Lecture 5: Mathematical Models of RL

Ziyu Shao

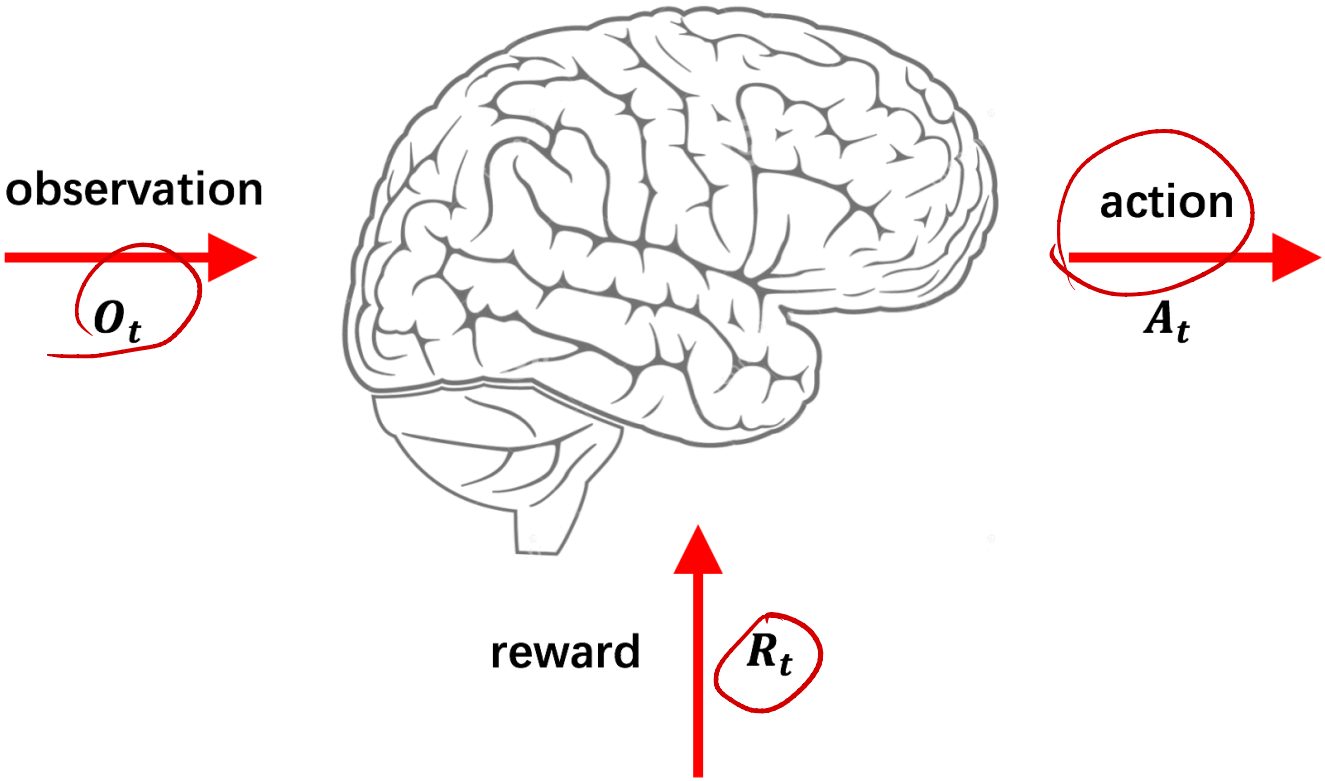
School of Information Science and Technology
ShanghaiTech University

April 09, 2025

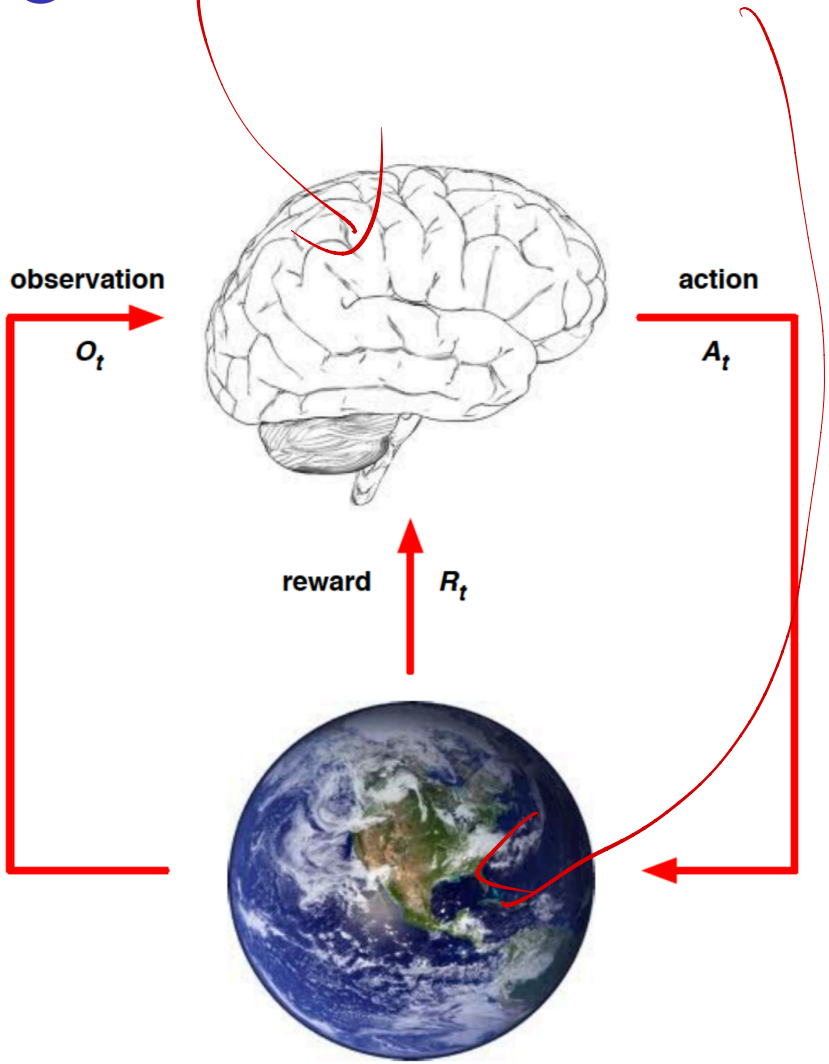
Outline

Outline

Agent and Environment



Agent and Environment

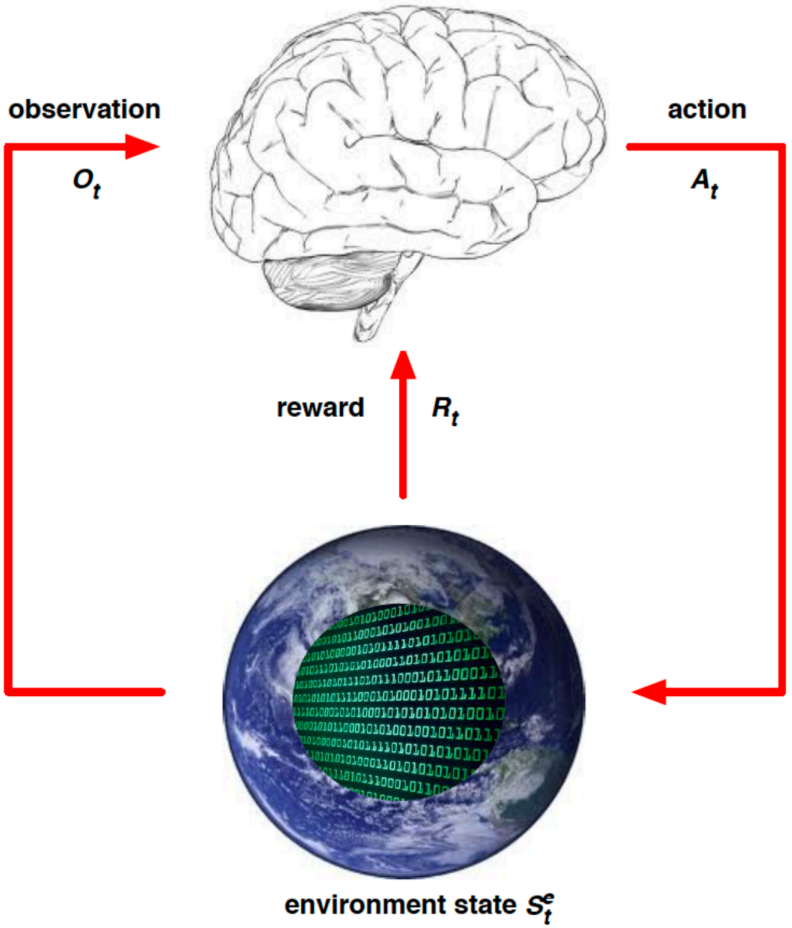


- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

History and State

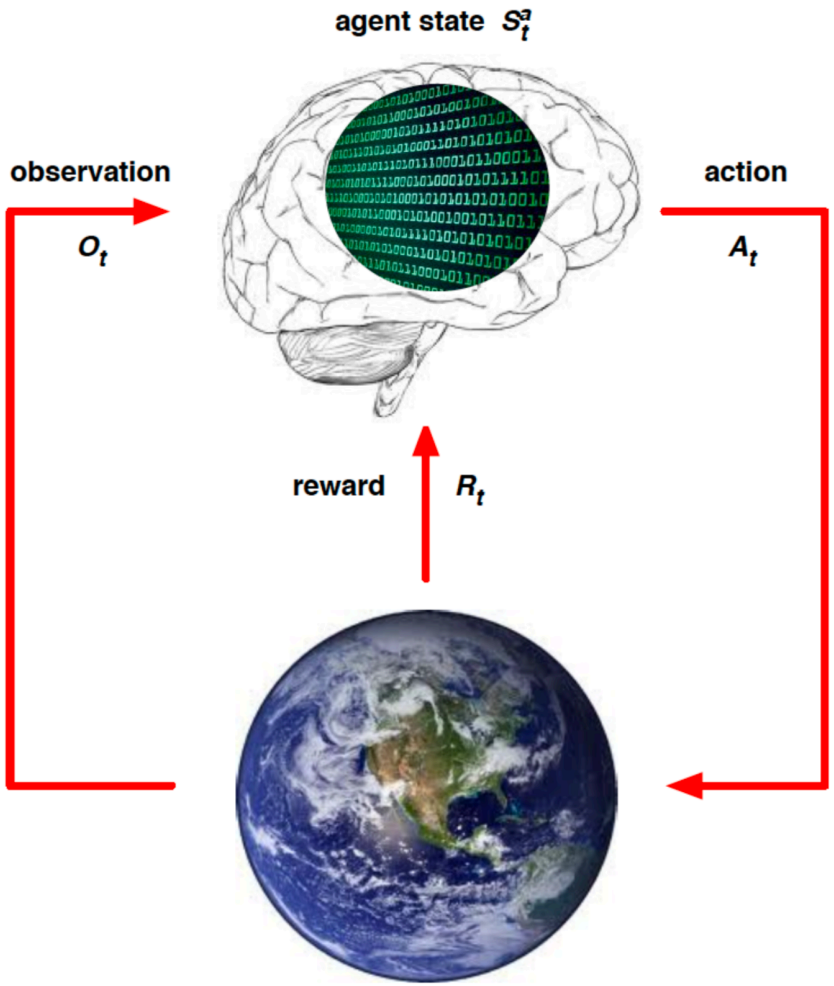
- The history is the sequence of observations, actions, rewards $H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$
- i.e. all observable variables up to time t
- i.e. the sensorimotor stream of a robot or embodied agent
- What happens next depends on the history:
  - The agent selects actions
  - The environment selects observations/rewards
- State is the information used to determine what happens next
- Formally, state is a function of the history: $S_t = f(H_t)$

Environment State



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

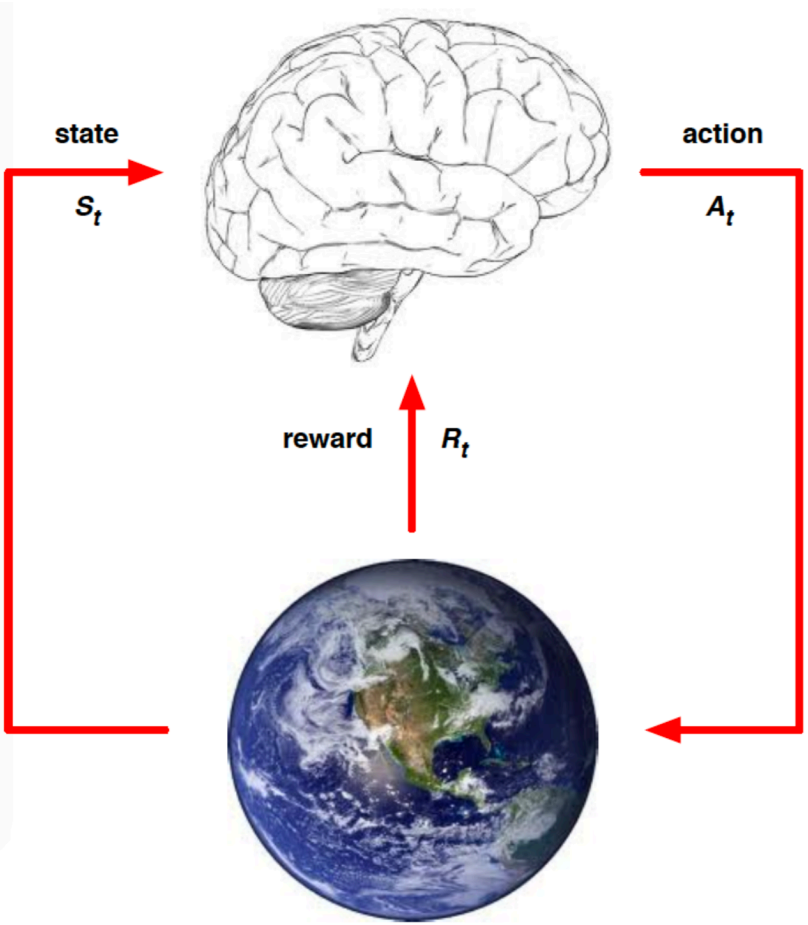
Agent State



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

$$S_t^a = f(H_t)$$

Fully Observable Environments



Full observability: agent directly observes environment state

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

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

Partially Observable Environments

- Partial Observability
  - A robot with camera vision isn't told its absolute location
  - A trading agent only observes current prices
  - A poker playing agent only observes public cards
- Now agent state $\neq$ environment state
- Formally, this is a partially Observable Markov decision process (POMDP)

Major Components of An RL Agent

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

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

Policy

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

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.

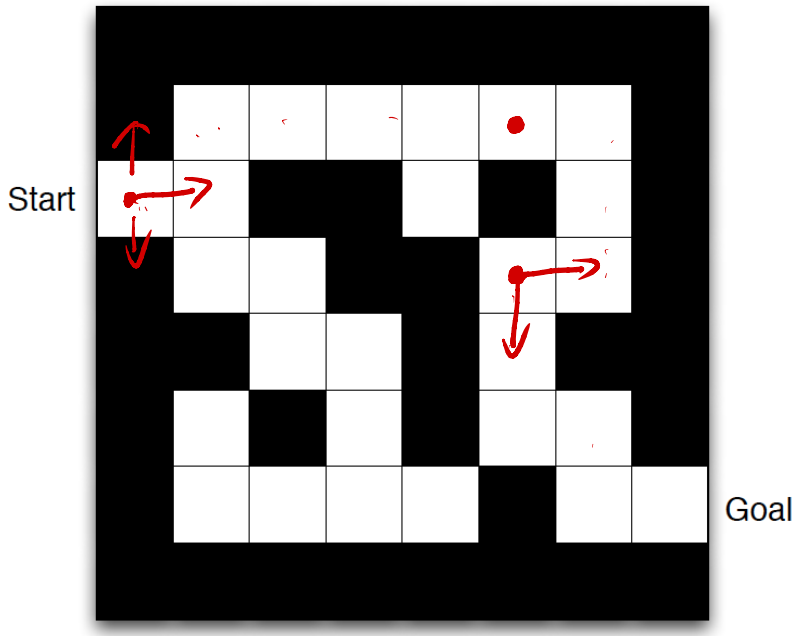
$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s]$$

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.

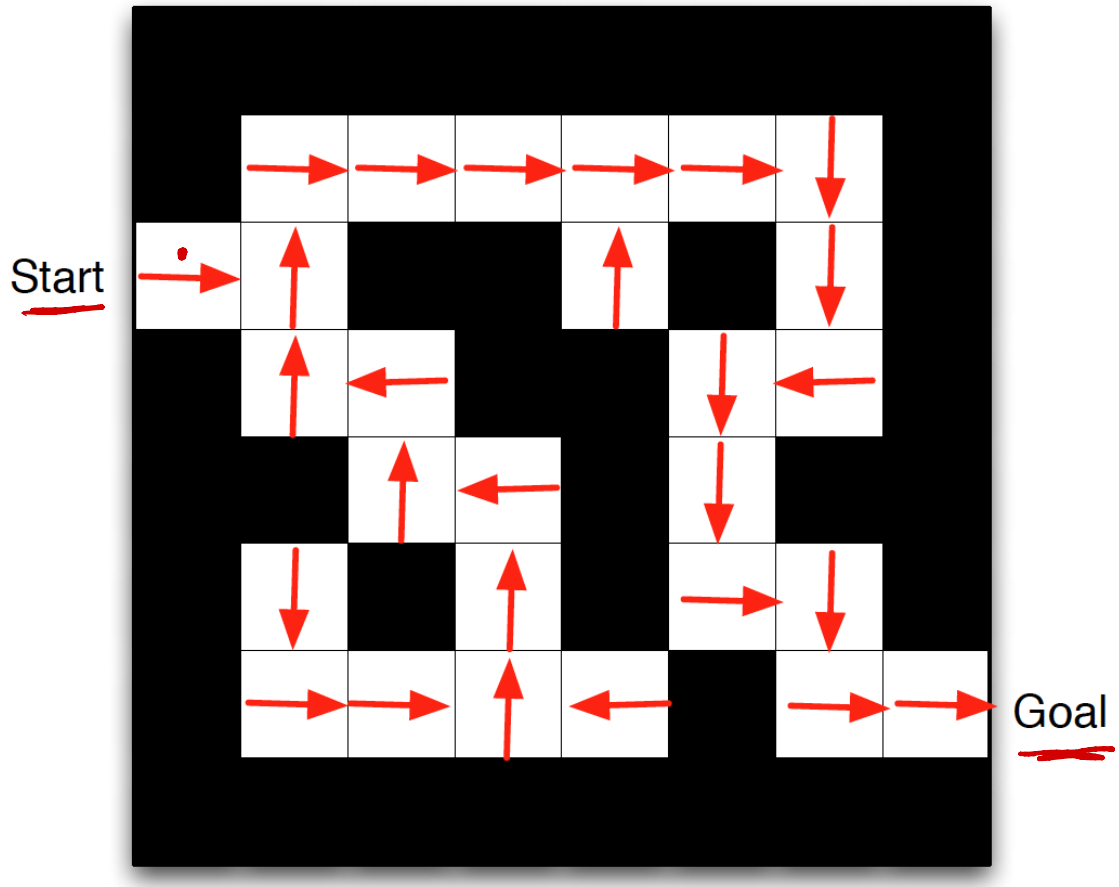
$$\mathcal{P}_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$
$$\mathcal{R}_{s}^{a} = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

Maze Example



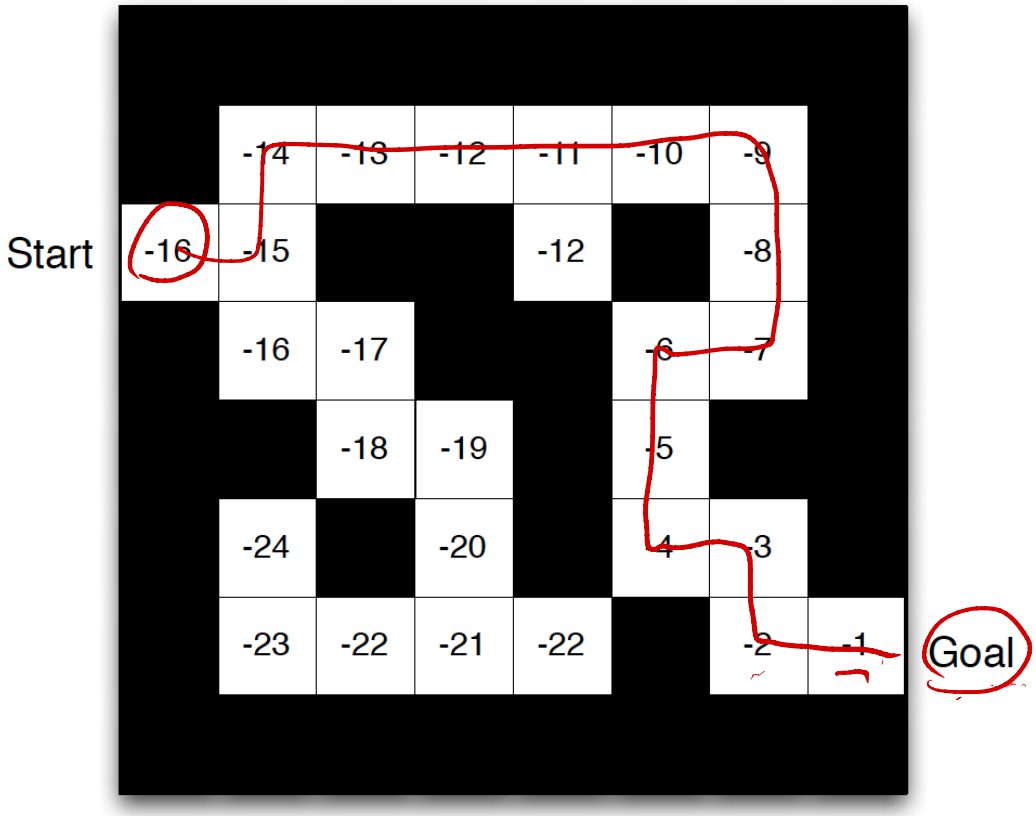
- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

Maze Example: Policy



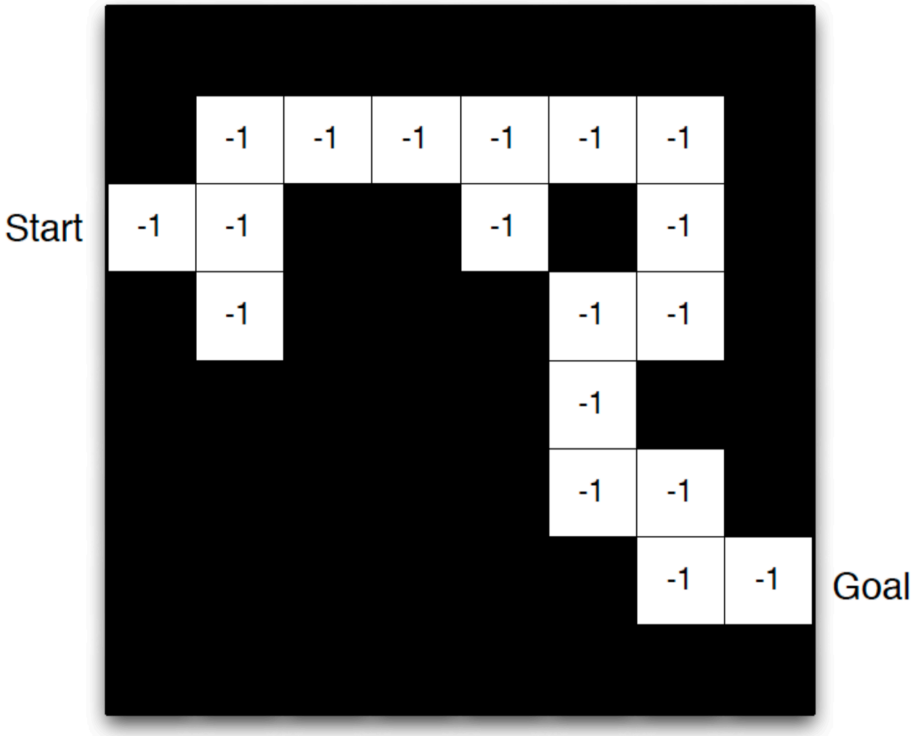
- Arrows represent policy $\pi(s)$ for each state s

Maze Example: Value Function



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

Maze Example: Model



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

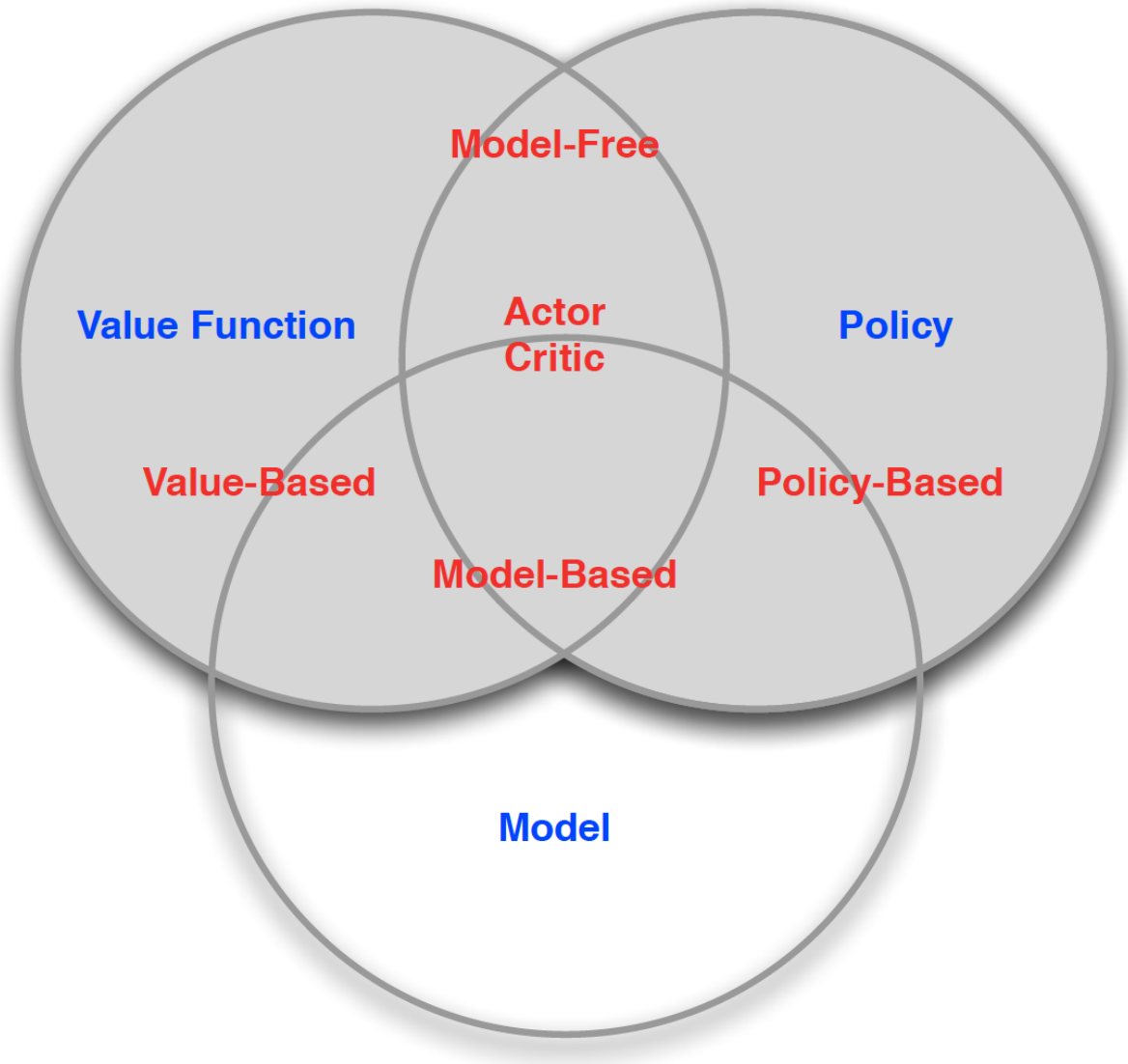
Categorizing RL Agents: I

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

Categorizing RL Agents: II

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

RL Agent Taxonomy

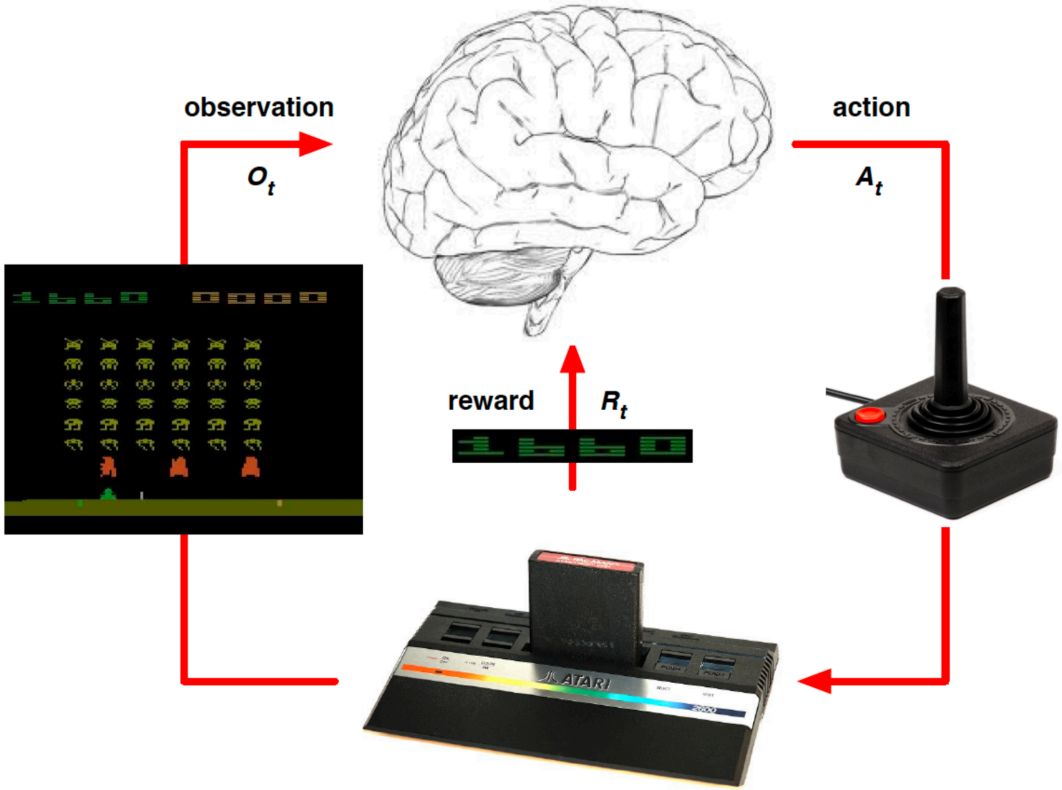


Learning and Planning

Two fundamental problems in sequential decision making

- Reinforcement Learning
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy
- Planning
  - A model of the environment is known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy
  - a.k.a. deliberation, reasoning, introspection, pondering, thought, search

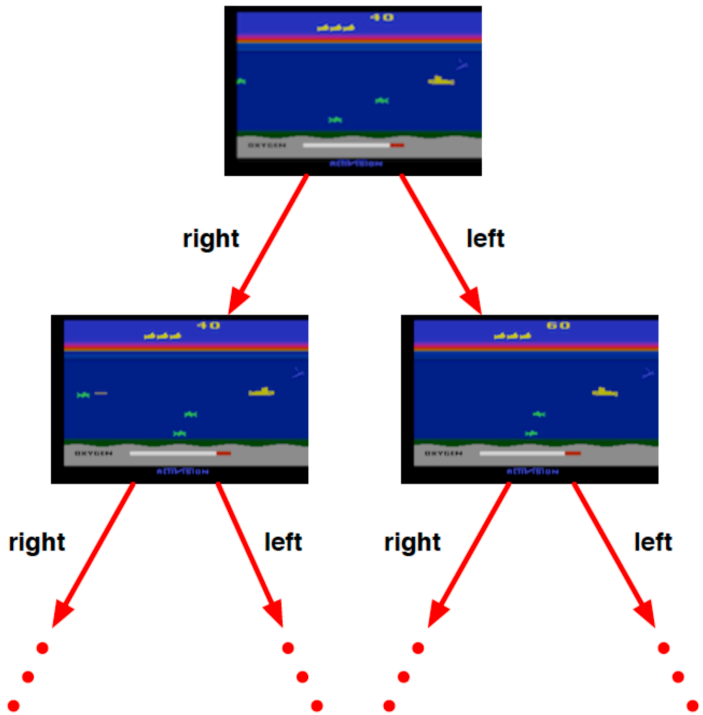
Atari Example: Reinforcement Learning



- Rules of the game are unknown
- Learn directly from interactive game-play
- Pick actions on joystick, see pixels and scores

Atari Example: Planning

- Rules of the game are known
- Can query emulator
  - perfect model inside agent's brain
- If I take action a from state s:
  - what would the next state be?
  - what would the score be?
- Plan ahead to find optimal policy
  - e.g. tree search

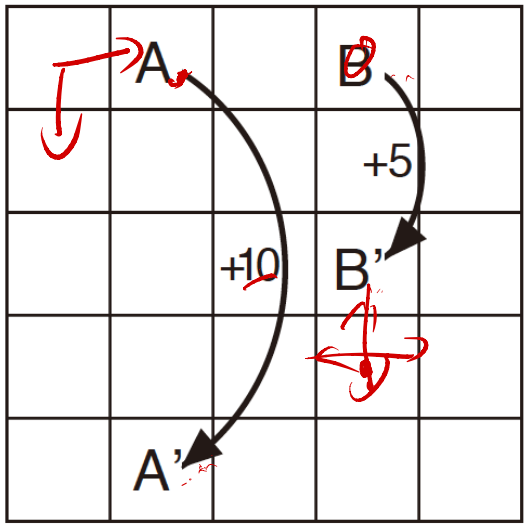


Prediction and Control

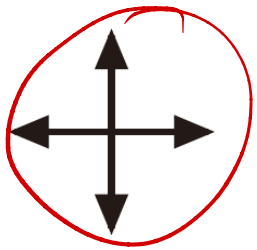
- Prediction: evaluate the future
  - Given a policy
- Control: optimize the future
  - Find the best policy

Gridworld Example: Prediction

estimation



(a)



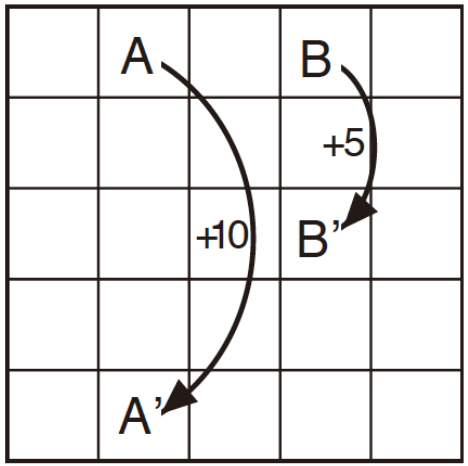
Actions

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

What is the value function for the uniform random policy?

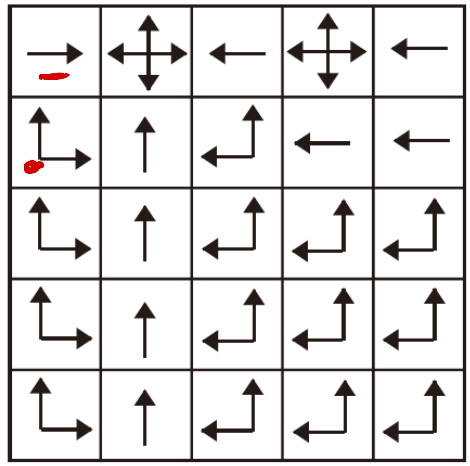
Gridworld Example: Control



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

b) v_*



c) π_*

What is the optimal value function over all possible policies?
What is the optimal policy?

Connections with Psychology

- Prediction & Control algorithms in reinforcement learning parallels Classical & Instrumental conditioning in animal learning.
- Environment Models in reinforcement learning parallels Cognitive Maps in animal learning.
  - they can be learned by supervised learning methods without relying on reward signals
  - they can be used later to plan behavior
- Model-free & Model-based algorithms in reinforcement learning parallels Habitual & Goal-directed behavior in pyschology.

Outline

Markov Property

Definition

A state S_t is Markovian if and only if

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

- "The future is independent of the past given the present"
- 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

Markov Chain

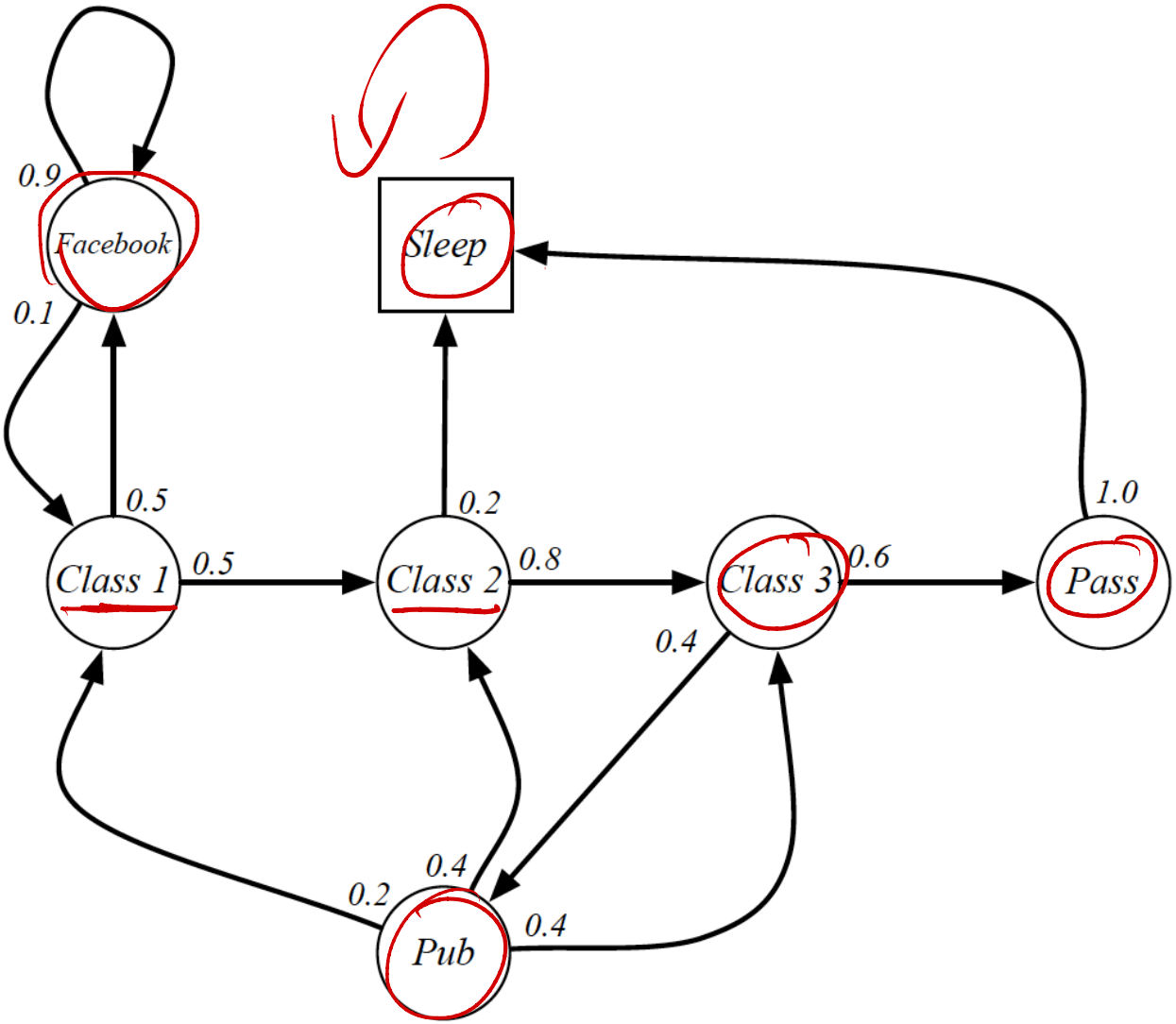
Definition

A discrete-time Markov chain is a tuple $< \mathcal{S}, \mathcal{P} >$

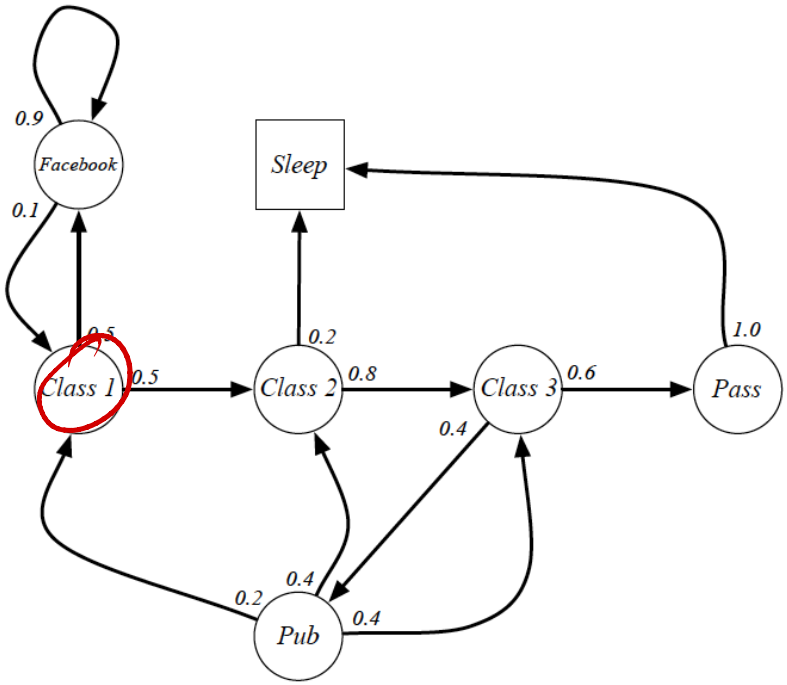
- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix

$$\mathcal{P}_{s,s'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$$

Example: Student Markov Chain



Example: Student Markov Chain Episodes

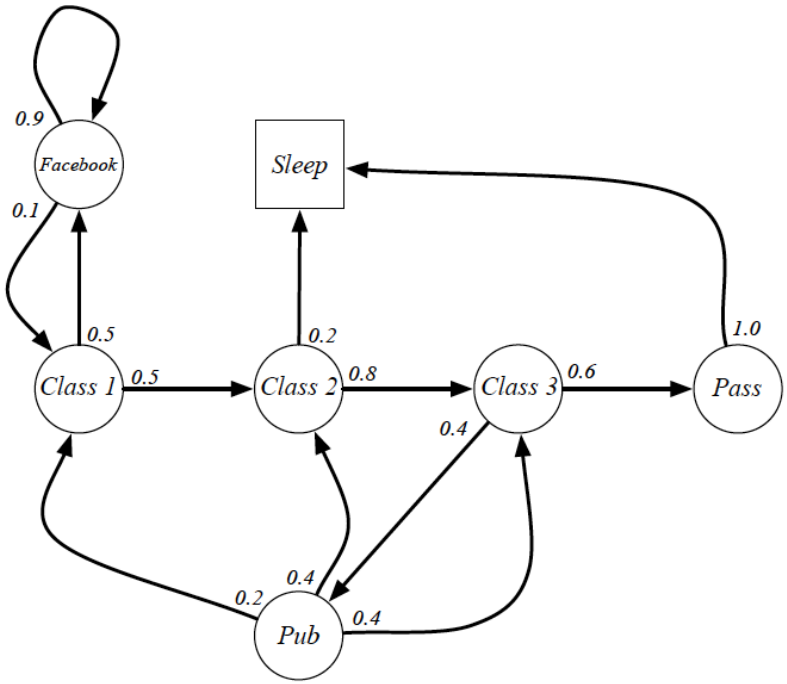


Sample episodes for Student Markov Chain starting from $S_1 = \text{C1}$

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

- 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

Example: Student Markov Chain Transition Matrix



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

Outline

Markov Reward Process

A Markov reward process is a Markov chain with values.

Definition

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

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix

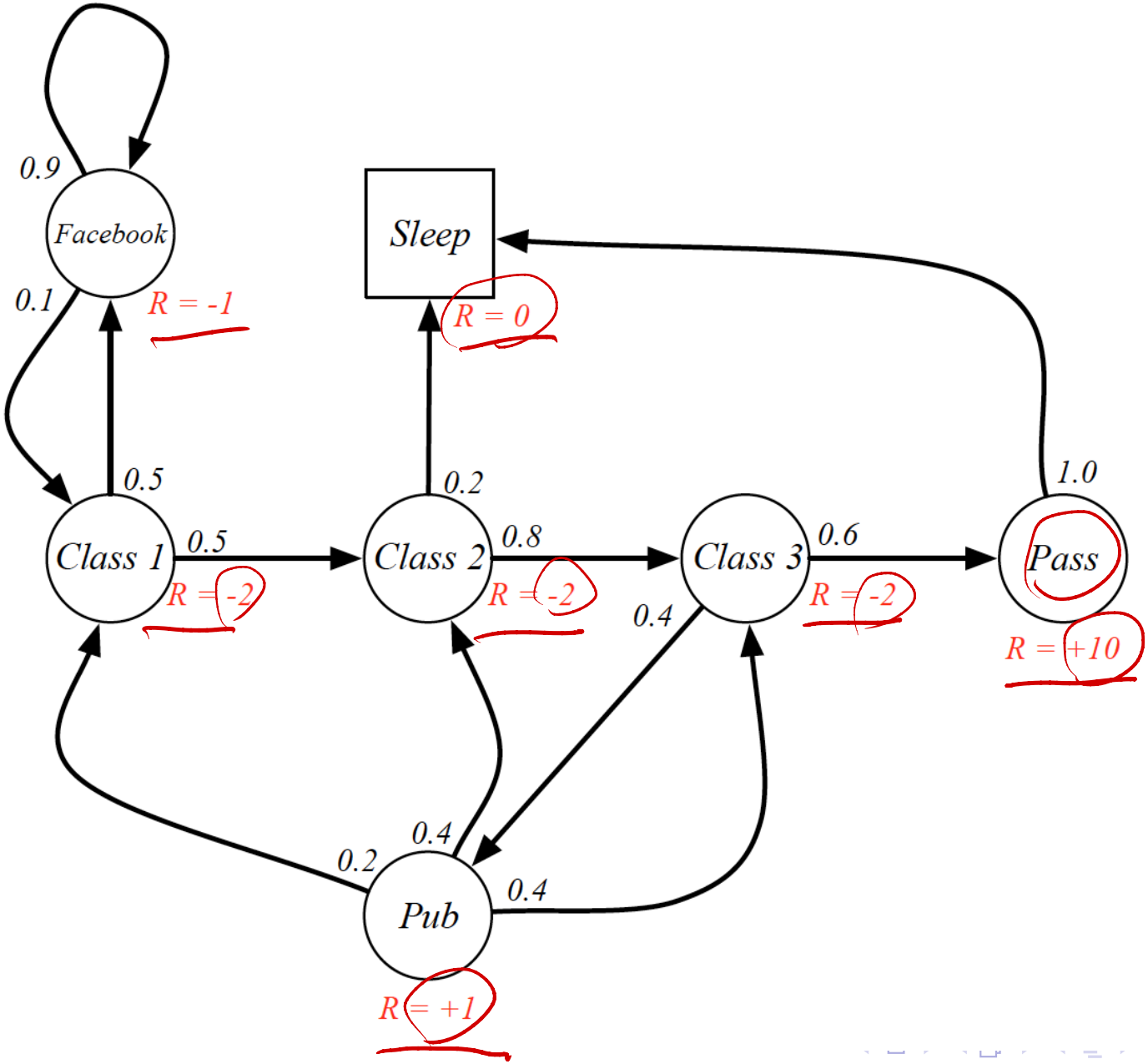
$$\mathcal{P}_{s,s'} = \mathbb{P}[S_{t+1} = s' | S_t = s]$$

- $\mathcal{R}$ is a reward function,

$$\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$$

- γ is a discount factor, $\gamma \in [0, 1]$

Example: Student MRP



Return

Definition

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

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

- The *discount* $\gamma \in [0, 1]$ is the present value of the future rewards
- The value of receiving reward R after $k+1$ time-steps is $\gamma^k R$.
- This values immediate reward above delayed reward.
  - γ close to 0 leads to "myopic" evaluation
  - γ close to 1 leads to "far-sighted" evaluation

Why Discount?

Most Markov reward and decision processes are discounted. Why?

- Mathematically convenient to discount rewards
- Avoids infinite returns in cyclic Markov processes
- Uncertainty about the future may not be fully represented
- If the reward is financial, immediate rewards may earn more interest than delayed rewards
- Animal & human behavior shows preference for immediate reward
- It is sometimes possible to use undiscounted Markov reward processes (i.e. $\gamma = 1$), e.g. if all sequences terminate.

Value Function

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

Definition

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

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

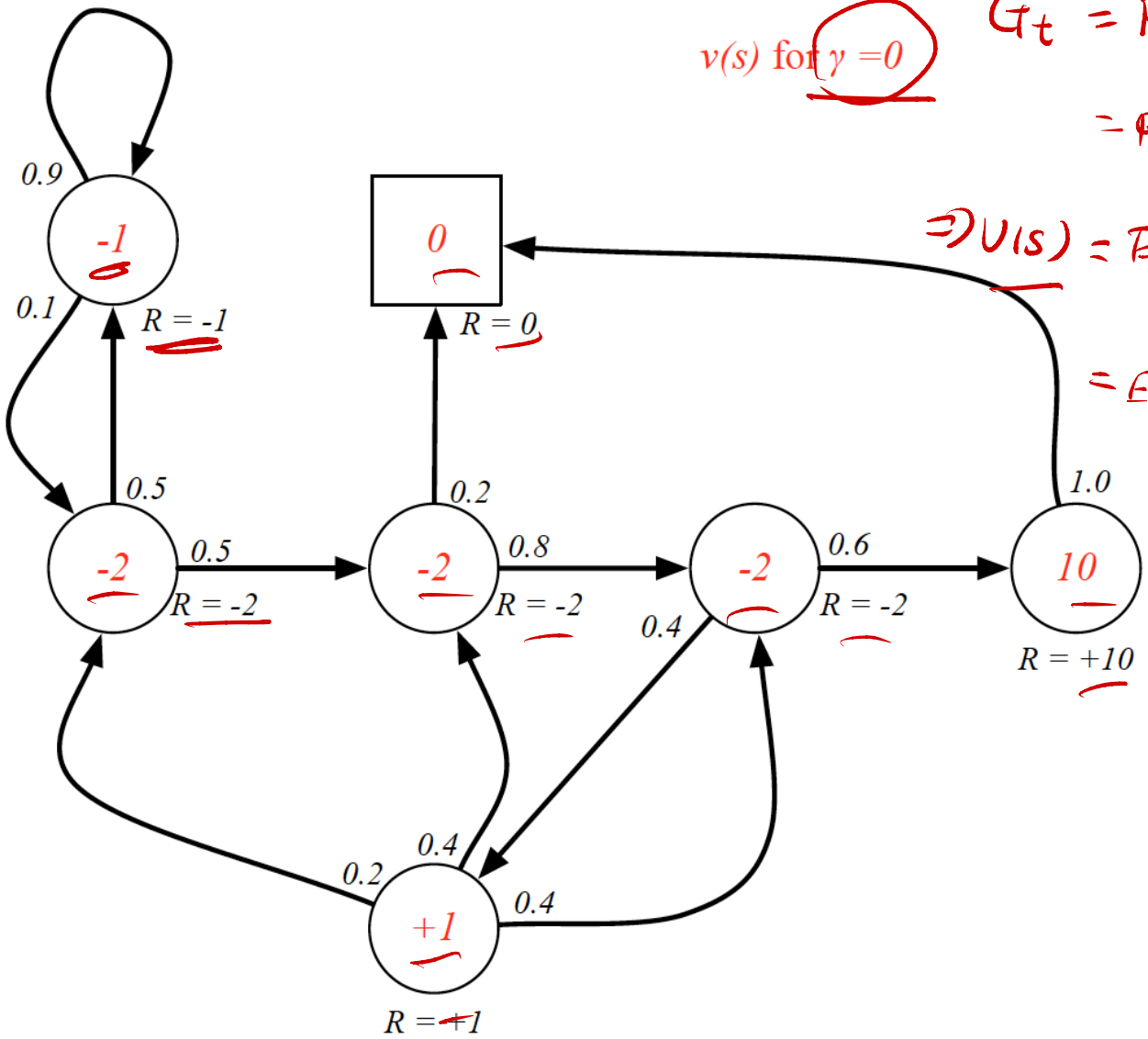
Example: Student MRP Returns

Sample returns for Student MRP:
Starting from $S_1 = \text{C1}$ 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	

Example: State-Value Function for Student MRP



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

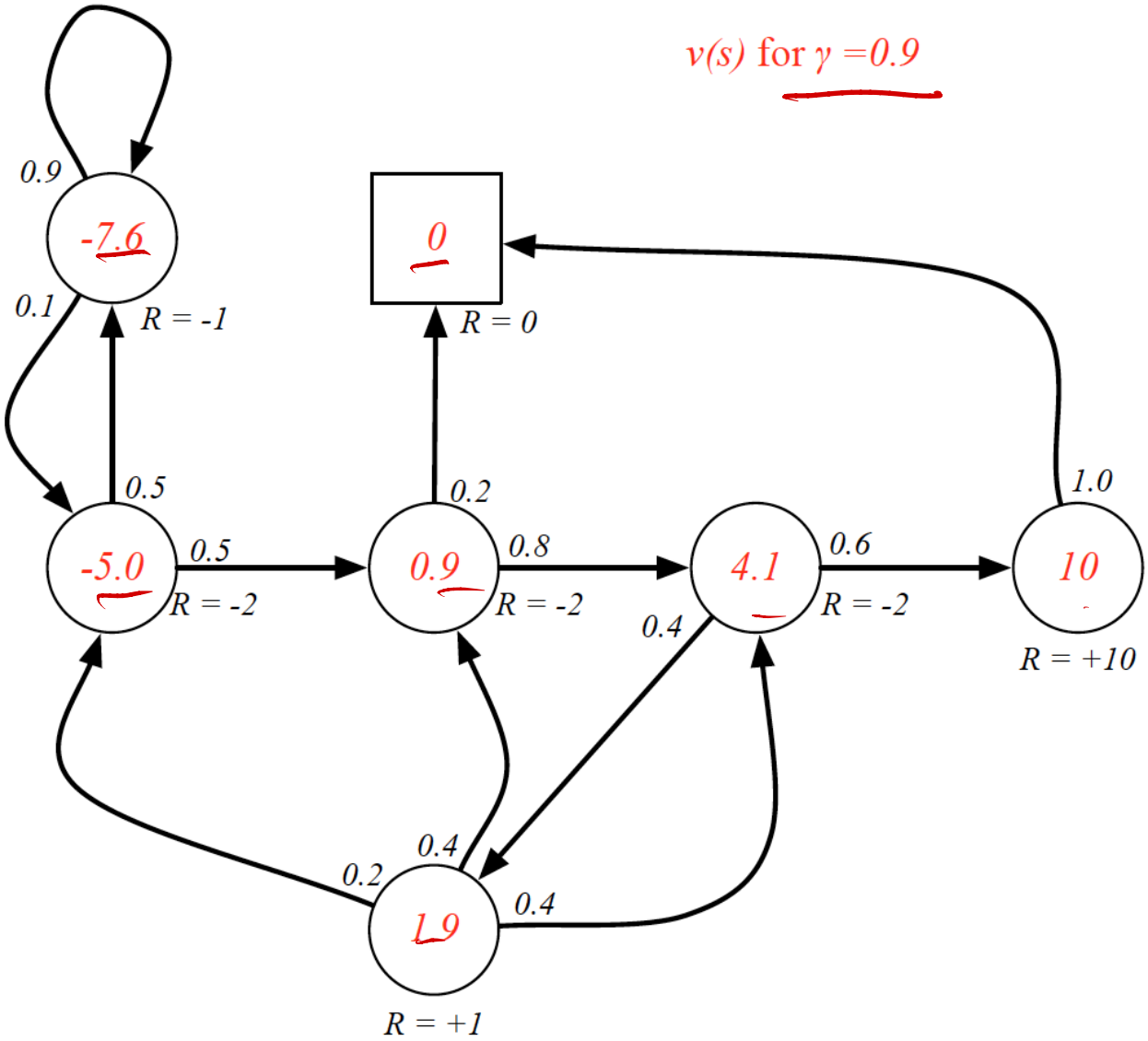
$= R_{t+1}$

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

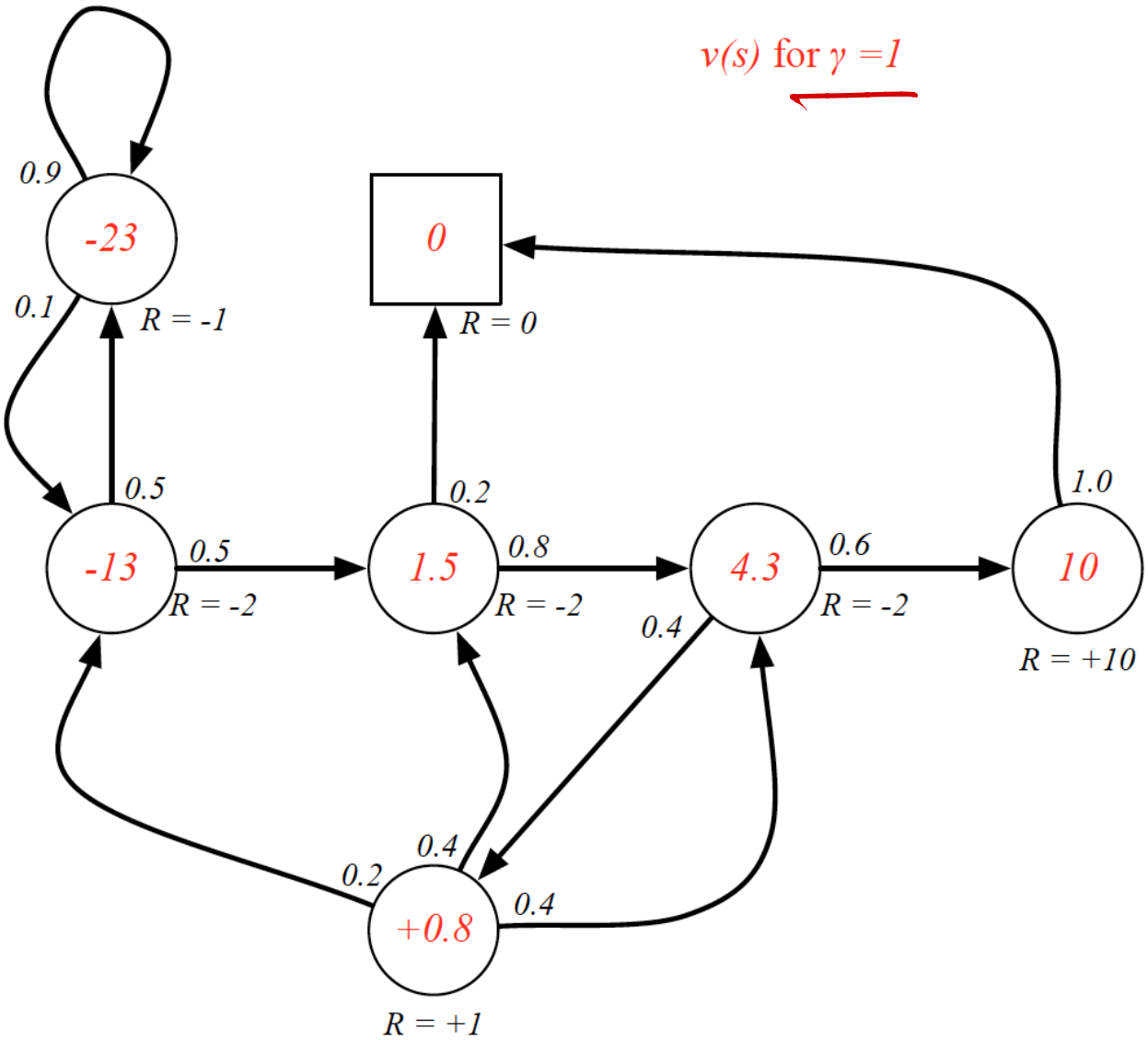
$= \mathbb{E}[R_{t+1} | S_t = s]$

$= R_s$

Example: State-Value Function for Student MRP



Example: State-Value Function for Student MRP



Bellman Equation for MRPs

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

The value function can be decomposed into two parts:

- immediate reward R_{t+1}
- discounted value of successor state $\gamma v(S_{t+1})$

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

Bellman Equation for MRPs

1°. $V(s) = E[G_t | S_t = s]$; $V(S_t) = E[G_t | S_t]$; $V(S_{t+1}) = E[G_{t+1} | S_{t+1}]$

2° Adam's Law. $E[E[Y|X]] = E[Y]$; $\hat{E}(\cdot) = E(\cdot|Z)$

Adam's Law with Extra Conditioning. $\hat{E}(\hat{E}(Y|X)) = \hat{E}(Y)$

$\Leftrightarrow$ $E[E[Y|X,Z]|Z] = E[Y|Z]$

3°. Let $Y = G_{t+1}$, $X = S_{t+1}$, $Z = S_t$,

$\Rightarrow E[E[G_{t+1}|S_{t+1}, S_t]|S_t] = E[G_{t+1}|S_t]$

↓ Markov property.

$E[\underbrace{E[G_{t+1}|S_{t+1}]}_{V(S_{t+1})}|S_t] \longrightarrow E[V(S_{t+1})|S_t]$

$\Rightarrow E[G_{t+1}|S_t] = E[V(S_{t+1})|S_t] \Rightarrow \forall s,\ E[G_{t+1}|S_t = s] = E[V(S_{t+1})|S_t = s]$

Bellman Equation for MRPs

4°. $v(s) = \mathbb{E}[G_t | S_t = s] = \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s]$

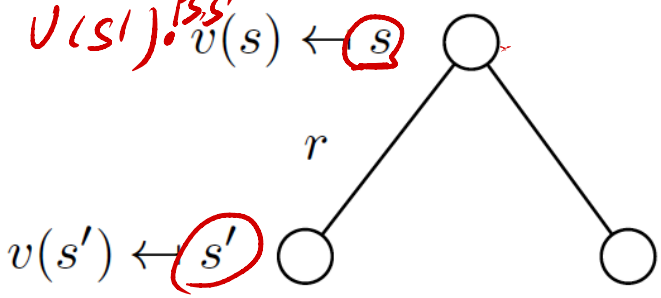
$= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \mathbb{E}[G_{t+1} | S_t = s]$

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

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

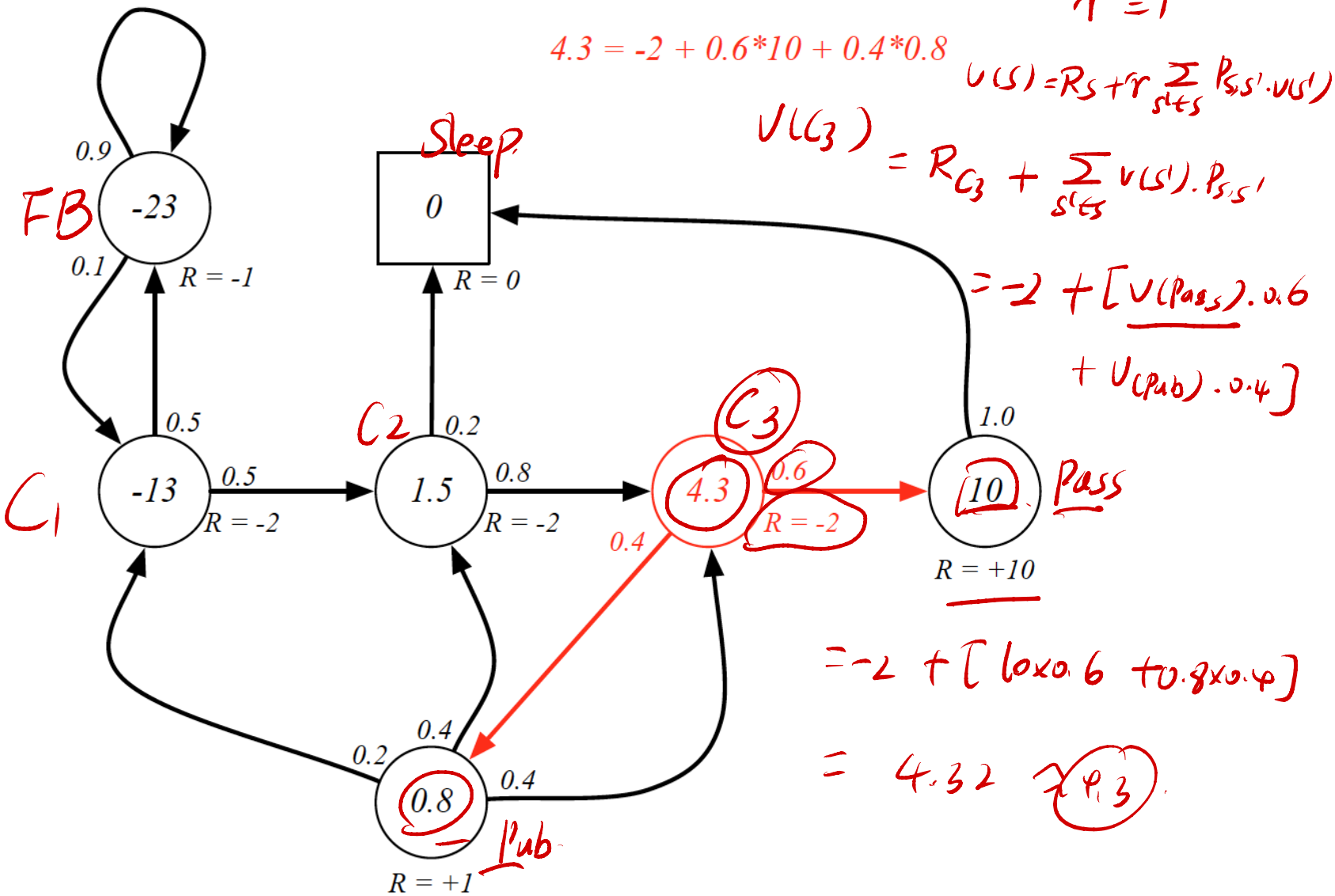
Bellman Equation for MRPs

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



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

Example: Bellman Equation for Student MRP



Bellman Equation in Matrix Form

The Bellman equation can be expressed concisely using matrices,

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

where $\boldsymbol{v}$ is a column vector with one entry per state

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

Solving the Bellman Equation

- The Bellman equation is a linear equation
- It can be solved directly:

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

- Computational complexity is $O(n^3)$ for n states
- Direct solution only possible for small MRPs
- There are many iterative methods for large MRPs, e.g.
  - Dynamic programming
  - Monte-Carlo evaluation
  - Temporal-Different learning

Outline

M.C.

M.C + State-Value. (reward)

M.C + reward, action | action-value (2), state-value

Markov Decision Process

A Markov decision process (MDP) is a Markov reward process with decisions. It is an environment in which all states are Markovian.

Definition

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

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix

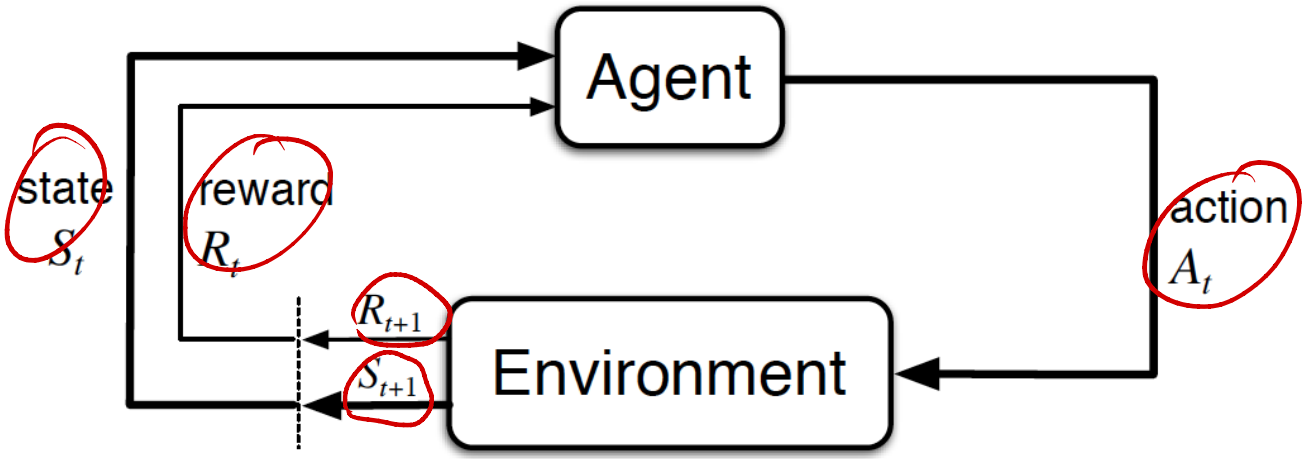
$$\mathcal{P}^{a}_{s,s'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

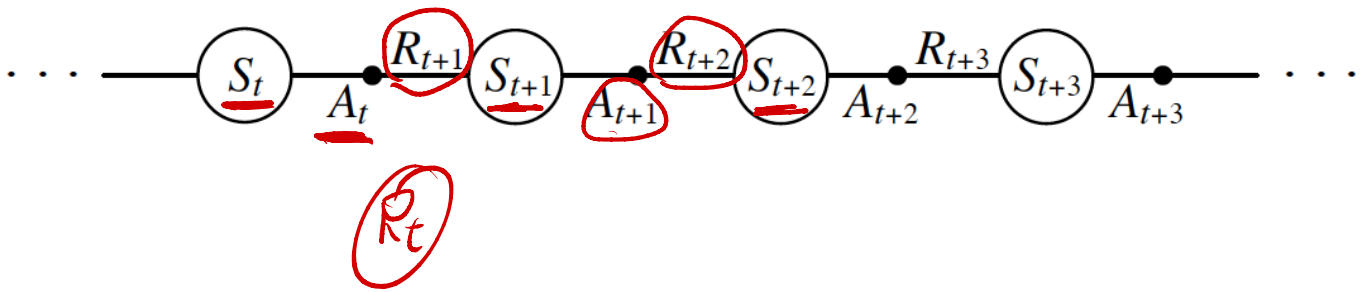
- $\mathcal{R}$ is a reward function,

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

- γ is a discount factor, $\gamma \in [0, 1]$

The Agent-Environment Interface

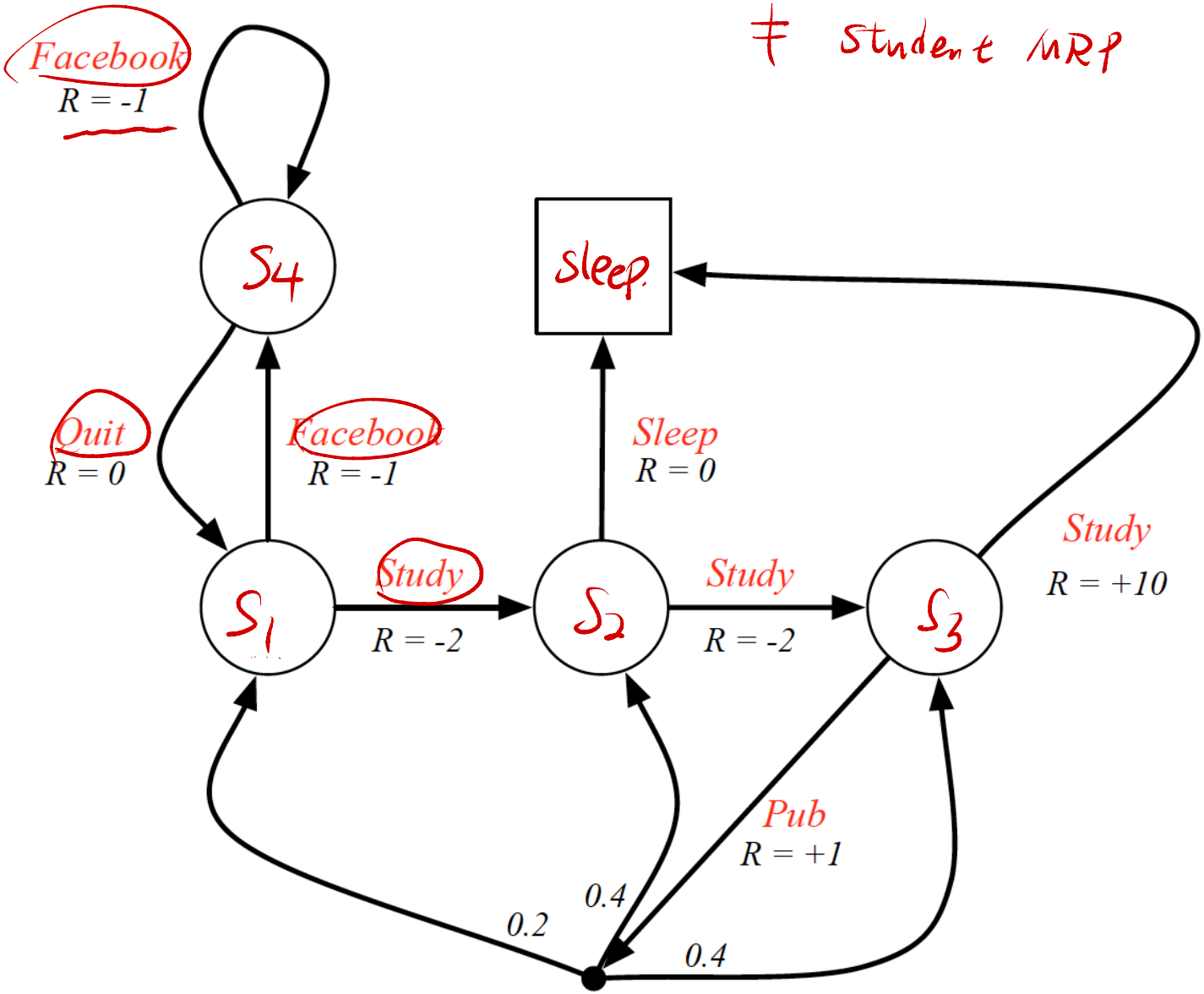




Example: Student MDP

≠ Student M.C

≠ Student MRP



Policy

Definition

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

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

- A policy fully defines the behavior of an agent
- MDP policies depend on the current state (not the history)
- i.e. Policies are *stationary* (time-independent), $A_t \sim \pi(\cdot|S_t), \forall t > 0$.

Policy

1^o $P^{\pi}_{s,s'} = P^{\pi}[S_{t+1}=s' \mid S_t=s] \overset{\text{LOTP}}{=}$

$= \sum_{a\in A} P[S_{t+1}=s' \mid A_t=a, S_t=s] \cdot P(A_t=a \mid S_t=s)$

$= \sum_{a\in A} P^a_{s,s'} \cdot \pi(a|s)$

- Given an MDP $\mathcal{M} = < \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ and a policy π
- The state sequence $S_1, S_2, \ldots$ is a Markov process $< \mathcal{S}, \mathcal{P}^\pi >$
- The state and reward sequence $S_1, R_2, S_2 \ldots$ is a Markov reward process $< \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma >$
- where

2^o $R^{\pi}_s = E_\pi[R_{t+1} \mid S_t=s] \overset{\text{LOTE}}{=} \sum_{a\in A} E[R_{t+1} \mid S_t=s, A_t=a] \cdot P(A_t=a \mid S_t=s)$

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

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

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

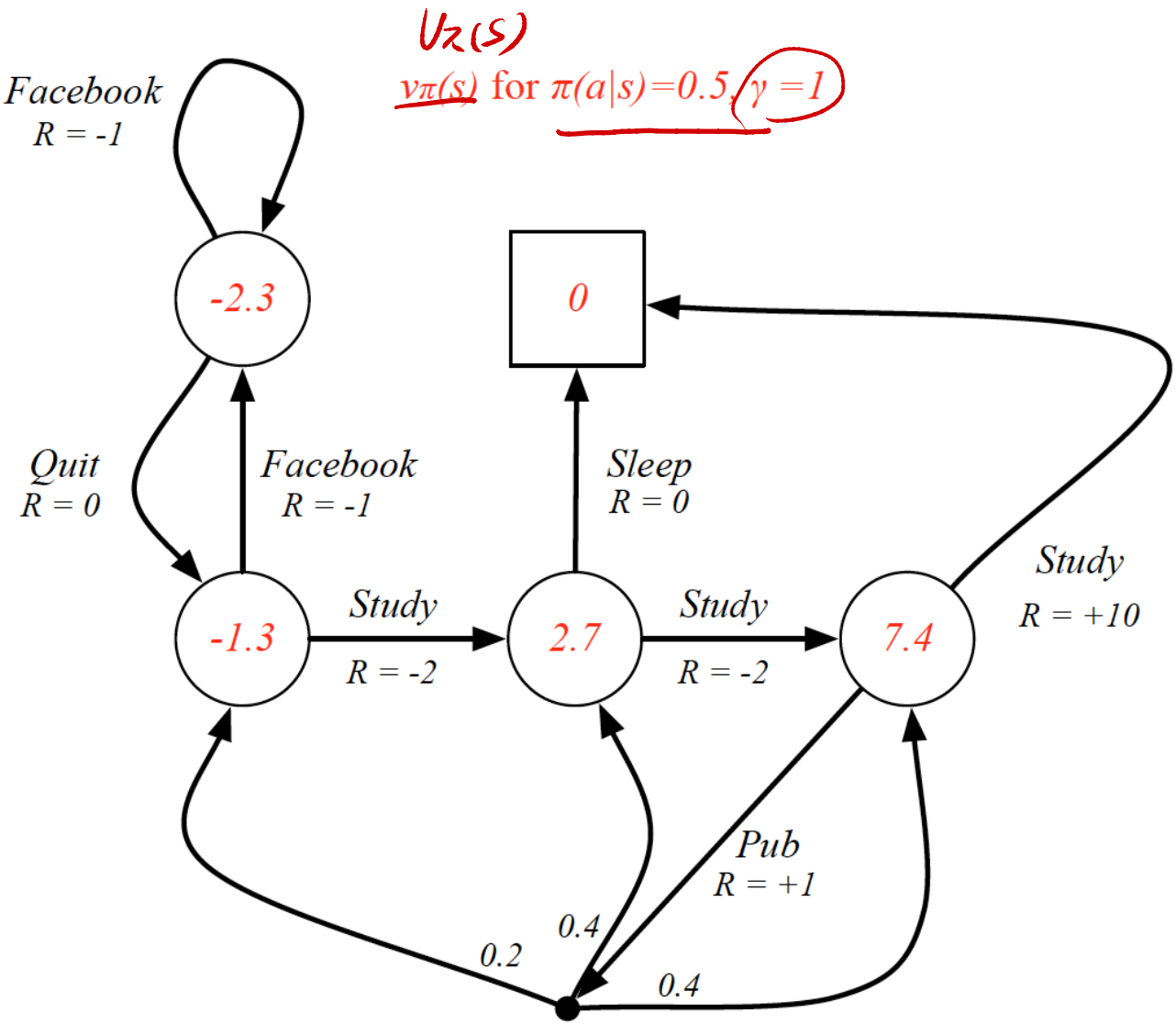
$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

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

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

Example: State-Value Function for Student MDP



Bellman Expectation Equation

The state-value function can again be decomposed into immediate reward plus discounted value of successor state,

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

The action-value function can similarly be decomposed,

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

Bellman Expectation Equation

1^0. $q_\pi(s,a) = E_\pi[G_t \mid S_t=s, A_t=a]$

$q_\pi(S_t,A_t) = E_\pi[G_t \mid S_t, A_t]$, $q_\pi(S_{t+1},A_{t+1}) = E_\pi[G_{t+1} \mid S_{t+1}, A_{t+1}]$

2^0. Adam's Law with extra conditioning.

$$E[Y|Z] = E[E[Y|X,Z]|Z]$$

Let $Y = G_{t+1}$, $Z = (S_t, A_t)$, $X = (S_{t+1}, A_{t+1})$.

$$\Rightarrow E[G_{t+1} \mid S_t, A_t] \overset{\text{Adam}}{=} E[E[G_{t+1} \mid S_{t+1}, A_{t+1}, S_t, A_t] \mid S_t, A_t]$$

$$\overset{\text{Markov}}{=} E[E[G_{t+1} \mid S_{t+1}, A_{t+1}] \mid S_t, A_t]$$

$$= E[q_\pi(S_{t+1}, A_{t+1}) \mid S_t, A_t].$$

$$\Rightarrow E[G_{t+1} \mid S_t=s, A_t=a] = E_\pi[q_\pi(S_{t+1}, A_{t+1}) \mid S_t=s, A_t=a]$$

$\forall s, a$.

Bellman Expectation Equation

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

3°. $q_\pi(s,a) \triangleq E_\pi[G_t \mid S_t=s, A_t=a]$

$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t=s, A_t=a]$

$= E_\pi[R_{t+1} \mid S_t=s, A_t=a] + \gamma E_\pi[G_{t+1} \mid S_t=s, A_t=a]$

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

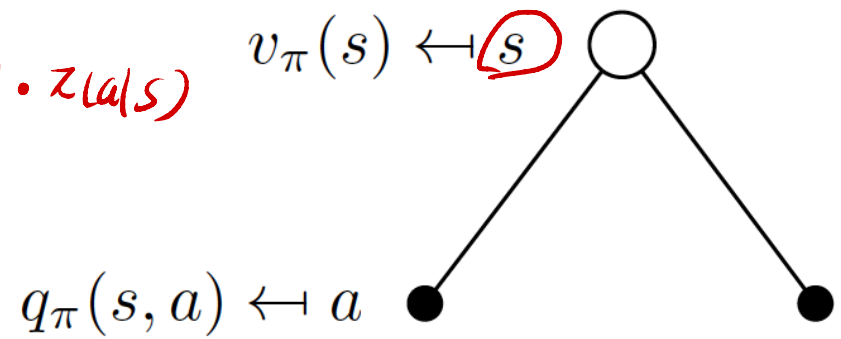
$= E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t=s, A_t=a]$

Bellman Expectation Equation for V^π

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

LoTE $= \sum_{a \in A} E_\pi[G_t | S_t = s, A_t = a] \cdot P(A_t = a | S_t = s)$

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



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

$v^2 \to v^2$

$v^2 \to q^2$

$q^2 \to v^2$

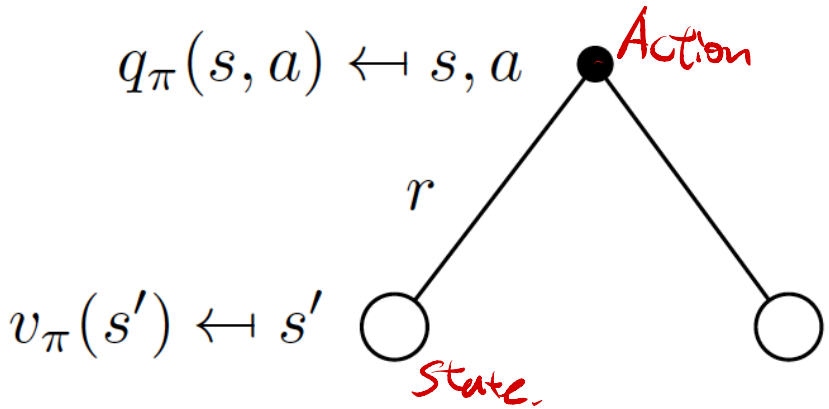
$q^2 \to q^2$

$E_\pi[q_\pi(s, A)]$

A: action

$\sim \pi(a|s)$

Bellman Expectation Equation for Q^π



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

Bellman Expectation Equation for Q^π

$$1^\circ.\quad \mathbb{E}_\pi\left[q_\pi(S_{t+1}, A_{t+1}) \mid S_{t+1}=s', S_t=s, A_t=a\right]$$

$$\overset{\text{LOTE}}{=} \sum_{a'\in\mathcal{A}} \mathbb{E}\left[q_\pi(S_{t+1}, A_{t+1}) \mid S_{t+1}=s', A_{t+1}=a', S_t=s, A_t=a\right] \cdot P(A_{t+1}=a' \mid S_{t+1}=s', S_t=s, A_t=a)$$

$$= \sum_{a'\in\mathcal{A}} \mathbb{E}\left[q_\pi(S_{t+1}, A_{t+1}) \mid S_{t+1}=s', A_{t+1}=a'\right] \cdot P(A_{t+1}=a' \mid S_{t+1}=s')$$

$$= \underbrace{\sum_{a'\in\mathcal{A}} q_\pi(s', a') \cdot \pi(a'|s')}_{=V_\pi(s')}$$

$$2^\circ.\quad \mathbb{E}_\pi\left[q_\pi(S_{t+1}, A_{t+1}) \mid S_t=s, A_t=a\right]$$

$$\overset{\text{LOTE}}{=} \sum_{s'\in\mathcal{S}} \mathbb{E}\left[q_\pi(S_{t+1}, A_{t+1}) \mid S_t=s, A_t=a, S_{t+1}=s'\right] \cdot P(S_{t+1}=s' \mid S_t=s, A_t=a)$$

$$\overset{1^\circ}{=} \underbrace{\sum_{s'\in\mathcal{S}} V_\pi(s') \cdot P^a_{s,s'}}$$

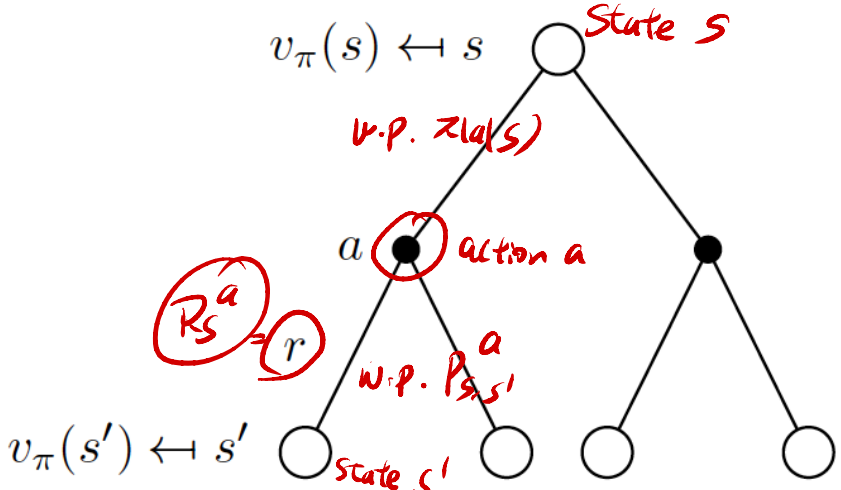
Bellman Expectation Equation for Q^π

$$3^\circ.\quad q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

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

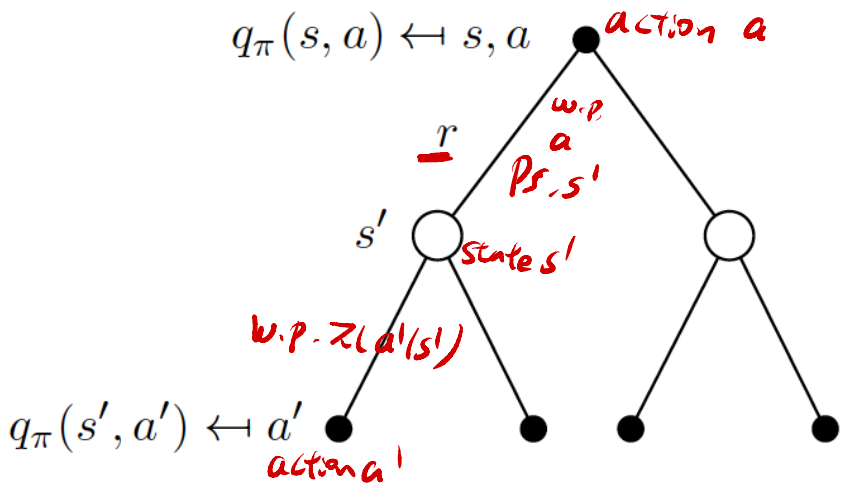
$$= R_s^a + \gamma \sum_{s' \in S} P_{s,s'}^a \cdot V_\pi(s')$$

Bellman Expectation Equation for v_π



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

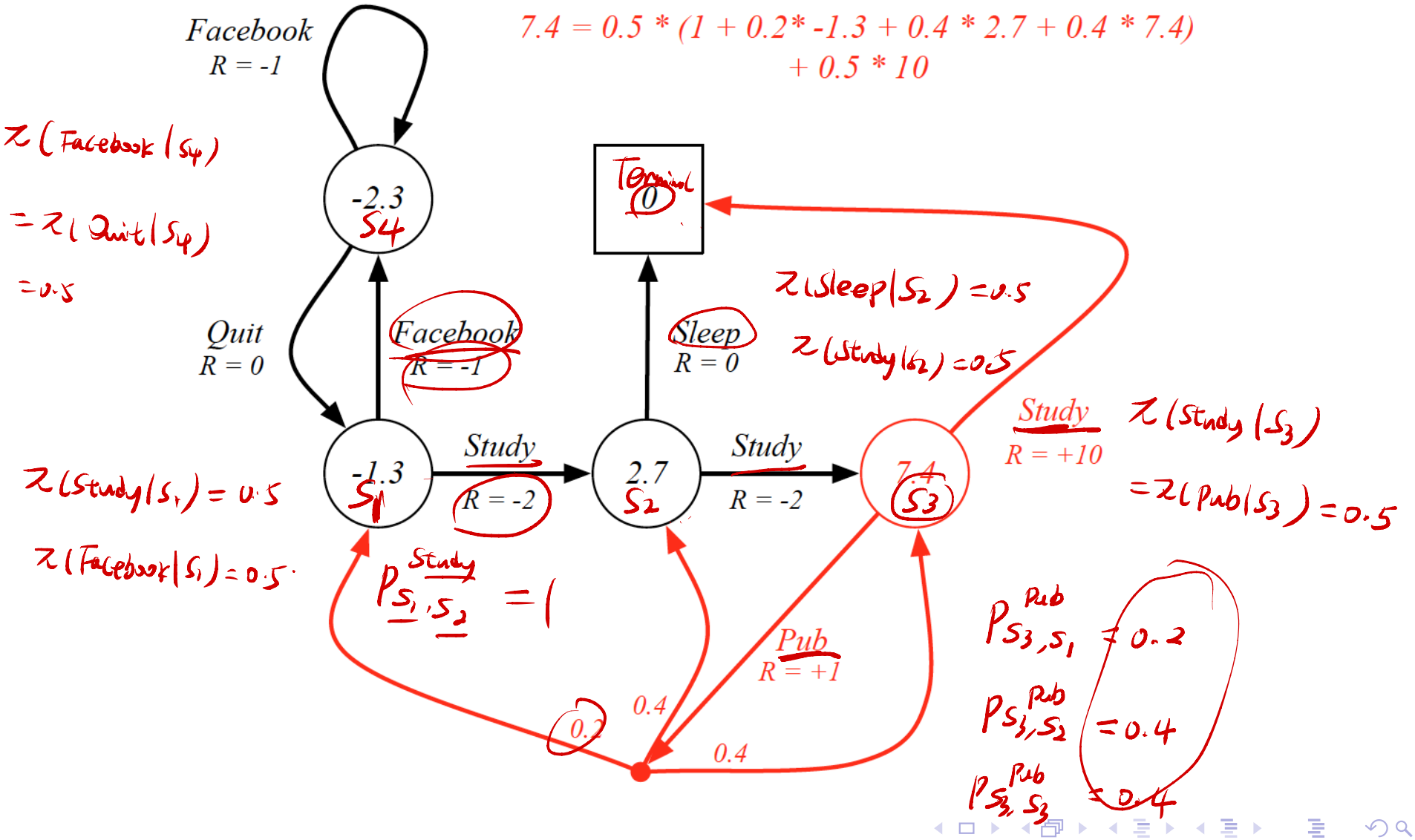
Bellman Expectation Equation for q_π



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

Example: Bellman Expectation Equation in Student MDP

Policy π = equal prob for all options / discount $\gamma = 1$



$7.4 = 0.5 * (1 + 0.2 * -1.3 + 0.4 * 2.7 + 0.4 * 7.4) + 0.5 * 10$

$\pi(\text{Facebook} \mid S_4) = \pi(\text{Quit} \mid S_4) = 0.5$

$\pi(\text{Sleep} \mid S_2) = 0.5$

$\pi(\text{Study} \mid S_2) = 0.5$

$\pi(\text{Study} \mid S_1) = 0.5$

$\pi(\text{Facebook} \mid S_1) = 0.5$

$\pi(\text{Study} \mid S_3) = \pi(\text{Pub} \mid S_3) = 0.5$

$P^{\text{Study}}_{S_1, S_2} = 1$

$P^{\text{Pub}}_{S_3, S_1} = 0.2$

$P^{\text{Pub}}_{S_3, S_2} = 0.4$

$P^{\text{Pub}}_{S_3, S_3} = 0.4$

Example: Bellman Expectation Equation in Student MDP

$V_j \triangleq V_\pi(S_j)$

$$1^0.\quad V_\pi(s) = \sum_{a\in A} \pi(a|s)\left[R_s^a + \gamma \sum_{s'\in S} P_{ss'}^a V_\pi(s')\right];$$

$$2^0.\quad V_1 \triangleq V_\pi(S_1) = \underbrace{\pi(\text{Study}|S_1)}\, 0.5\left[R_{S_1}^{\text{Study}} + 1\cdot 1\cdot V_\pi(S_2)\right]$$
$$+ \underbrace{\pi(\text{Facebook}|S_1)}\, 0.5\left[R_{S_1}^{\text{Facebook}} + 1\cdot 1\cdot V_\pi(S_4)\right]$$
$$= 0.5(-2+V_2) + 0.5(-1+V_4)$$

$$V_2 \triangleq V_\pi(S_2) = 0.5(-2+V_3) + 0.5(0+0)$$

$$V_3 \triangleq V_\pi(S_3) = 0.5(1+0.2V_1+0.4V_2+0.4V_3) + 0.5(0+0)$$

$$V_4 \triangleq V_\pi(S_4) = 0.5(0+V_1) + 0.5(-1+V_4)$$

Example: Bellman Expectation Equation in Student MDP

3°. $V_1 = -1.3$; $V_2 = 2.7$, $V_3 = 7.4$; $V_4 = -2.3$

4°. $q_\pi(s, a) = R_s^a + \gamma \sum_{s' \in S} P_{s,s'}^a V_\pi(s')$

$\Rightarrow$ $q_\pi(S_1, \text{study}) = -2 + 1 \cdot V_2 = -2 + 2.7 = 0.7$

$q_\pi(S_1, \text{Facebook}) = -1 + 1 \cdot V_4 = -1 - 2.3 = -3.3$

$q_\pi(S_2, \text{Sleep}) = 0 + 0 = 0$

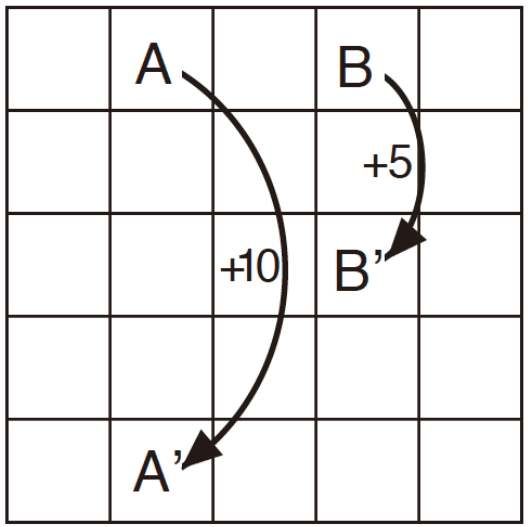
$q_\pi(S_2, \text{Study}) = -2 + 1 \cdot V_3 = -2 + 7.4 = 5.4$

$q_\pi(S_3, \text{Study}) = 10$, $q_\pi(S_3, \text{Pub}) = 4.78$.

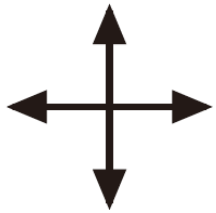
$q_\pi(S_4, \text{Facebook}) = -3.3$; $q_\pi(S_4, \text{Quit}) = -1.3$.

Example: Bellman Expectation Equation in Student MDP

Homework: Bellman Expectation Equation in Gridworld



(a)



Actions

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

What is the value function for the uniform random policy?

Bellman Expectation Equation (Matrix Form)

The Bellman expectation equation can be expressed concisely using the induced MRP,

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

with direct solution

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

Optimal Value Function

Definition

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

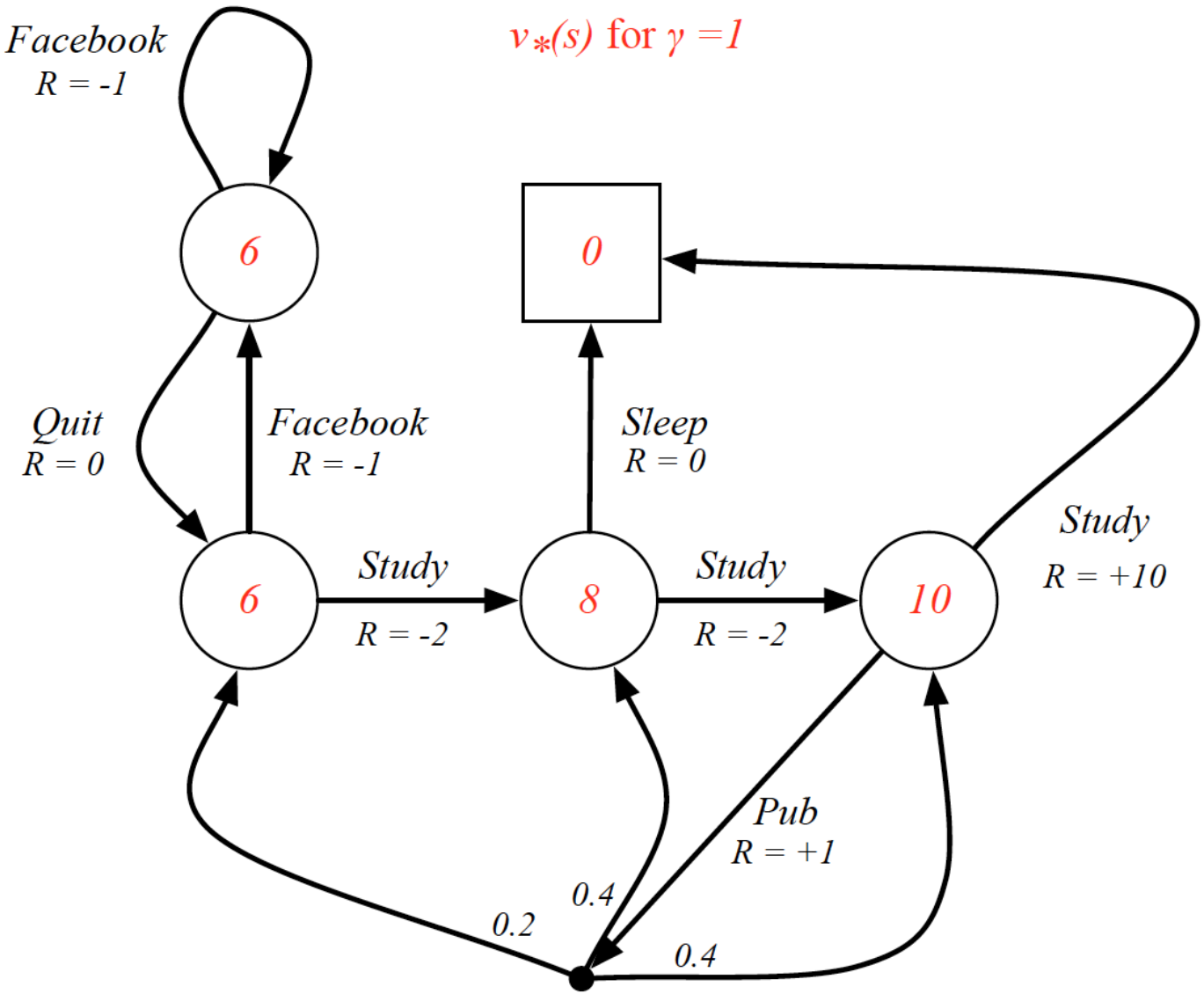
$$v_*(s) = \max_\pi v_\pi(s)$$

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

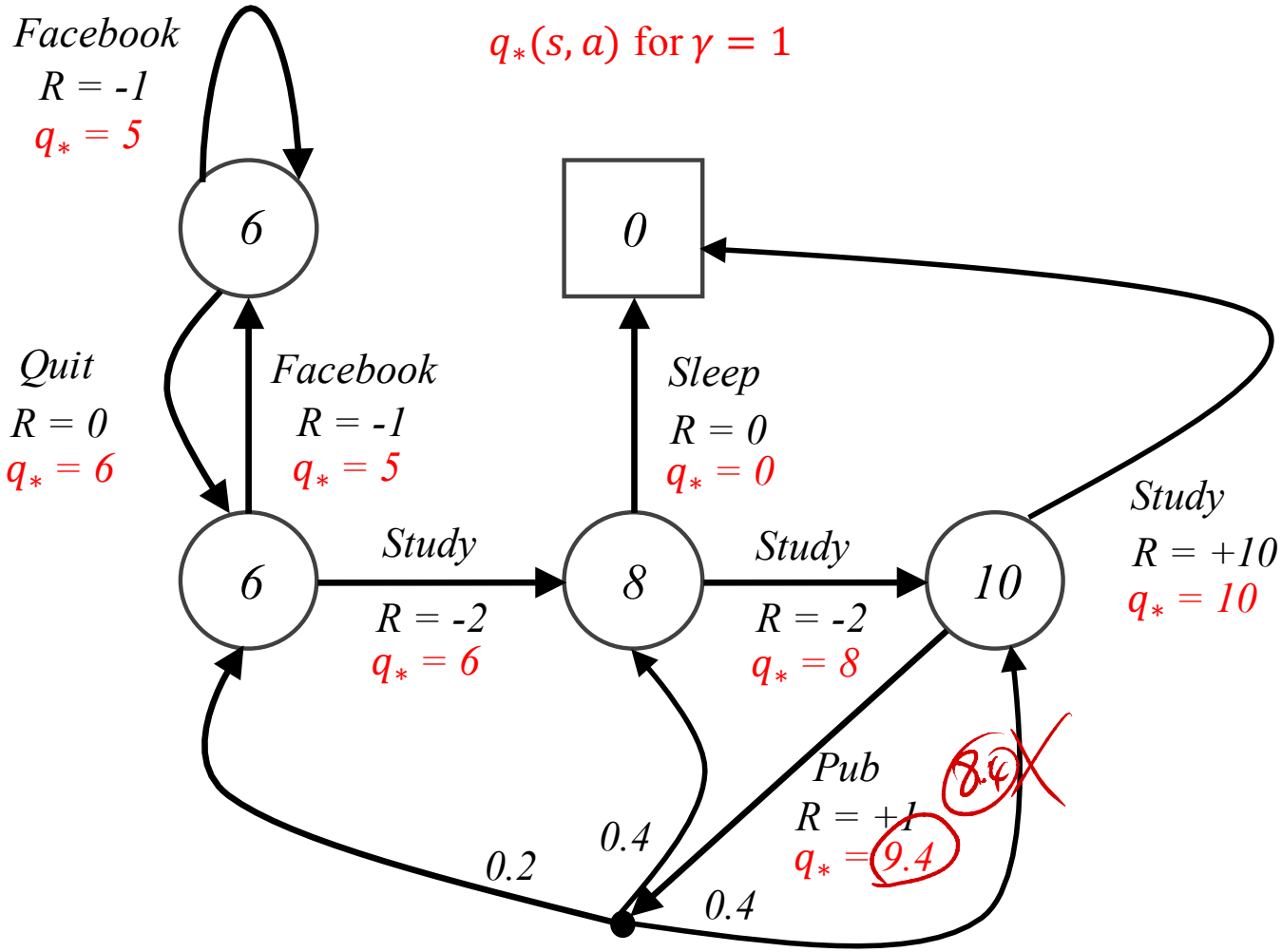
$$q_*(s, a) = \max_\pi q_\pi(s, a)$$

- The optimal value function specifies the best possible performance in the MDP.
- An MDP is "solved" when we know the optimal value function.

Example: Optimal Value Function for Student MDP



Example: Optimal Action-Value Function for Student MDP



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

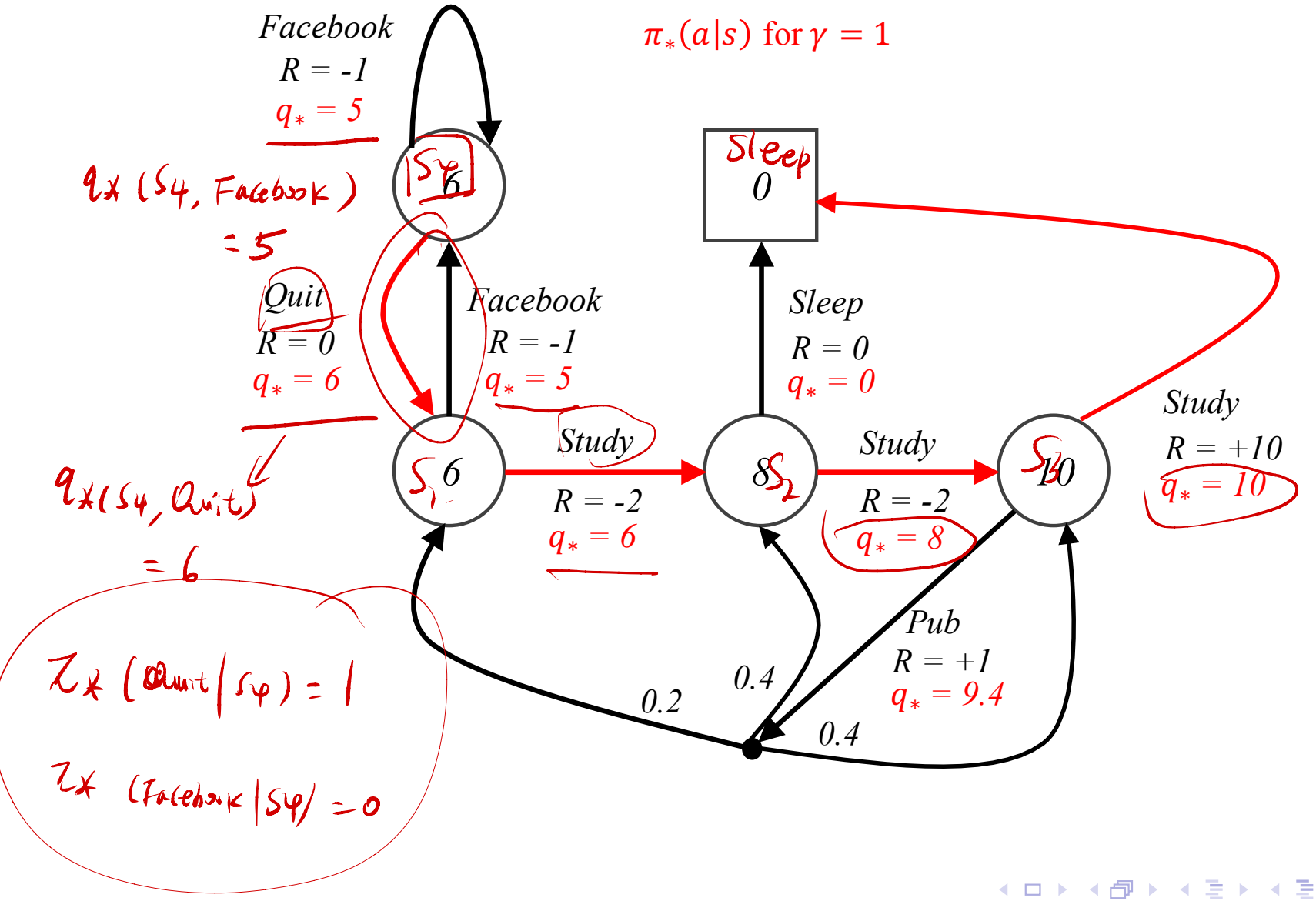
Finding an Optimal Policy

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

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

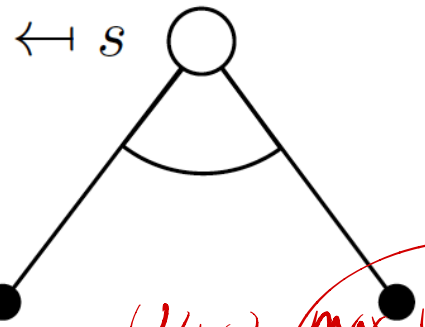
- There is always a deterministic optimal policy for any MDP
- If we know $q_*(s, a)$, we immediately have the optimal policy

Example: Optimal Policy for Student MDP



Bellman Optimality Equation for v_*

The optimal value functions are recursively related by the Bellman optimality equations:



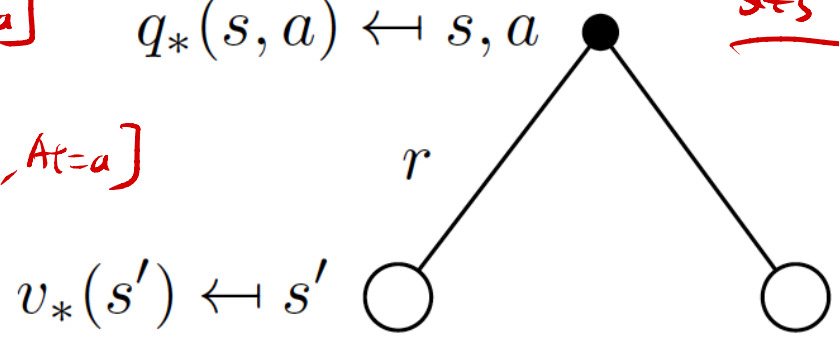
$$v_*(s) \leftarrow s$$

$$q_*(s,a) \leftarrow a$$

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

Bellman Optimality Equation for Q^*

$$q_*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a\right]$$

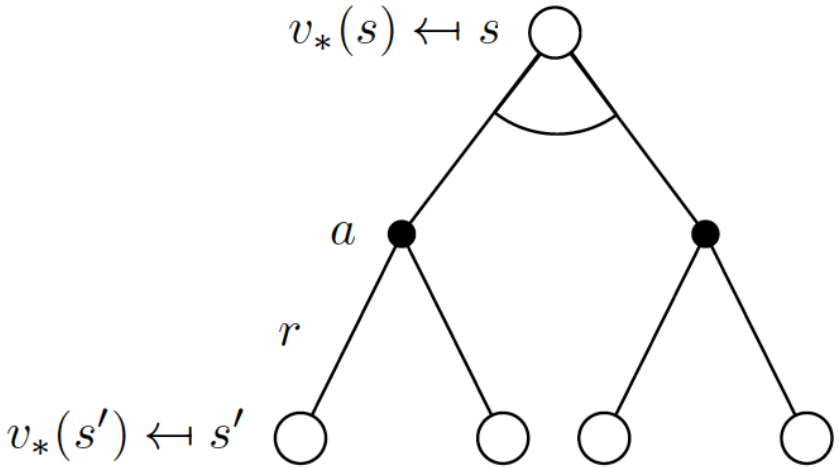


$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'}^a v_*(s')$$

Bellman Optimality Equation for V^*

2°. $V_*(s) = \max_a q_*(s,a)$; $q_*(s,a) = E[R_{t+1} + \gamma V_*(S_{t+1}) \mid S_t = s, A_t = a]$

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

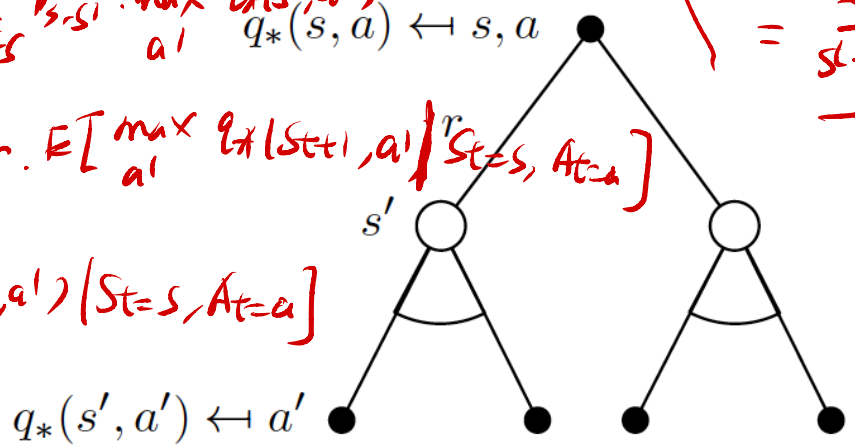


1°. $V_*(s) = \max_a q_*(s,a)$; $q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \cdot V_*(s')$

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

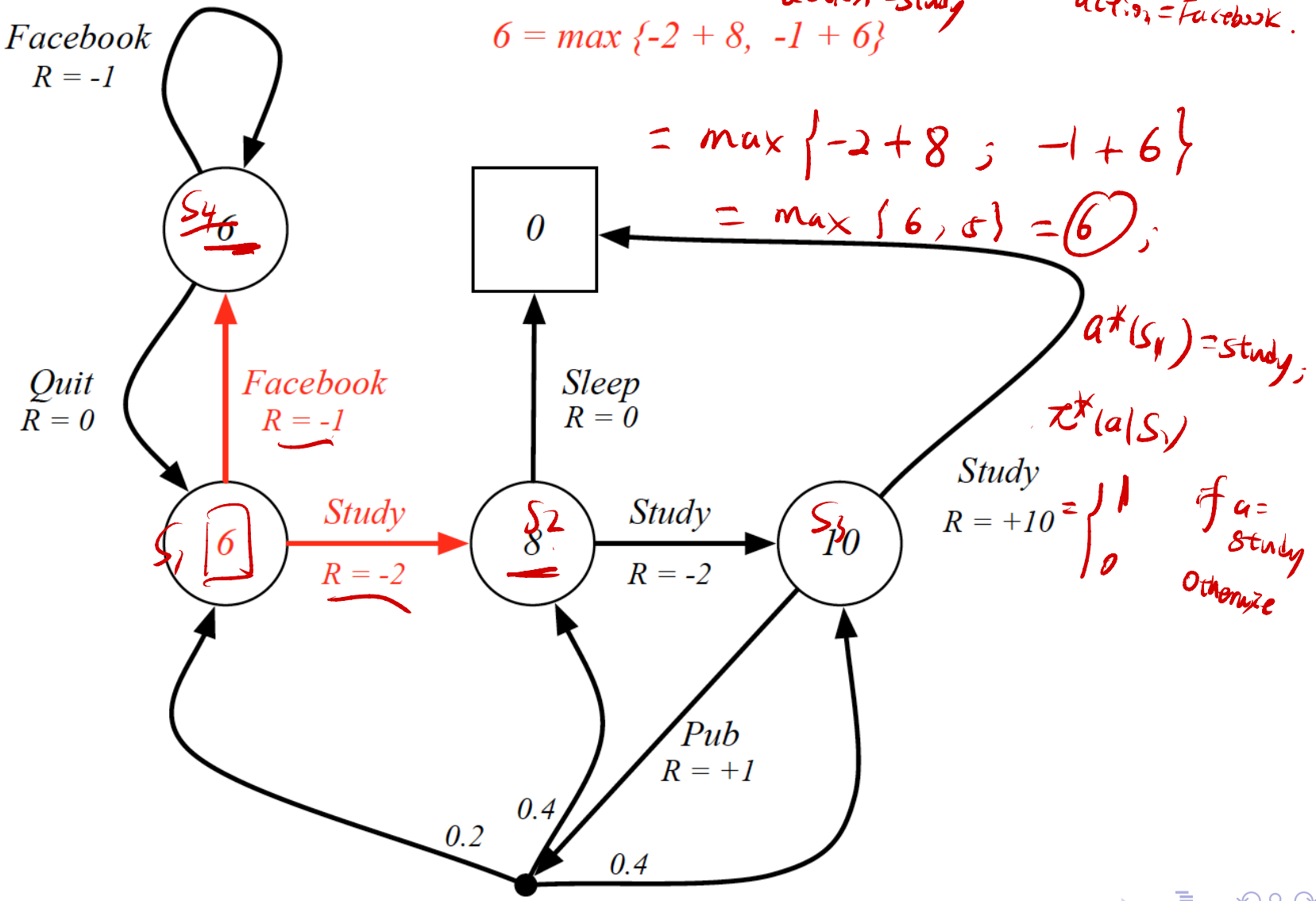
Bellman Optimality Equation for Q^*

$$q_*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \middle| S_t = s, A_t = a\right]$$



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

Example: Bellman Optimality Equation in Student MDP



Solving the Bellman Optimality Equation in General

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

Solving the Bellman Optimality Equation

- Finding an optimal policy by solving the Bellman Optimality Equation requires the following:
  - accurate knowledge of environment dynamics
  - we have enough space and time to do the computation
  - the Markov Property
- How much space and time do we need?
  - polynomial in number of states
  - BUT, number of states is often huge
  - So exhaustive sweeps of the state space are not possible

Approximation and Reinforcement Learning

ADP

OR / Control / AI

- RL methods: Approximating Bellman optimality equations
- Balancing reward accumulation and system identification (model learning) in case of unknown dynamics
- The on-line nature of reinforcement learning makes it possible to approximate optimal policies in ways that put more effort into learning to make good decisions for frequently encountered states, at the expense of less effort for infrequently encountered states.

Outline

Markov Decision Process

- Markov decision processes formally describe an environment for reinforcement learning
- Where the environment is fully observable
- i.e. The current state completely characterizes the process
- Almost all RL problems can be formalized as MDPs, e.g
  - Optimal control primarily deals with continuous MDPs
  - Partially observable problems can be converted into MDPs
  - Bandits are MDPs with one state

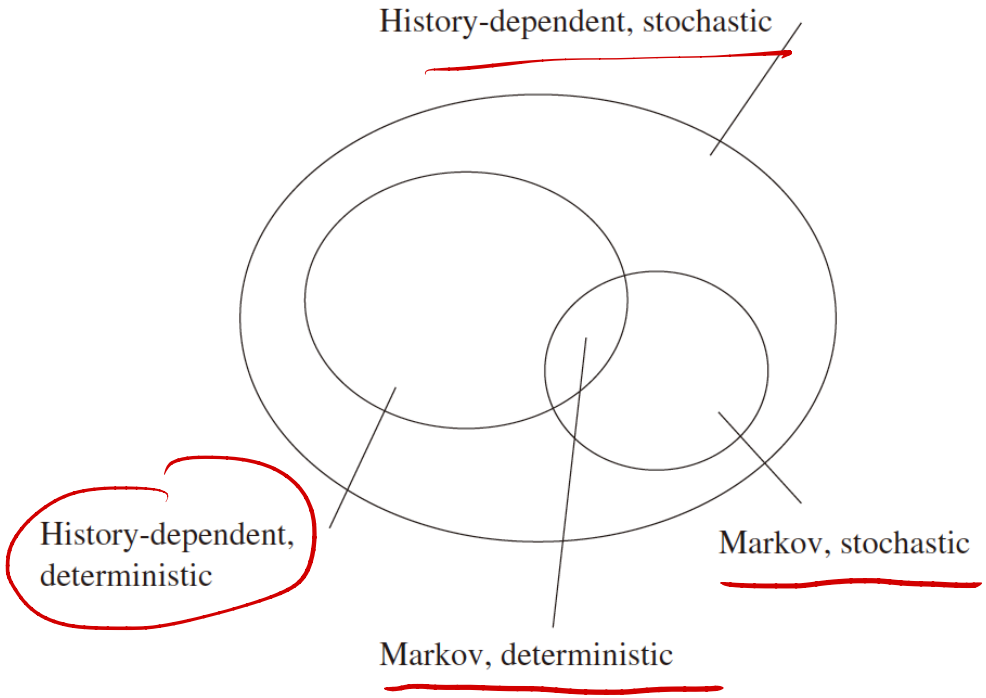
Markov Models in General

Markov Models		Do we have control over the state transitions?	
		No	Yes
Are the states completely observable?	Yes	Markov Chain	MDP Markov Decision Process
	No	HMM Hidden Markov Model	POMDP Partially Observable Markov Decision Process

Different Policy Families for MDP

Policy π_t	Deterministic	Stochastic
Markov	$s_t \longrightarrow a_t$	$a_t,\ s_t \longrightarrow [0,1]$
History-dependent	$h_t \longrightarrow a_t$	$h_t,\ s_t \longrightarrow [0,1]$

Different Policy Families for MDP



Main Performance Criteria of MDP

- The finite criterion

$$\mathbb{E}\left[r_0 + r_1 + \cdots + r_{N-1} | s_0\right]$$

- The discounted criterion

$$\mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 \cdots + \gamma^t r_t + \cdots | s_0\right]$$

- The total reward criterion

$$\mathbb{E}\left[r_0 + r_1 + \cdots + r_t + \cdots | s_0\right]$$

- The average criterion

$$\lim_{n\to\infty} \frac{1}{n} \mathbb{E}\left[r_0 + r_1 + \cdots + r_{n-1} | s_0\right]$$

Outline

Main References

- Reinforcement Learning: An Introduction (second edition), R. Sutton & A. Barto, 2018.
- RL course slides from Richard Sutton, University of Alberta.
- RL course slides from David Silver, University College London.