Reinforcement Learning: Markov Decision Process

AI/ML Teaching

Goals

- Brief concept of Reinforcement Learning (RL)
- Markov Decision Process (MDP) for RL
- Why model-free RL?

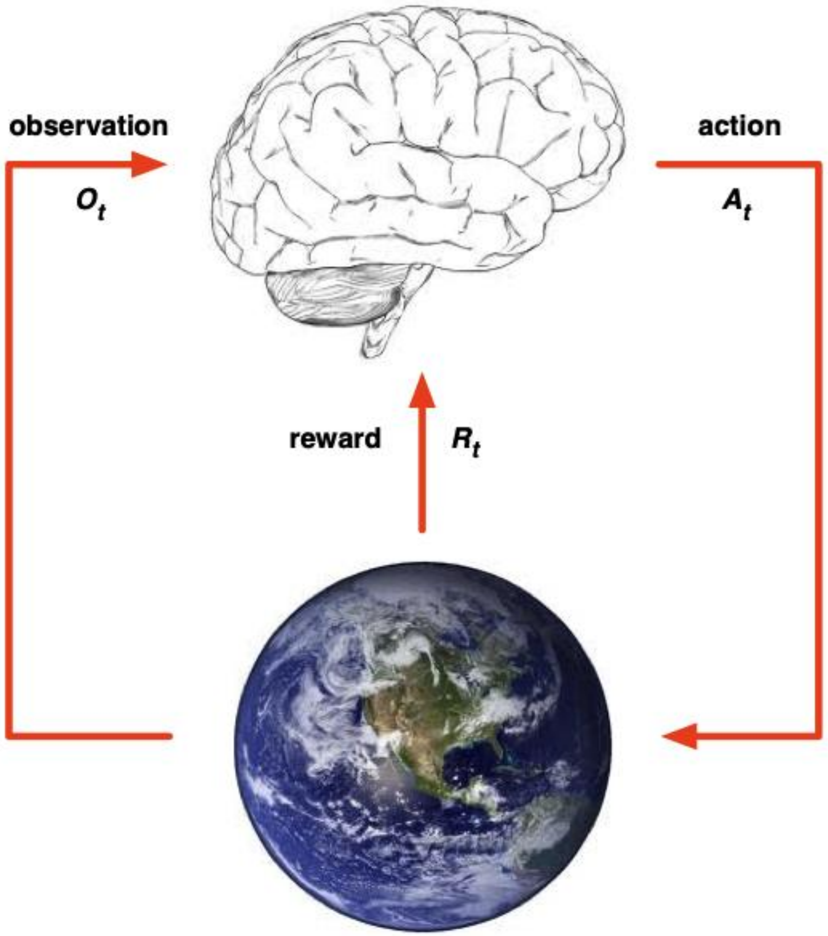
Reinforcement Learning (RL)

- No supervisor, only a reward
- Feedback can be delayed
- Agent’s actions affect the subsequent data it receives
- Example: AlphaGo
  - **Goal: select actions to maximize total future reward**
  - Actions may have long term consequences
  - Reward may be delayed
    - May be better to sacrifice immediate reward go gain more long-term reward

Agent and Environment

- Agent at step t
  - Receives observation O_t
  - Executes action A_t
  - Receives reward R_t

- Environment
  - Receives observation A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}



- State is a function of the history $(O_1, R_1, A_1, \ldots A_{t-1}, O_t, R_t)$

Markov Decision Process

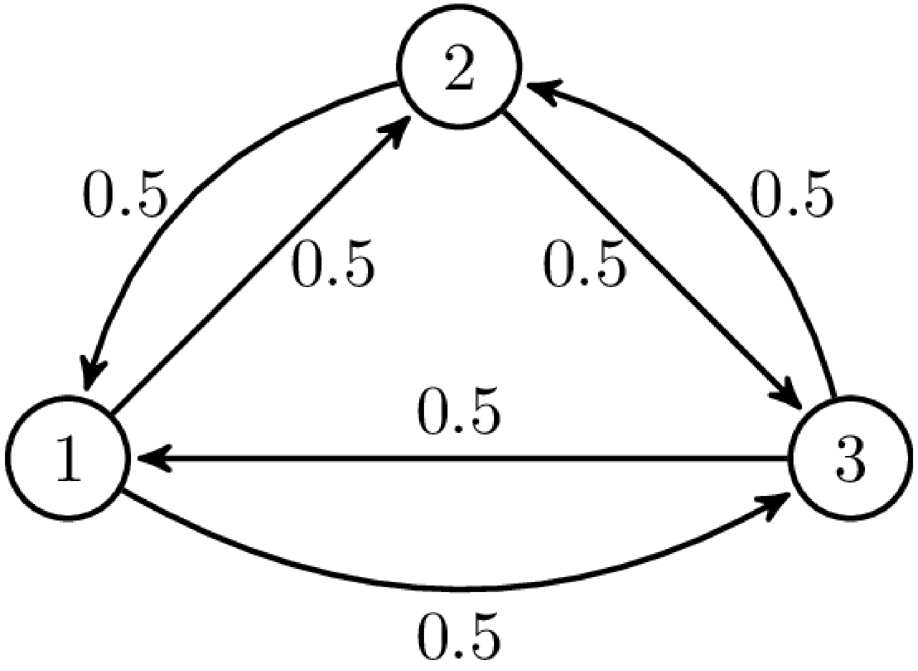
- Markov decision processes (MDPs) formally describe an environment for RL where the environment is fully observable
  - Real world is partially observable
  - Well-defined environment/state or summarized state → Markov approximation
  - Mathematically tractable

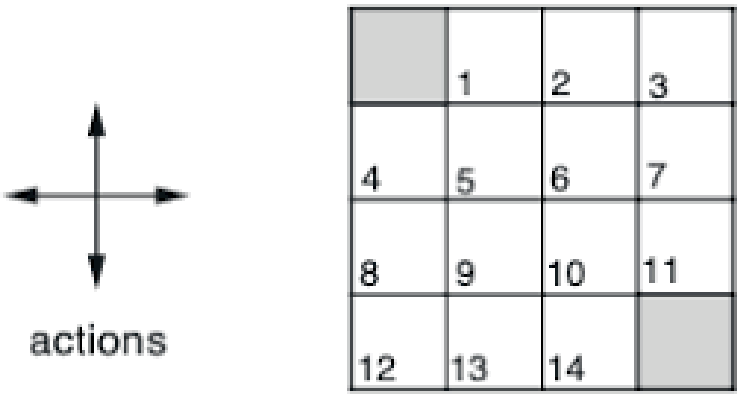
- Markov state: A state S_t is Markov if and only if

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

  - Future is independent of the past given the present
  - The state is a sufficient statistic of the future

Markov State Example





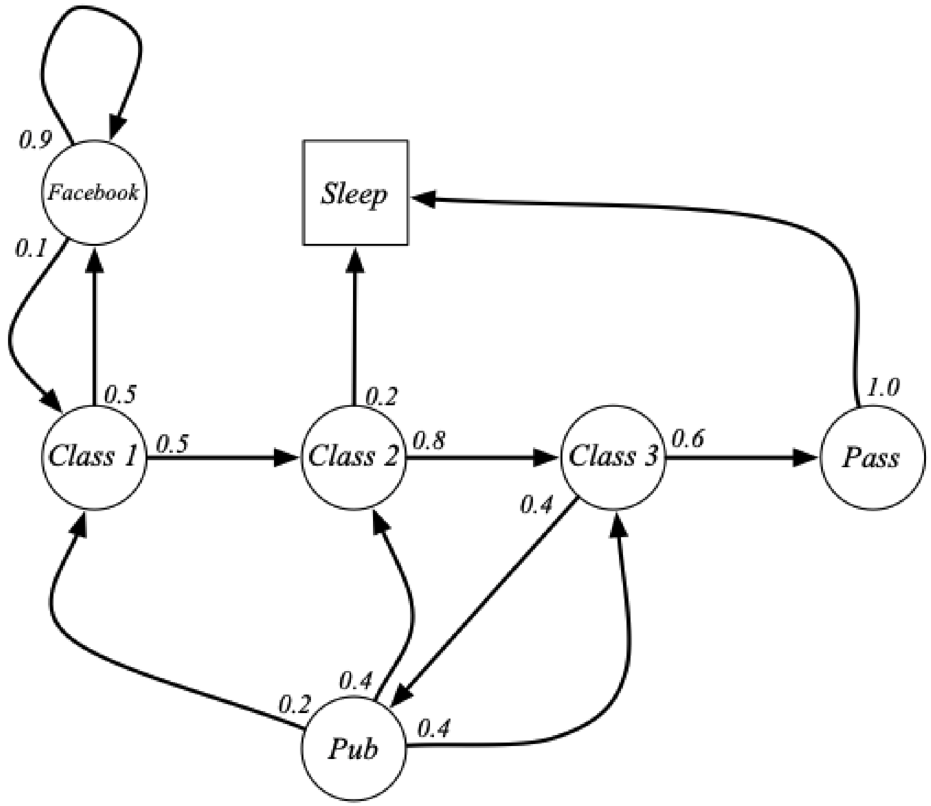
Markov Process (or Markov Chain)

- Markov Process is a memoryless random process
  - Tuple $\langle \mathcal{S}, \mathcal{P} \rangle$
    - $\mathcal{S}$: set of state $(S_1, S_2, \dots)$
    - $\mathcal{P}$: state transition probability matrix

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

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

Markov Chain Example



$\mathcal{P} =$

	C1	C2	C3	Pass	Pub	FB	Sleep
C1		0.5				0.5	
C2			0.8				0.2
C3				0.6	0.4		
Pass							1.0
Pub	0.2	0.4	0.4				
FB	0.1					0.9	
Sleep							1

Markov Reward Process

- Markov chain with values
- Markov Reward Process: Tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$
  - $\mathcal{S}$: set of state $(S_1, S_2, \ldots)$
  - $\mathcal{P}$: state transition probability matrix
  - $\mathcal{R}$: reward function ($\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$)
  - γ: discount factor

- Return $G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
  - γ close to 0 leads to myopic evaluation
  - γ close to 0 leads to far-sighted evaluation

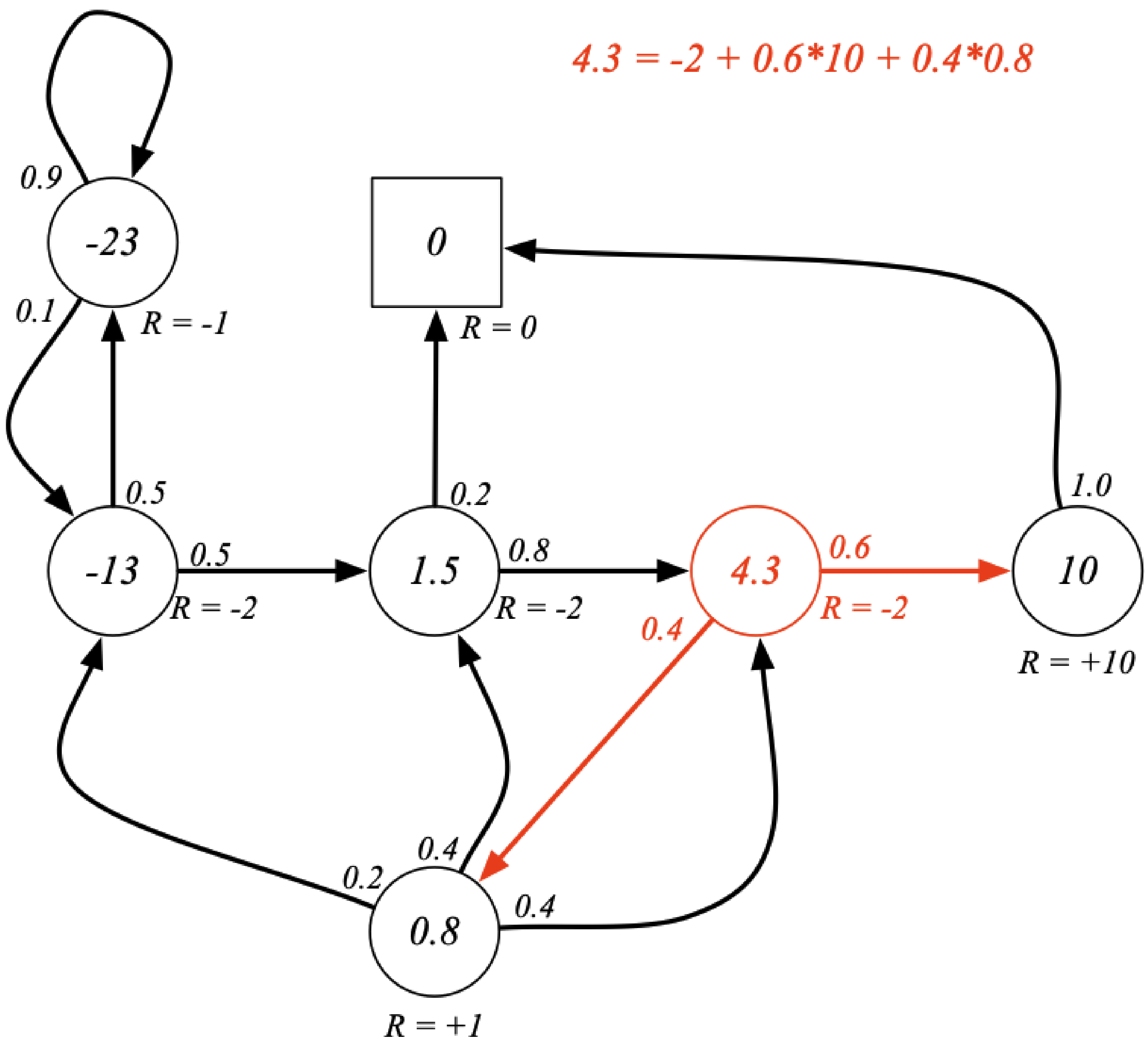
Value function

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

$$v(s) = \mathbb{E}[G_t | S_t = s]$$

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

Bellman equation



4.3 = -2 + 0.6*10 + 0.4*0.8
0.9
-23
R = -1
0.1
0
R = 0
0.5
0.2
1.0
-13
0.5
R = -2
1.5
0.8
R = -2
4.3
0.6
R = -2
0.4
10
R = +10
0.4
0.2
0.8
0.4
R = +1

Bellman Equation

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

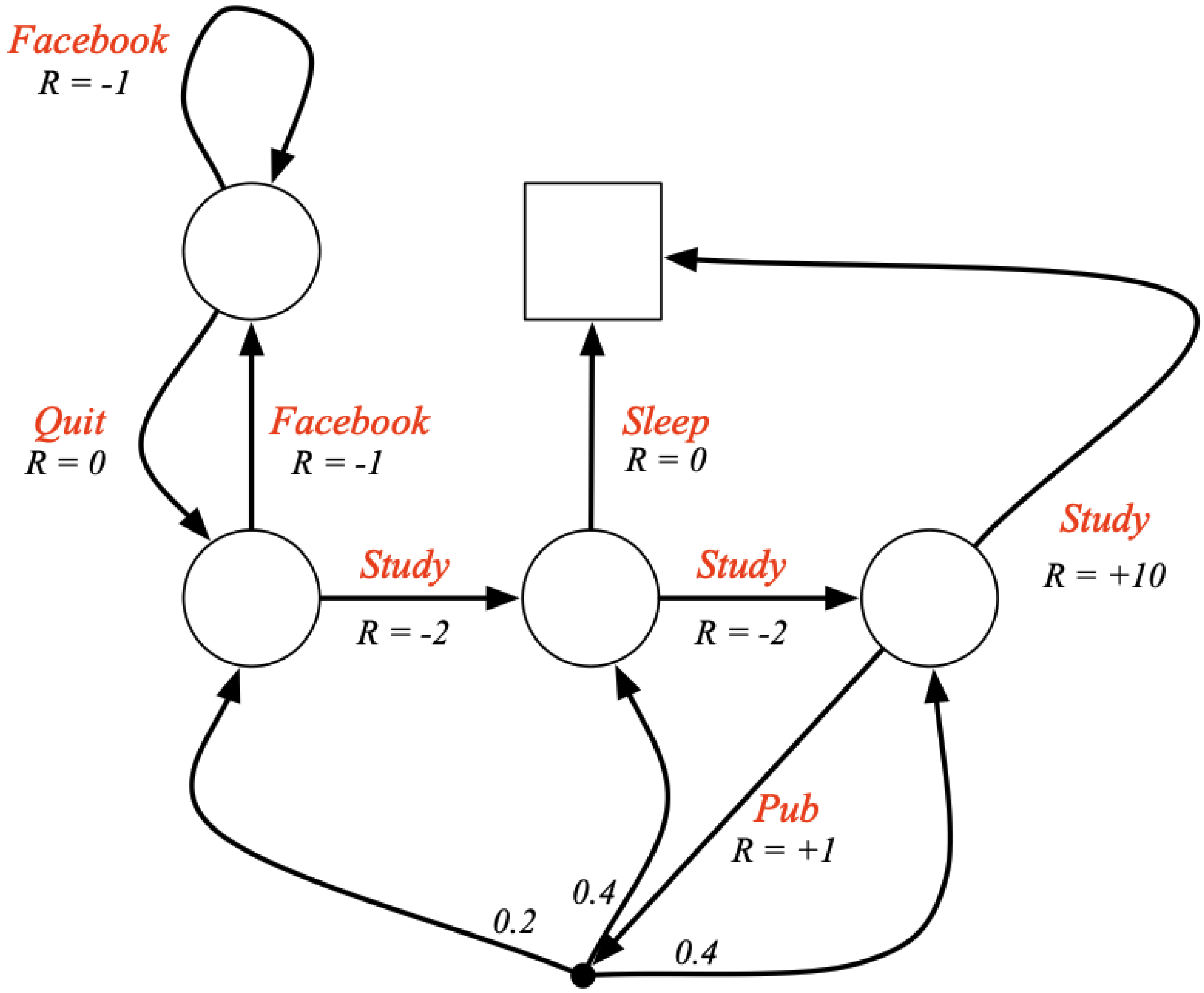
where v is a column vector with one entry per state

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

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

Markov Decision Process

- Markov chain with values
- Markov Reward Process: Tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$
  - $\mathcal{S}$: set of state $(S_1, S_2, \ldots)$
  - $\mathcal{A}$: set of actions
  - $\mathcal{P}$: state transition probability matrix
    - $\mathcal{P}_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$
  - $\mathcal{R}$: reward function $(\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a])$
  - γ: discount factor



Facebook
R = -1
Quit
R = 0
Facebook
R = -1
Sleep
R = 0
Study
R = -2
Study
R = -2
Study
R = +10
Pub
R = +1
0.2
0.4
0.4

Policies

A policy π is a distribution over actions given states,

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

- 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 $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$
  - $\mathcal{P}^\pi_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{ss'}$
  - $\mathcal{R}^\pi_s = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^a_s$

Value function

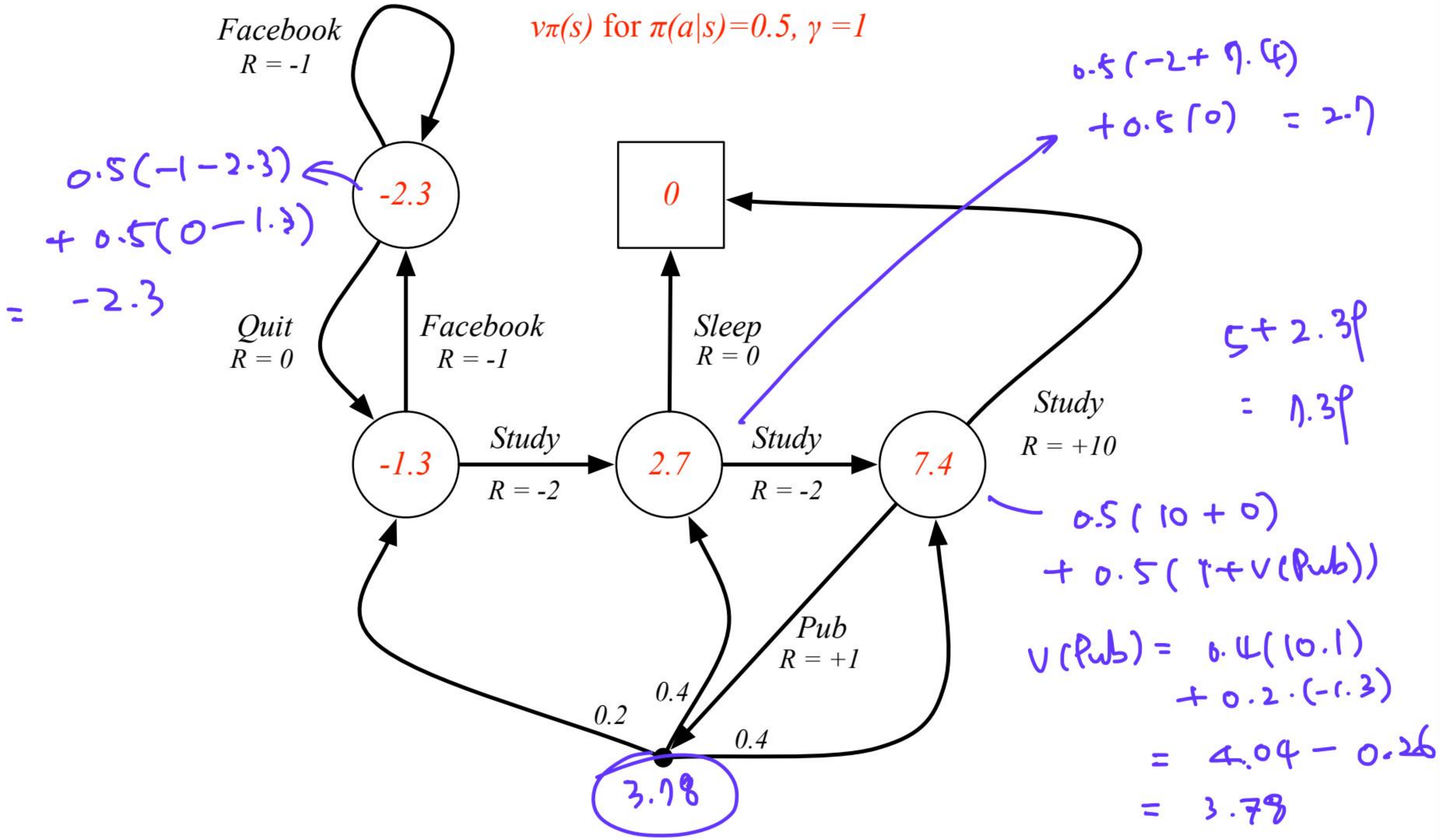
- State-value function

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

- Action-value function

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

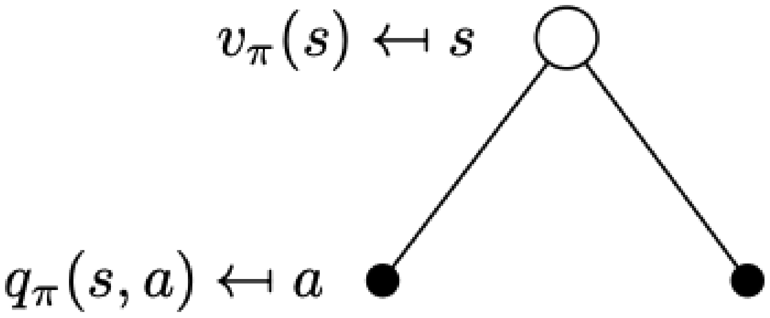
$v_\pi(s)$ for $\pi(a|s)=0.5$, $\gamma = 1$

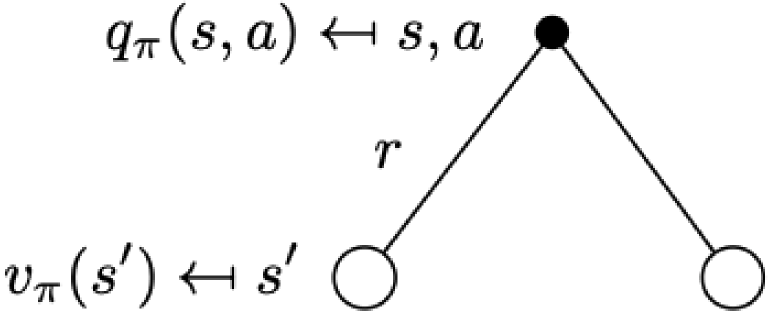


Bellman Expectation Equation

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

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

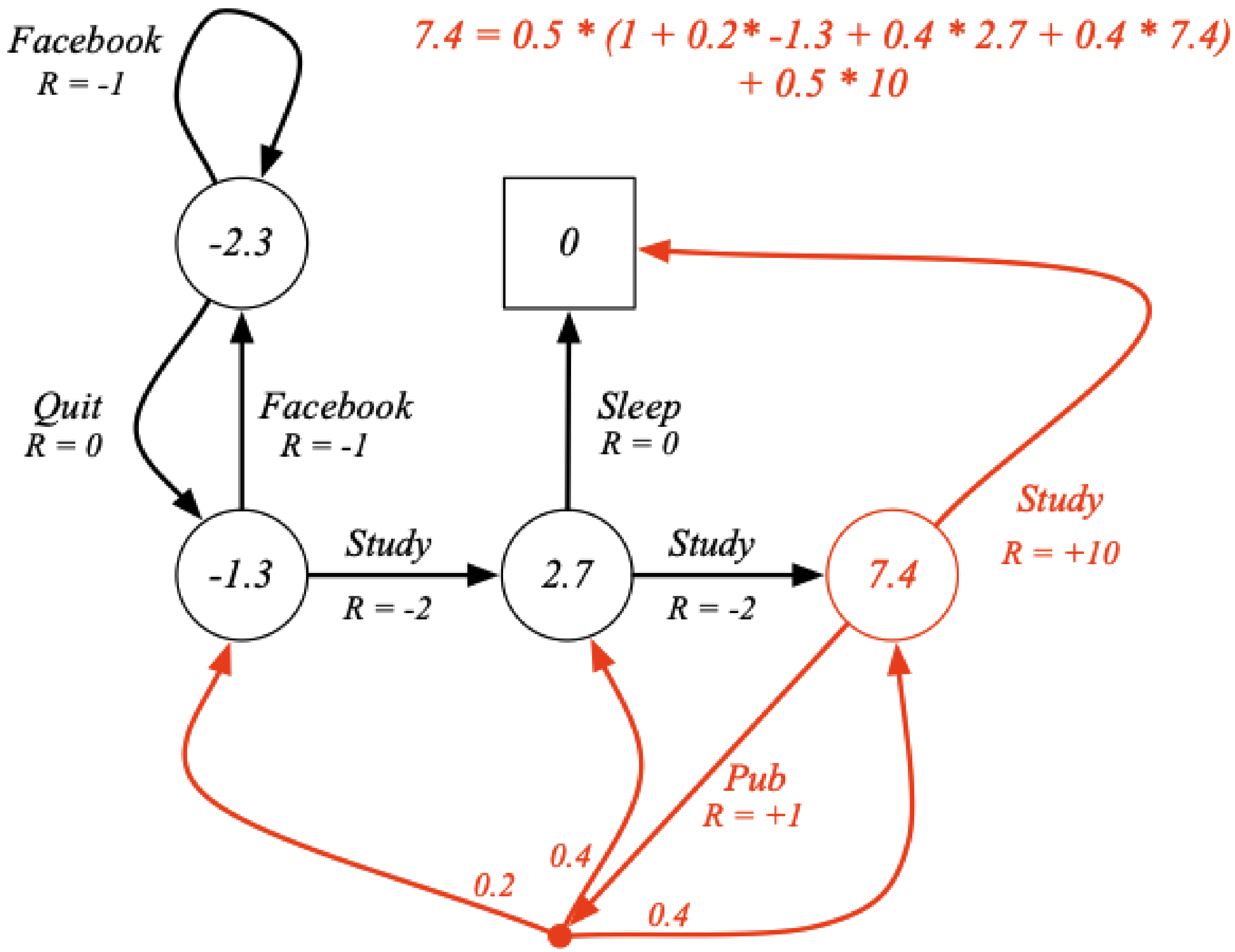




$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

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

Bellman Expectation Equation



Optimal Value Function

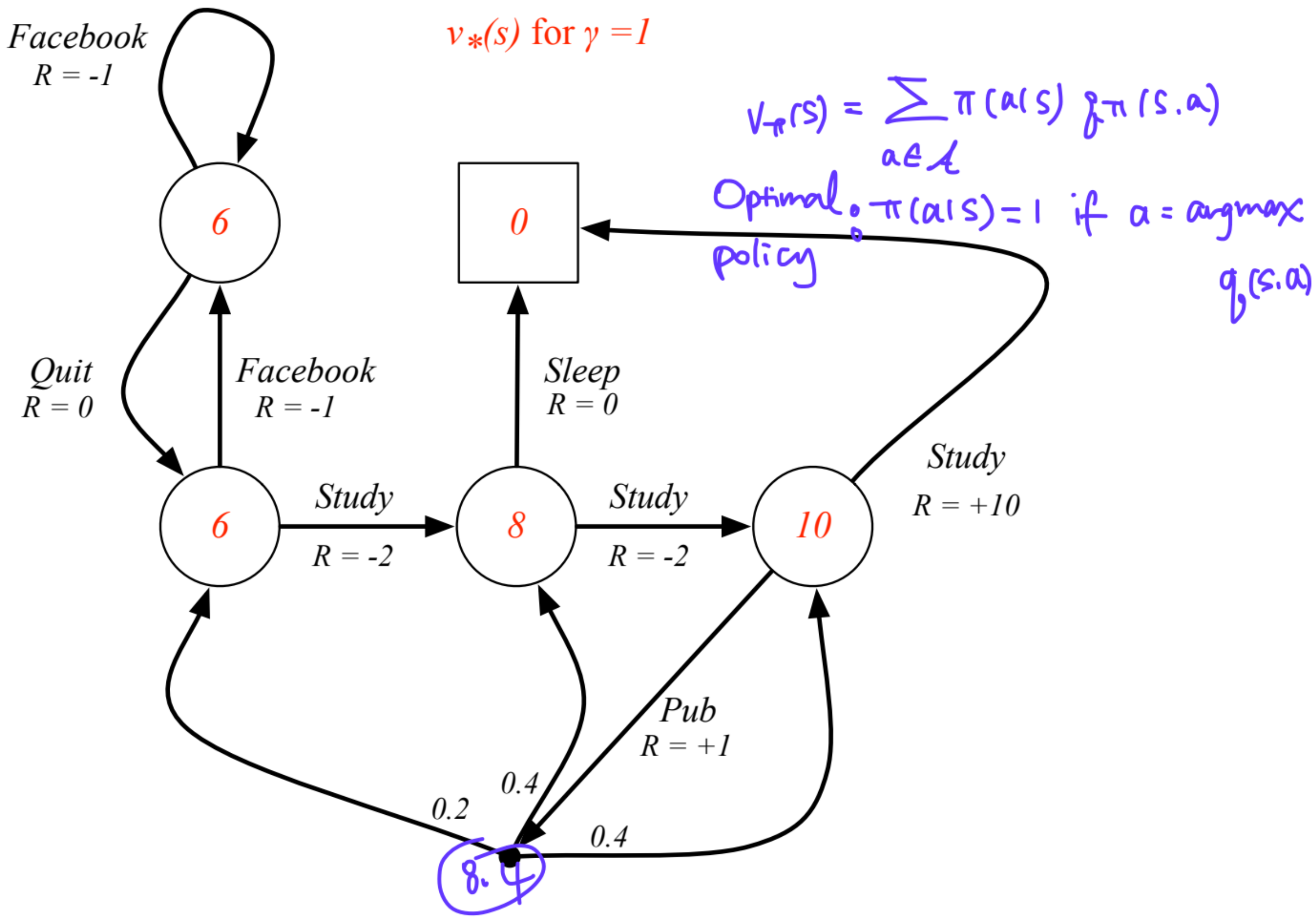
The *optimal state-value function* $v_*(s)$ is the maximum value function over all policies

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

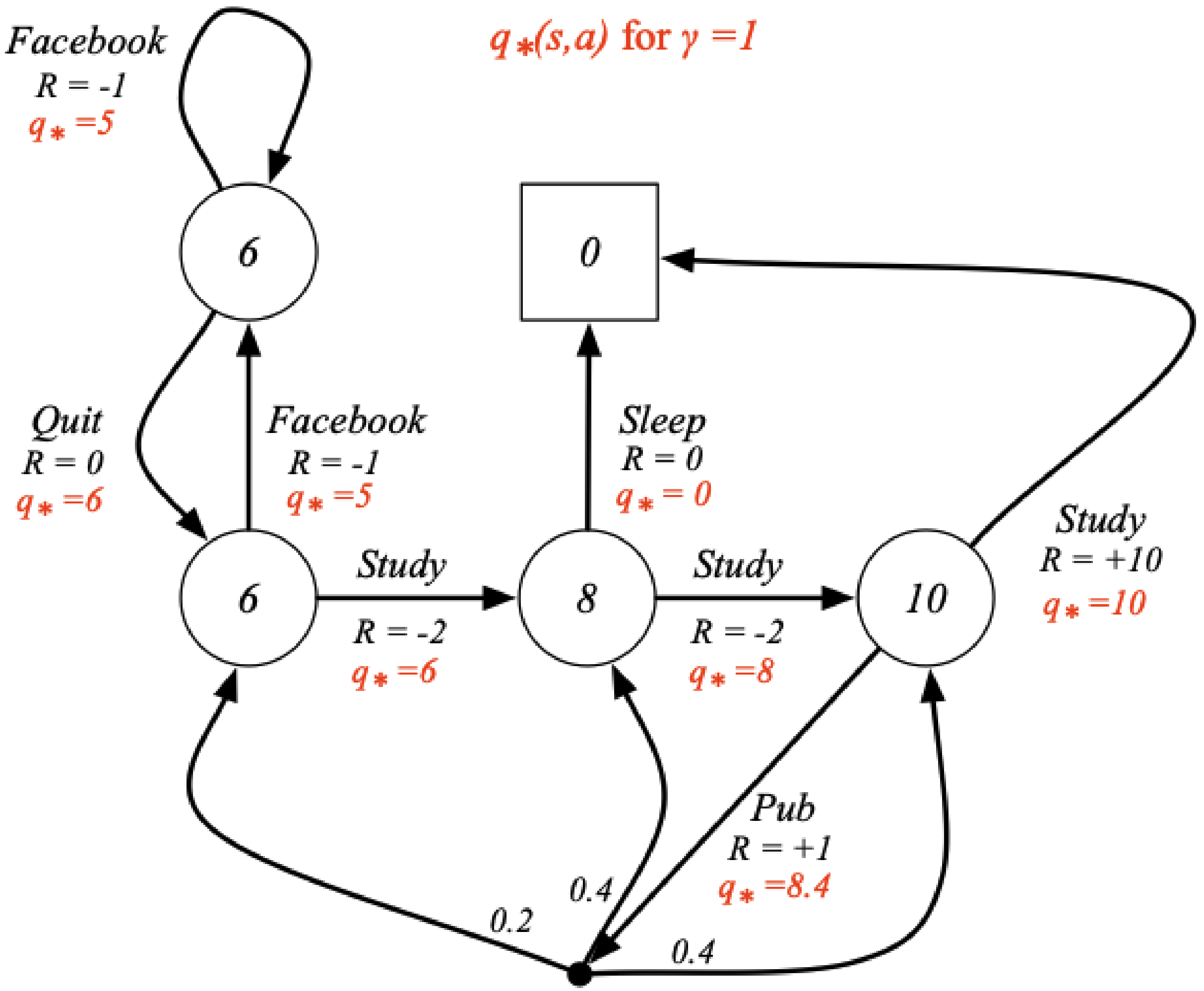
The *optimal action-value function* $q_*(s, a)$ is the maximum action-value function over all policies

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

Optimal Value Function

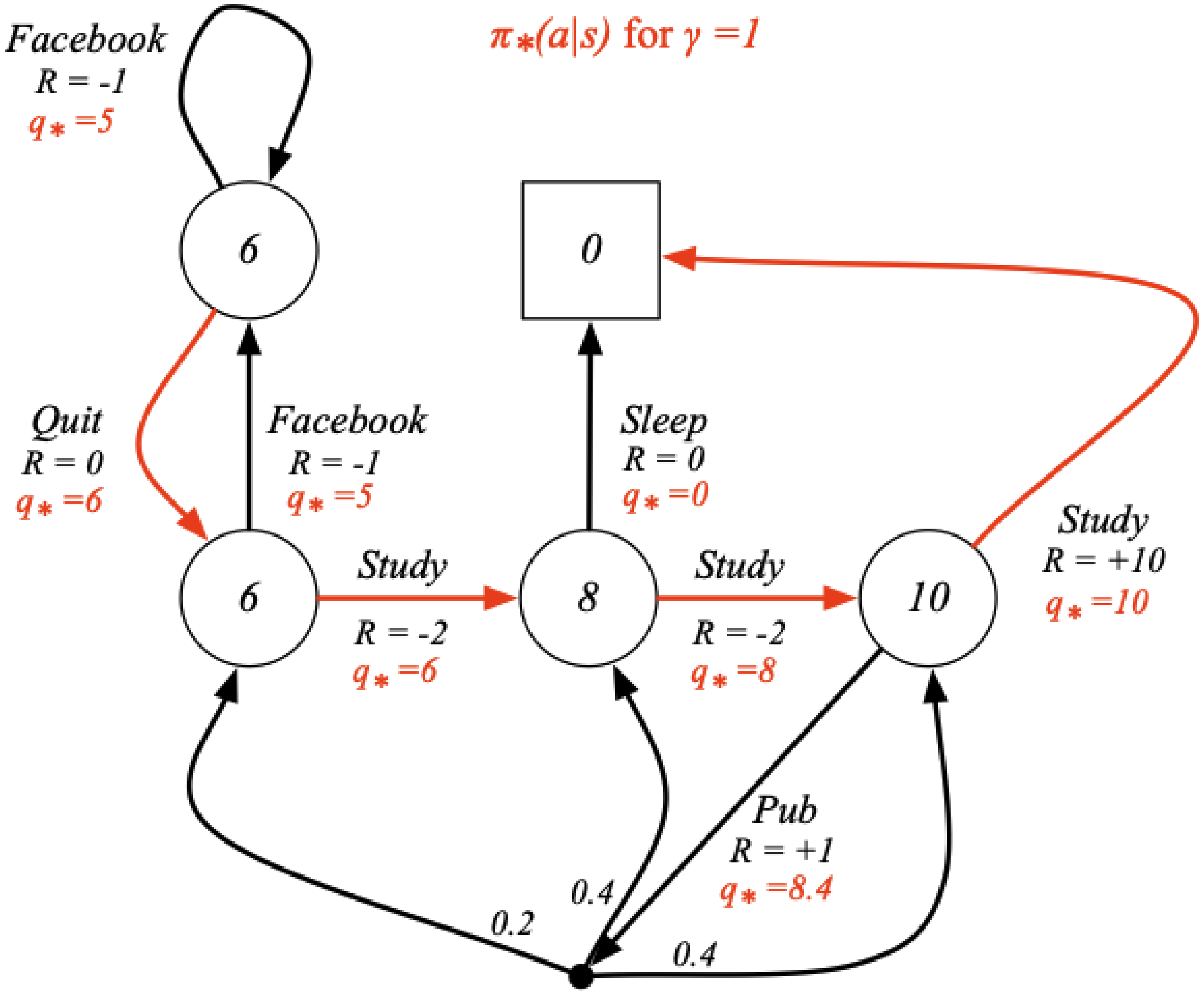


Optimal Action-Value Function

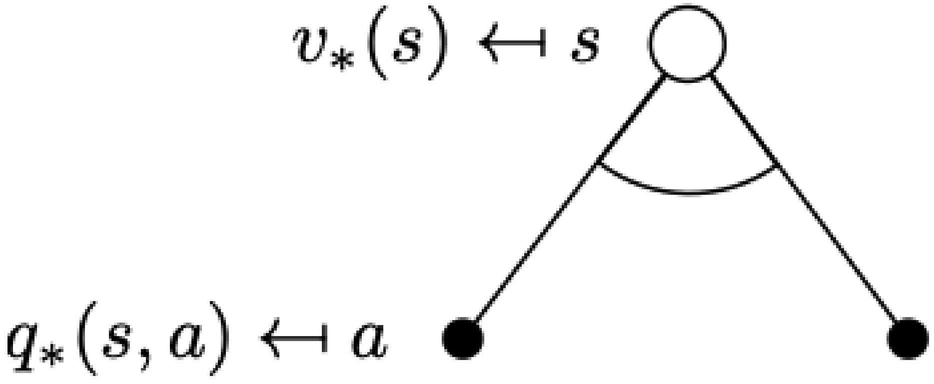


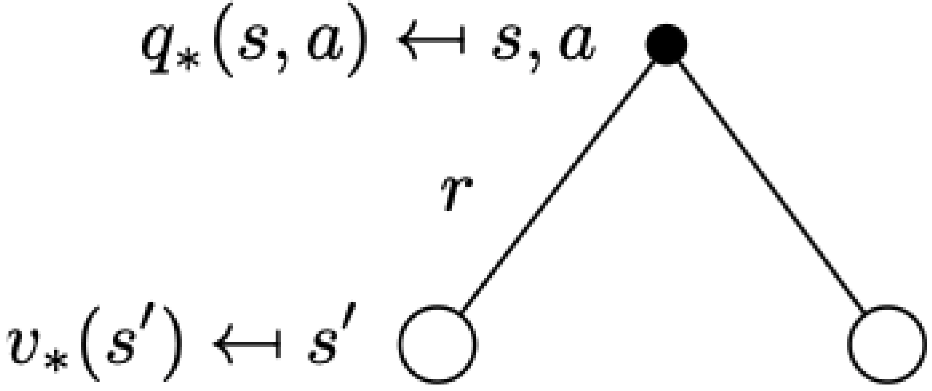
Optimal Policy

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



Bellman Optimality Equation





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

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

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

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

Bellman Optimality Equation

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')]$$

- Bellman Optimality Equation is non-linear
- No closed form solution
- Many iterative solution methods
  - Value/policy iteration
  - Q-learning
  - SARSA
- Bellman Expectation Equation ($v = \mathcal{R} + \gamma \mathcal{P} v$)
  - Usually, $\mathcal{R}$ and $\mathcal{P}$ are unknown (model-free)
  - Too many states → infeasible

$$v^\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v^\pi(s')]$$

Also should be iterative

Reference

- David Silver, COMPM050/COMPGI13 Lecture Notes