** : agent's representation of the environment

Inside An RL Agent

Policy

- A policy is the agent's behaviour
- It is a map from state to action, e.g.
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$

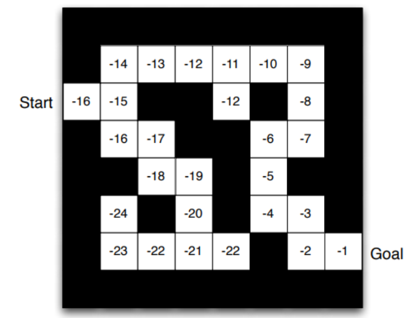


Arrows represent policy $\pi(s)$ for each state s

Value Function

- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- And therefore to select between actions, e.g.

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s \right]$$

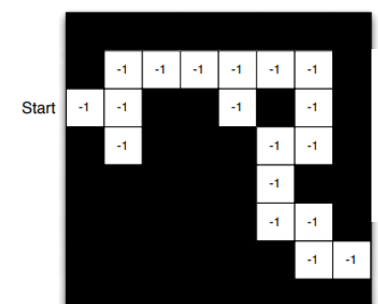


Numbers represent value $v_\pi(s)$ of each state s

Model

- A model predicts what the environment will do next
- $\mathcal{P}$ predicts the next state
- $\mathcal{R}$ predicts the next (immediate) reward, e.g.

$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$$
$$\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$



- Agent may have an internal model of the environment
- Dynamics: how actions change the state
- Rewards: how much reward from each state
- The model may be imperfect

Grid layout represents transition model $\mathcal{P}^a_{ss'}$

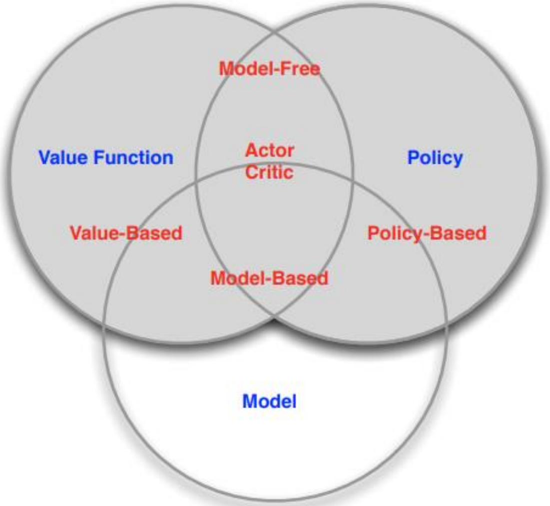
Numbers represent immediate reward $\mathcal{R}^a_s$ from each state s (same for all a)

Networking for Intelligence Communications and Energy

Categorizing RL agents

- Value Based
  - No Policy (Implicit)
  - Value Function
- Policy Based
  - Policy
  - No Value Function
- Actor Critic
  - Policy
  - Value Function

- Model Free
  - Policy and/or Value Function
  - No Model
- Model Based
  - Policy and/or Value Function
  - Model



Markov Decision Processes

- ☐ Markov Process
- ☐ Markov Reward Processes
- ☐ Markov Decision Processes
- ~~☐ Extensions to MDPs~~

NICELAB
Networking for Intelligence Communications and Energy

서강대학교
SOGANG UNIVERSITY

Markov Processes

☐ Markov Property

Definition

A state S_t is Markov *if and only if*

$$\boldsymbol{P}[S_{t+1} | S_t] = \boldsymbol{P}[S_{t+1} | S_1, \ldots, S_t]$$

- The state captures all relevant information from the history
- Once the state is known, the history may be thrown away
- i.e. The state is a sufficient statistic of the future

☐ State Transition Matrix

$$\boldsymbol{P}_{ss'}[S_{t+1} = s' | S_t = s]$$

$$\boldsymbol{P} = \begin{array}{cc} & \textbf{to} \\ \textbf{from} & \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} \end{array}$$

Each row of the matrix sums to 1

Markov Processes

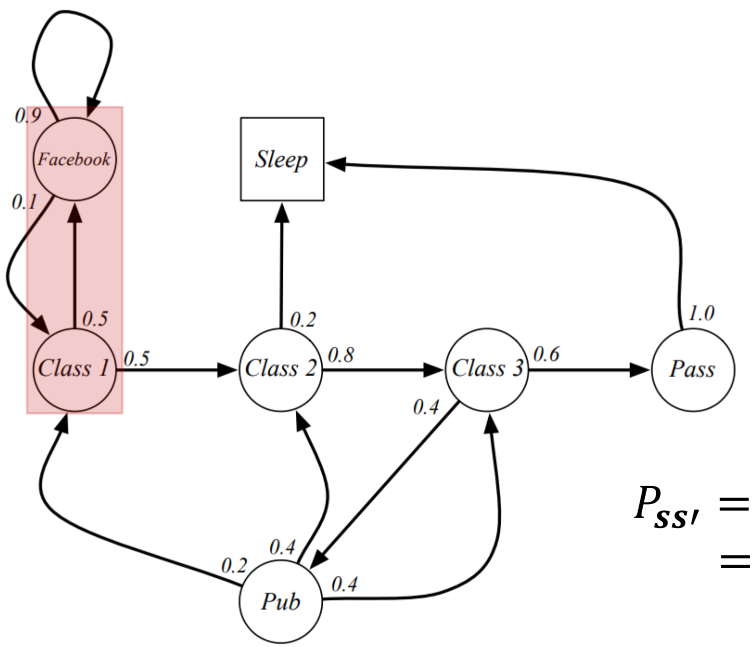
☐ Markov Process

Definition

A Markov Process(or Markov Chain) is a tuple $<S, P>$

- S is a (finite) set of states
- $P_{ss'} = \boldsymbol{P}\left[S_{t+1} = s' | S_t = s\right]$

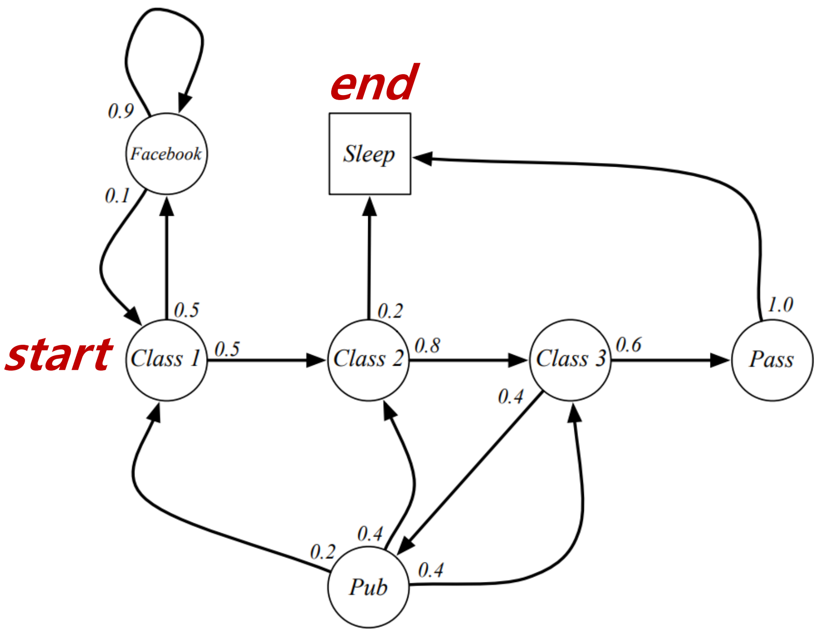
☐ Example

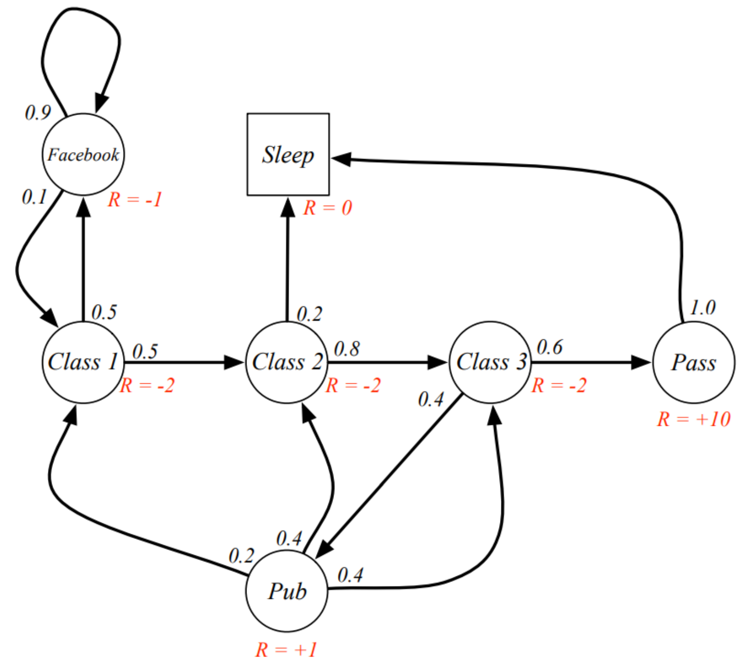


$$P_{ss'} = \boldsymbol{P}\left[S_{t+1} = \color{red}{Class\ 1} \mid S_t = \color{red}{Facebook}\right]$$
$$= 0.5$$

Markov Processes

☐ Example : Student Markov Chain Episodes





OV

☐ Transition Matrix

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

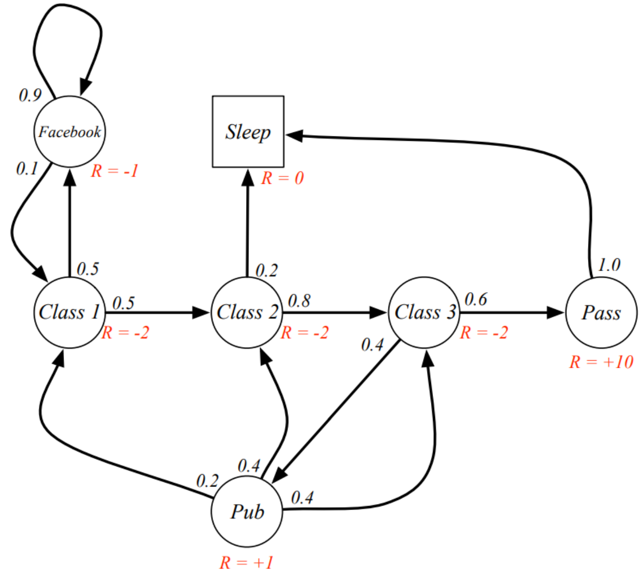
Markov Reward Processes

☐ **A Markov reward process is a Markov chain with values**

Definition

A Markov Reward Process is a tuple $<S, P, R, \gamma>$

- S is a finite set of states
- P is a state transition probability matrix,
  $P_{ss'} = \boldsymbol{P}\,[S_{t+1} = s' | S_t = s]$
- R is a reward function, $R_s = \mathrm{E}[R_{t+1} | S_t = s]$
- γ is a discount factor, $\gamma \in [0, 1]$



Markov Reward Processes

Return

Definition

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

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

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

Value Function

Definition

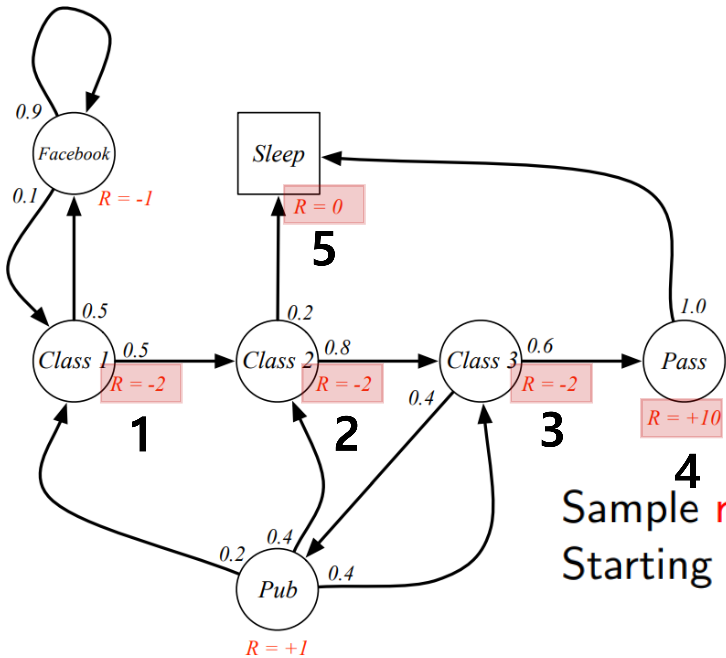
The *state value function* $v(s)$ of an MRP is the expected return starting from state s

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

The value function gives $v(s)$ the long-term value of state s

Markov Reward Processes

Returns



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

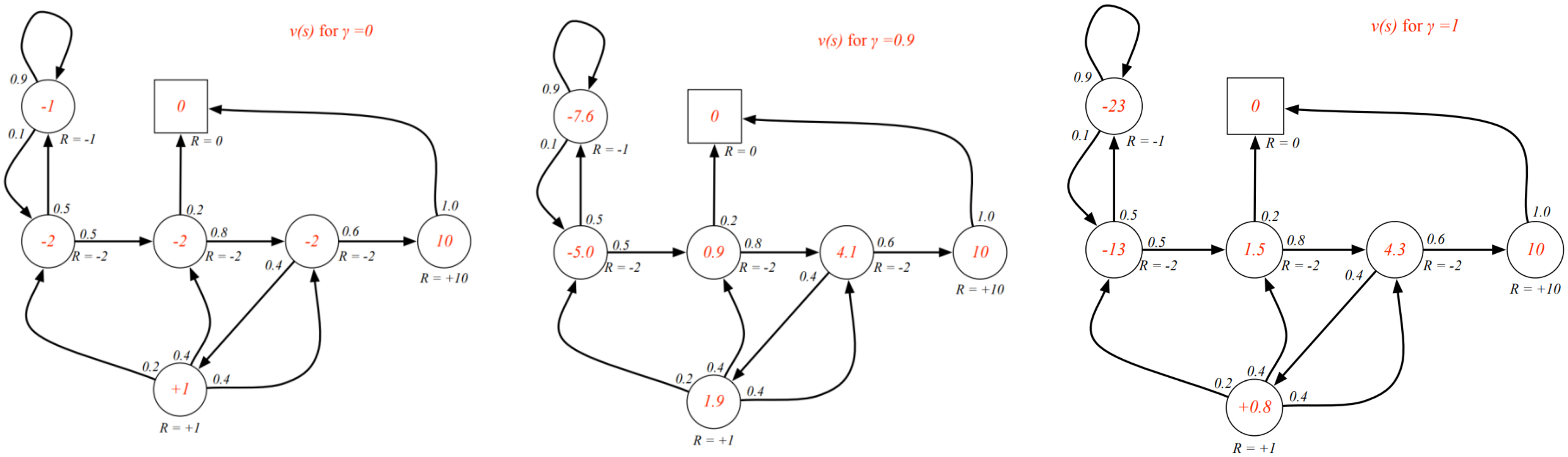
Sample returns for Student MRP:
Starting from $S_1 = C1$ with $\gamma = \frac{1}{2}$

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

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			

Markov Reward Processes

☐ Example : Returns by γ



Markov Reward Processes

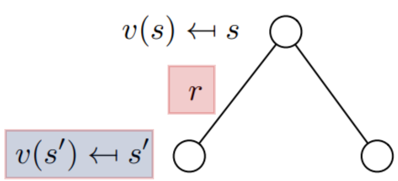
☐ Bellman Equation for MRPs

The value function can be decomposes into two parts:

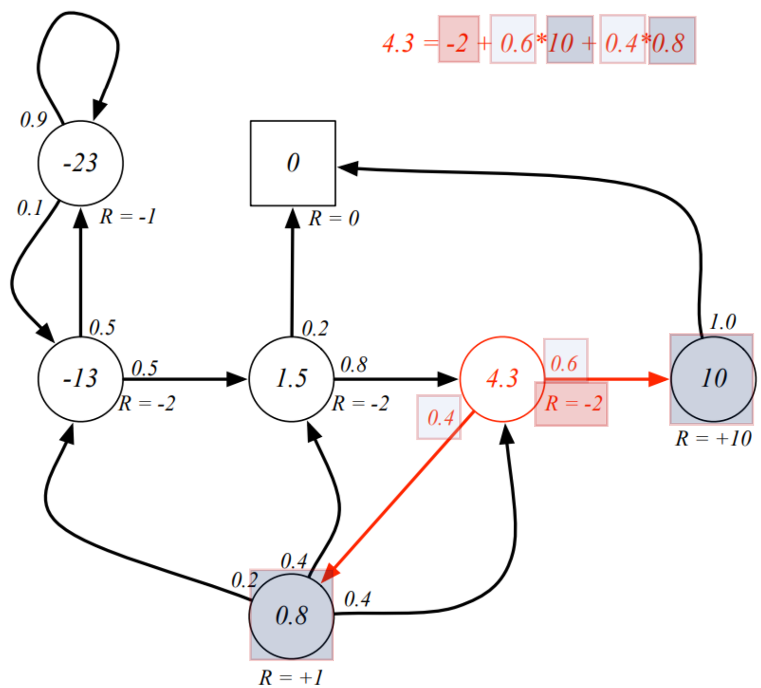
- Immediate reward R_{t+1}
- Discounted value of successor state $\gamma\, v(S_{t+1})$

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

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



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



Markov Reward Processes

☐ Bellman Equation in Matrix Form

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

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

☐ Solving the Bellman Equation

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

Big O : O(n^3)
Only small MRPs
There are many iterative methods for large MRPs, e.g.

- Dynamic programming
- Monte-Carlo evaluation
- Temporal-Difference learning

Markov Decision Processes

☐ **MDP is a MRP with decisions(actions).**

Definition

A Markov Decision Process is a tuple $< S, A, P, R, \gamma >$

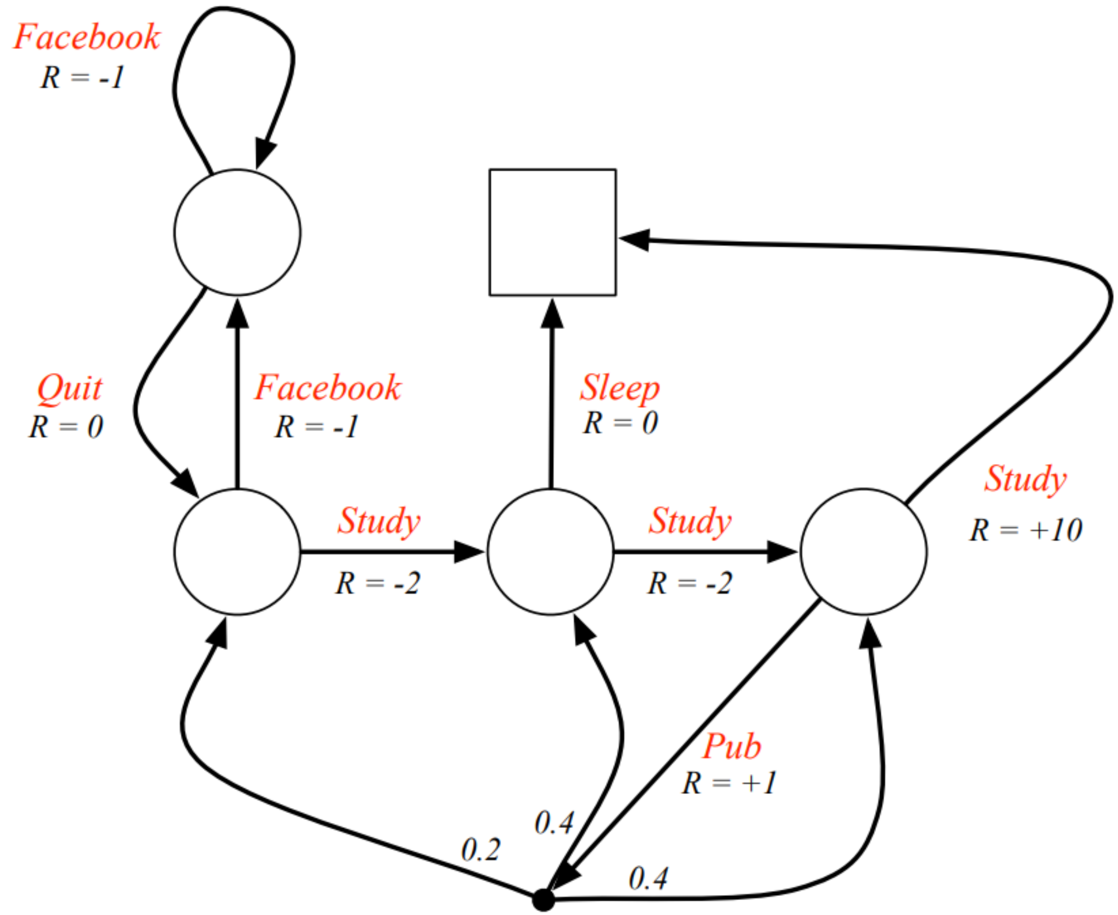
- S is a finite set of states
- A is a finite set of actions
- P is a state transition probability matrix,
  $P_{ss'}^{a} = \boldsymbol{P}\,[S_{t+1} = s' | S_t = s, A_t = a]$
- R is a reward function, $R_s^{a} = \mathrm{E}[R_{t+1} | S_t = s, A_t = a]$
- γ is a discount factor, $\gamma \in [0, 1]$

NICELAB Networking for Intelligence Communications and Energy

서강대학교 SOGANG UNIVERSITY

Markov Decision Processes

- Example : Student MDP



Markov Decision Processes

☐ Policy

Definition

A Policy π is a distribution over actions given states,

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

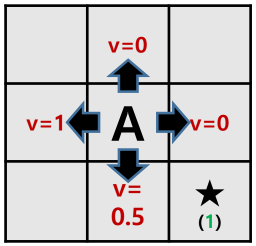
- A policy fully defines the behaviour of an agent
- MDP policies depend on the current state (not the history)
- i.e. Policies are *stationary* (time-independent),
  $A_t \sim \pi(\cdot|S_t), \forall t > 0$
- Given an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and a policy π
- The state sequence $S_1, S_2, \ldots$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^{\pi} \rangle$
- The state and reward sequence $S_1, R_2, S_2, \ldots$ is a Markov reward process $\langle \mathcal{S}, \mathcal{P}^{\pi}, \mathcal{R}^{\pi}, \gamma \rangle$
- where

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

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

Markov Decision Processes

☐ Example



If the current(A) value-function is 0,
Let's get it through the bellman equation

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

$$v_\pi(s) = \sum_{\mathrm{a} \in A} \pi(\mathrm{a}|s) 1 (R_{t+1} + 0.9 v_\pi(s'))$$

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

$$A_t = \{\text{left, right, up, down}\} = \frac{1}{4}$$

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

$$\gamma = 0.9$$

1	Action = left	$\frac{1}{4}$ x (0 + 0.9 x 1) = 0.225
2	Action = right	$\frac{1}{4}$ x (1 + 0.9 x 0) = 0.25
3	Action = up	$\frac{1}{4}$ x (0 + 0.9 x 0) = 0
4	Action = down	$\frac{1}{4}$ x (0 + 0.9 x 0.5) = 0.1125
Total	0.225 + 0.25 + 0 + 0.1125 = 0.5875	

Markov Decision Processes

Value function

Definition

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

$$v_\pi(s) = E_\pi[G_t |\ S_t = s]$$

Q-function (action-value function)

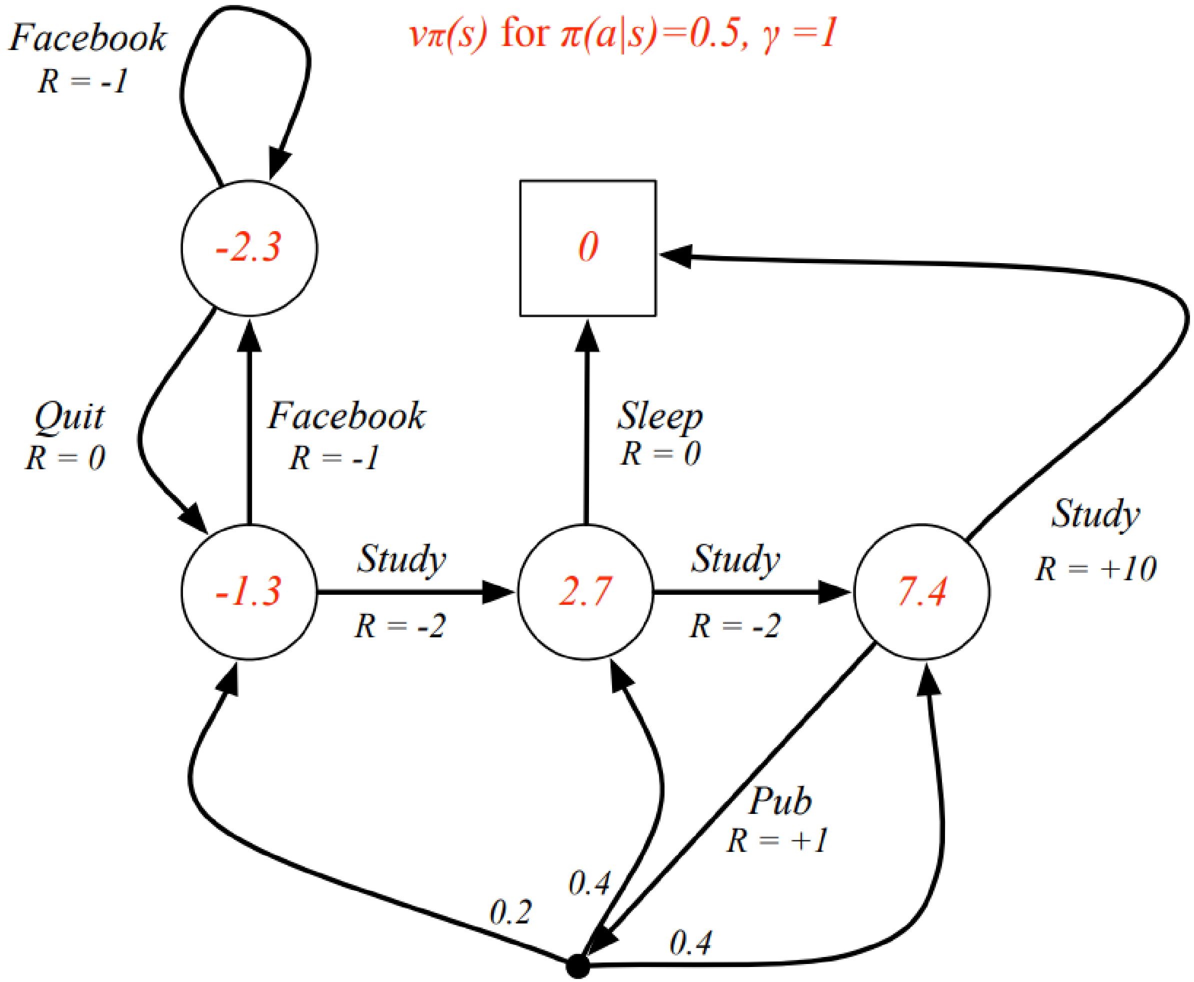
Definition

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

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

Markov Decision Processes

☐ **Example : State-Value Function for Student MDP**



Markov Decision Processes

☐ Bellman Expectation Equation

The state-value function

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

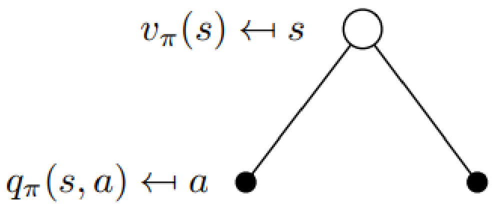
The action-value function

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

Markov Decision Processes

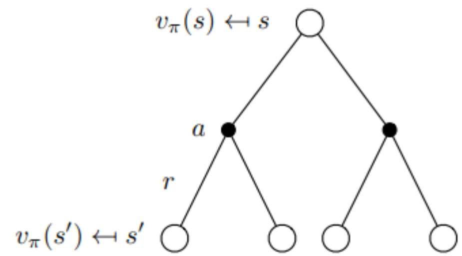
☐ Bellman Expectation Equation for V^{π}

①



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

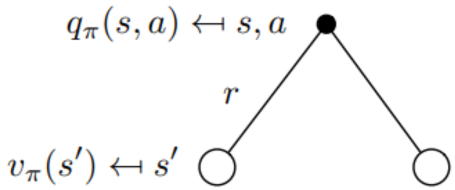
③



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

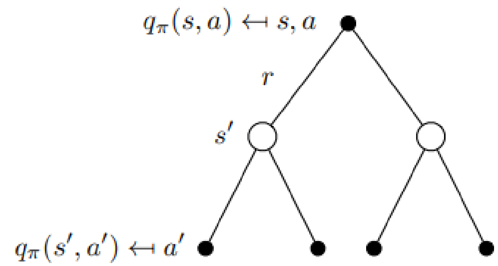
☐ Bellman Expectation Equation for Q^{π}

②



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

④



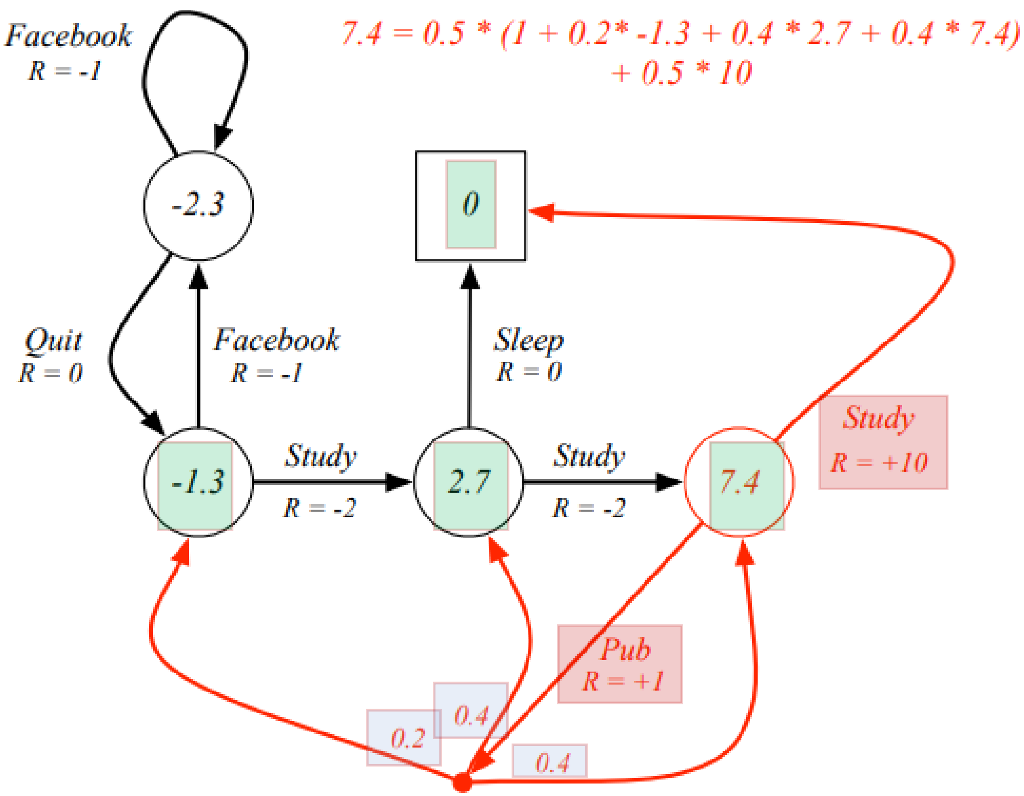
next

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

Markov Decision Processes

□ Example : State-Value Function for Student MDP

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')\right) = \frac{1}{2}*(10 + 1*[0]) + \frac{1}{2}*(1 + 1*[0.2*(-1.3) + 0.4*(2.7) + 0.4*(7.4)])$$



Markov Decision Processes

☐ Bellman Expectation Equation (Matrix Form)

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

$$\Downarrow$$

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

Markov Decision Processes

☐ Optimal Value Function

Definition

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

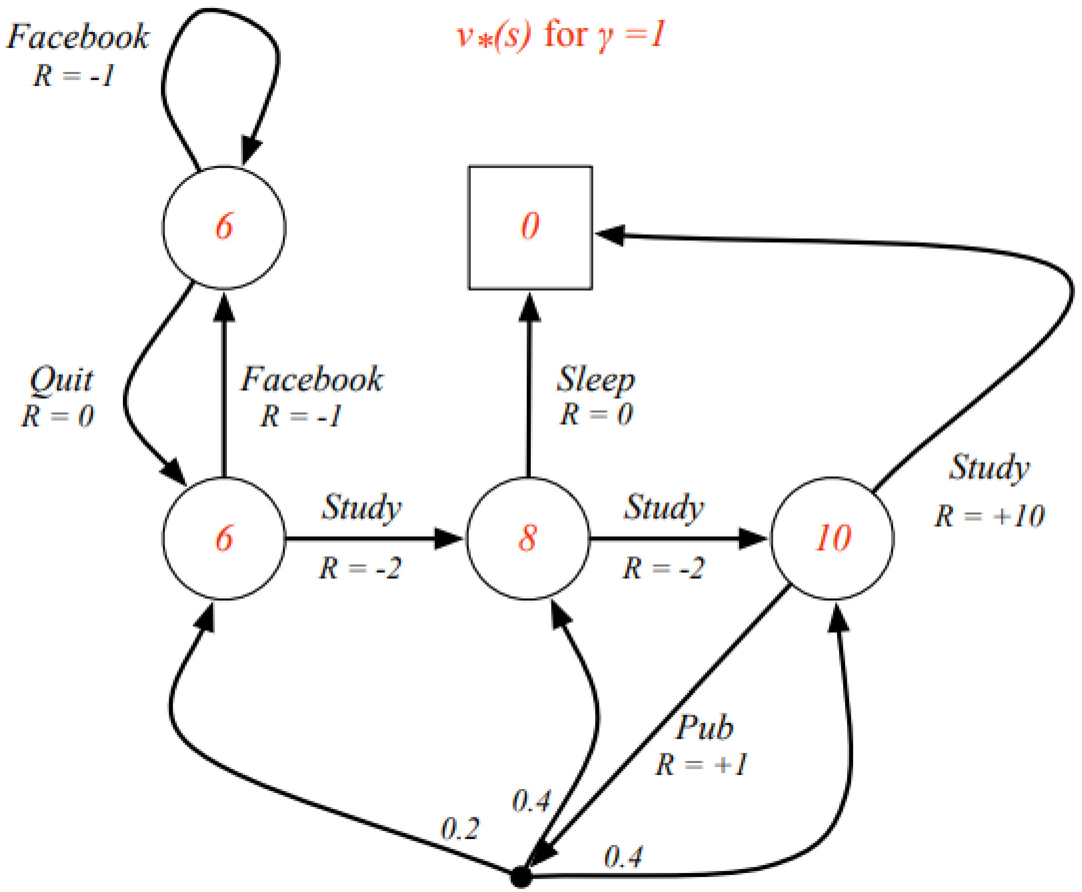
$$v_*(s) = \max_\pi v_\pi(s)$$

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

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

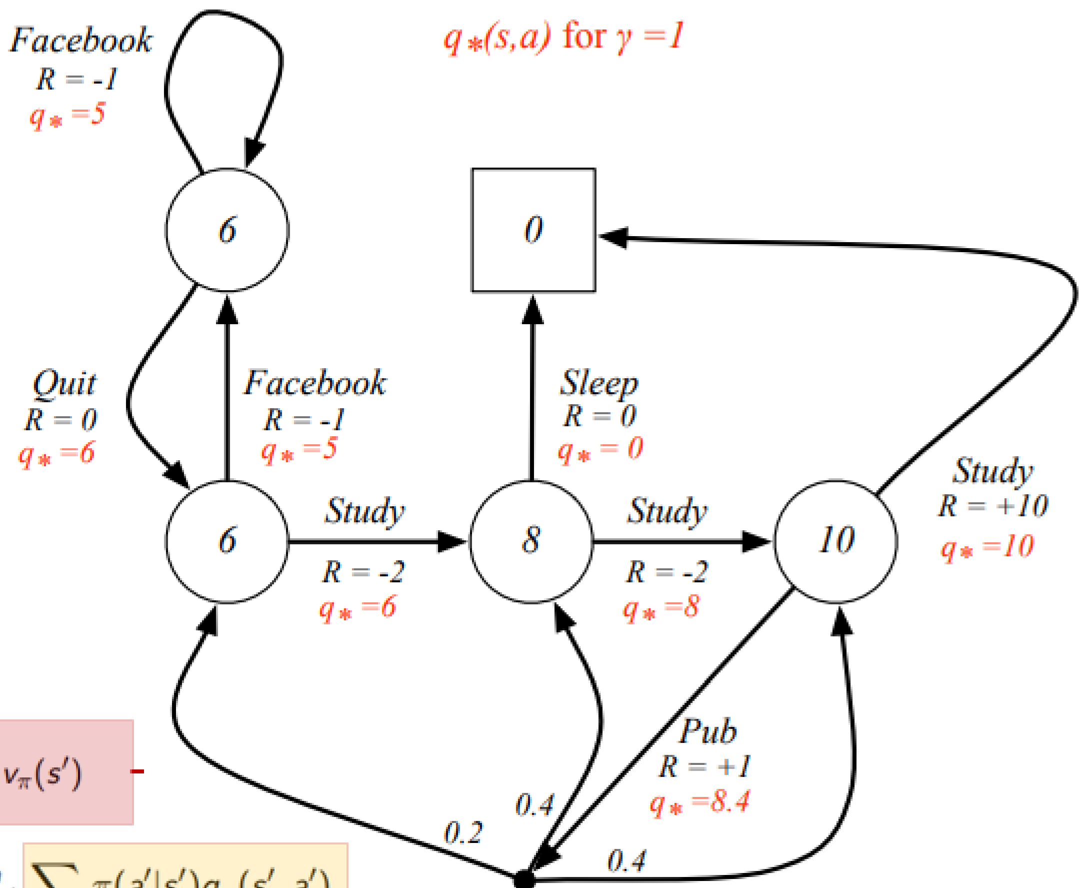
Markov Decision Processes

- Example : Optimal **Value** Function for Student MDP



Markov Decision Processes

☐ **Example : Optimal Action-Value Function for Student MDP**



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

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

NICELAB
Networking for Intelligence Communications and Energy

서강대학교 SOGANG UNIVERSITY

Markov Decision Processes

Optimal Policy

Define a partial ordering over policies

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

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

증명은 생략...

Finding an Optimal Policy

An optimal policy can be found by maximising over $q_*(s, a)$,

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

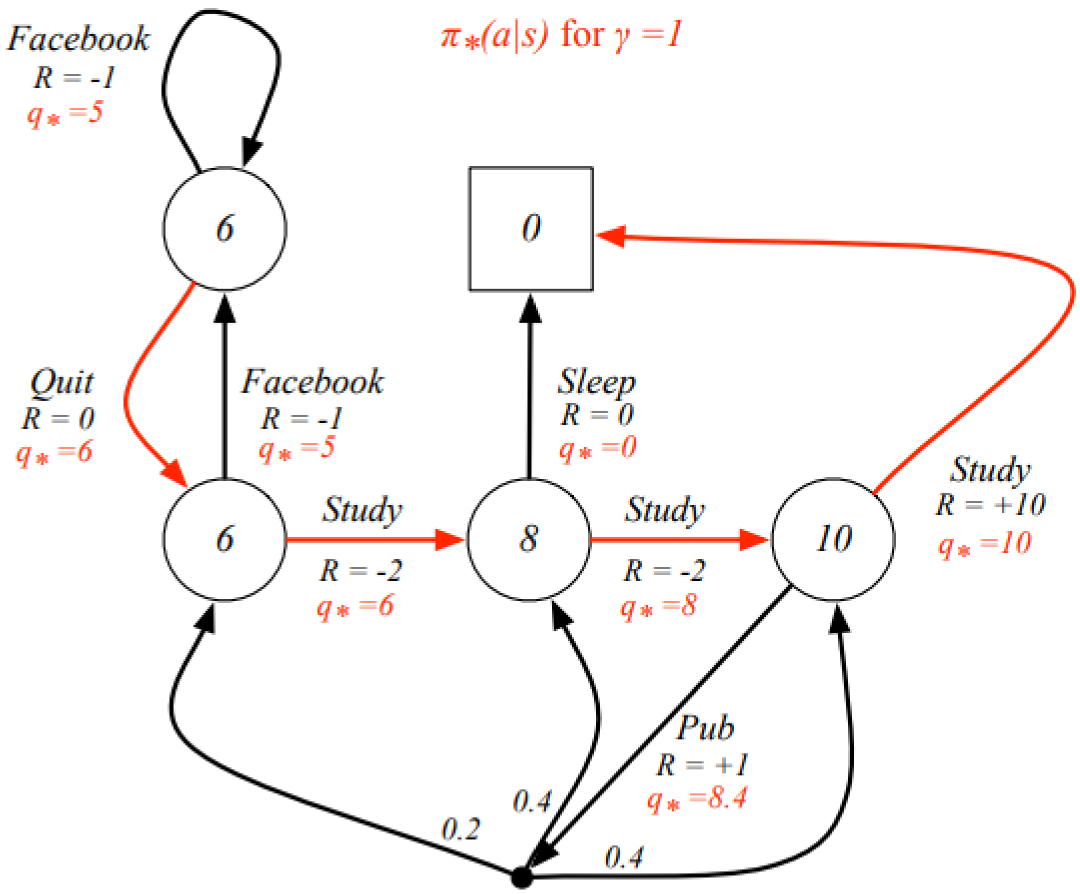
- There is always a deterministic optimal policy for any MDP
- If we know $q_*(s, a)$, we immediately have the optimal policy

NICELAB
Networking for Intelligence Communications and Energy

서강대학교 SOGANG UNIVERSITY

Markov Decision Processes

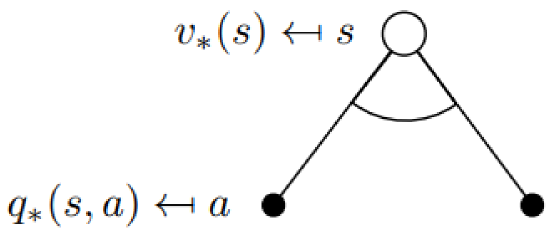
☐ Example: Optimal Policy for Student MDP



Markov Decision Processes

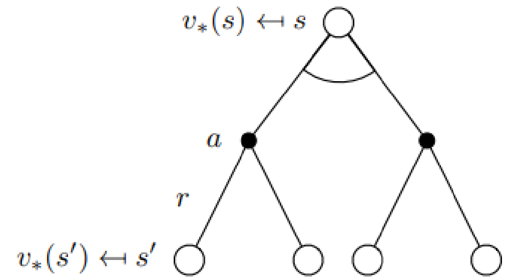
☐ Bellman Expectation Equation for V_*

①



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

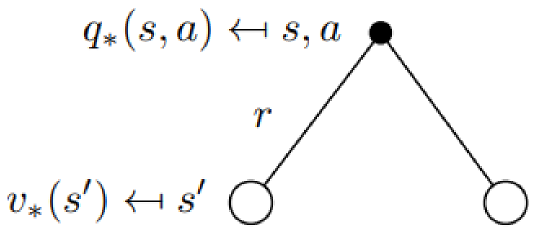
③



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

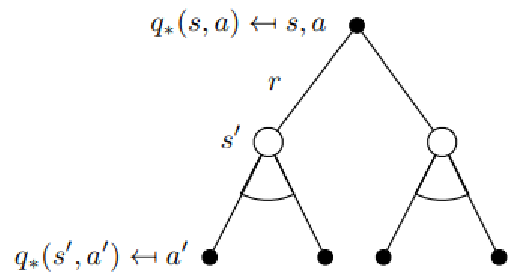
☐ Bellman Expectation Equation for Q_*

②



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

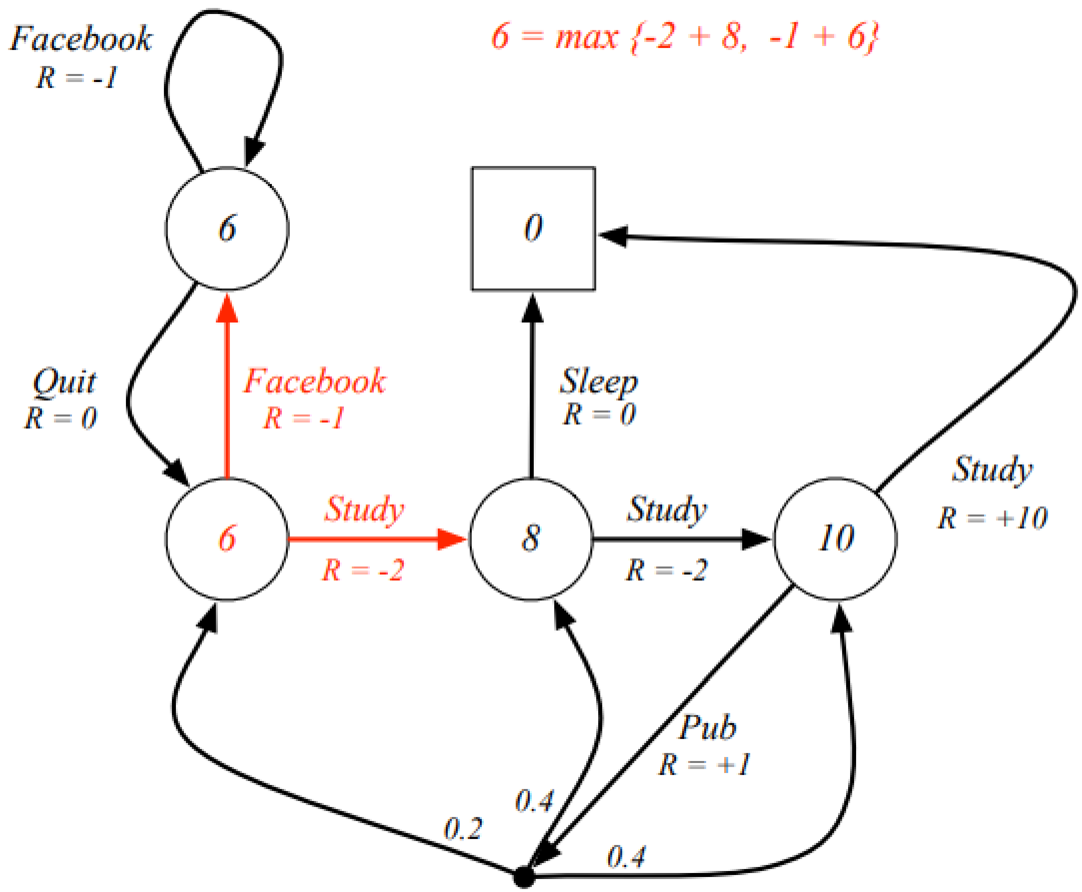
④



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

Markov Decision Processes

☐ Example: Bellman Optimality Equation in Student MDP



Markov Decision Processes

☐ Solving the Bellman Optimality Equation

- Bellman Optimality Equation is non-linear
- No closed form solution (in general)
- Many iterative solution methods
  - Value Iteration
  - Policy Iteration
  - Q-learning
  - Sarsa

Networking
Next

Intelligence
Innovative

Communications
Creative

Energy
Envisioning

NICELAB

Networking for Intelligence Communications and Energy