强化学习

第二讲：马尔可夫决策过程

教师：赵冬斌　朱圆恒

中国科学院大学
中国科学院自动化研究所

上节回顾

- 强化学习介绍
- 强化学习与其它机器学习的不同
- 强化学习发展历史
- 强化学习基本元素
- 强化学习算法分类

马尔可夫性

马尔可夫性 Markov property

- 强化学习研究的是序贯决策问题 (sequential decision)，智能体的状态会随时间发生转移

马尔可夫性

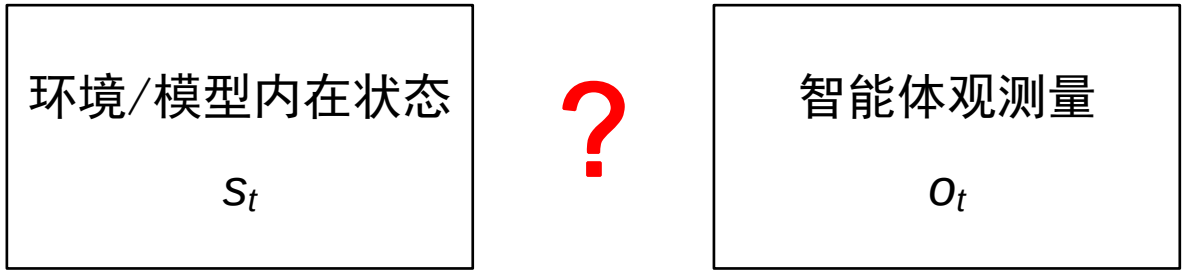
在给定现在状态及所有过去状态下，智能体未来状态的条件概率分布 仅依赖于当前状态；
换句话说，在给定现在状态时，未来状态与过去状态 (即智能体的历史轨迹) 是 条件独立的

$$\mathbb{P}[s_{t+1}|s_1, \ldots, s_t] = \mathbb{P}[s_{t+1}|s_t]$$

- 一旦当前状态确定了，历史状态都可以丢弃
- 当前状态足以决定未来状态是什么样的
- 强化学习主要研究的是具有马尔可夫性的问题.

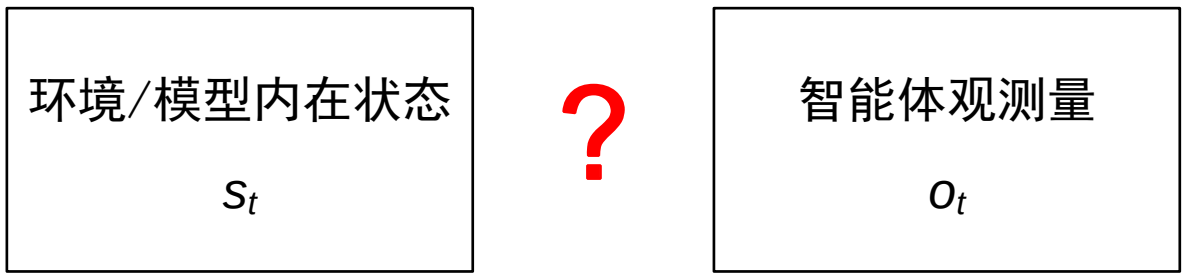
全观测/部分可观测

- 有时智能体不能完全获得环境/模型的全部状态信息，只能通过观测获得观测量
  - e.g. 牌桌上玩家只能观测自己的牌 (vs 其他玩家手里的牌)
  - e.g. 病人的血常规报告 (vs 实际的身体健康状况)



全观测/部分可观测

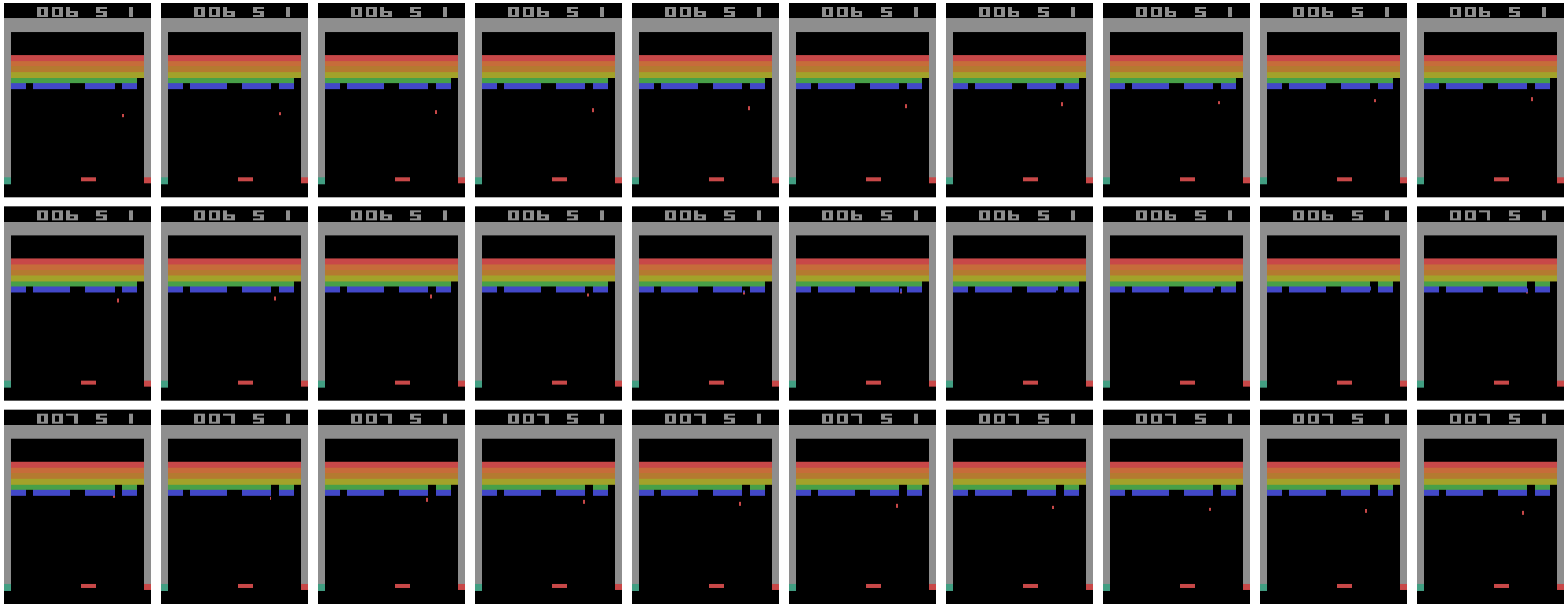
- 有时智能体不能完全获得环境/模型的全部状态信息，只能通过观测获得观测量
  - e.g. 牌桌上玩家只能观测自己的牌 (vs 其他玩家手里的牌)
  - e.g. 病人的血常规报告 (vs 实际的身体健康状况)



- 全观测 full observable：$s_t = o_t$
- 部分可观测 partial observable：$s_t \neq o_t$

部分可观测

- 有些场景下根据观测量可以推出状态，将部分可观测问题转化成全观测问题
- e.g. DQN 中利用连续 4 张游戏截图 (每张相隔 4 帧) 作为输入，获得动态信息 $(o_{t-3}, o_{t-2}, o_{t-1}, o_t) \Rightarrow s_t$



部分可观测

- 有些场景下根据观测量可以推出状态，将部分可观测问题转化成全观测问题
- e.g. DQN 中利用连续 4 张游戏截图 (每张相隔 4 帧) 作为输入，获得动态信息 $(o_{t-3}, o_{t-2}, o_{t-1}, o_t) \Rightarrow s_t$
- e.g. 红移现象
  - 在遥远的星系、类星体，星系间的气体云的光谱中观察到的红移现象，其红移增加的比例与距离成正比 $\rightarrow$ 宇宙膨胀

状态转移矩阵

- 对于一个马尔可夫状态 s 和后继状态 s'，状态转移概率定义为

$$\mathcal{P}_{ss'} = \mathbb{P}[s_{t+1} = s' | s_t = s]$$

- 状态转移矩阵 $\mathcal{P}$ 定义从所有状态 s 到所有后继状态 s' 的转移概率

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

- 矩阵每行元素的和等于 1, $\sum_i \mathcal{P}_{*i} = 1$

马尔可夫过程

马尔可夫过程 Markov Process

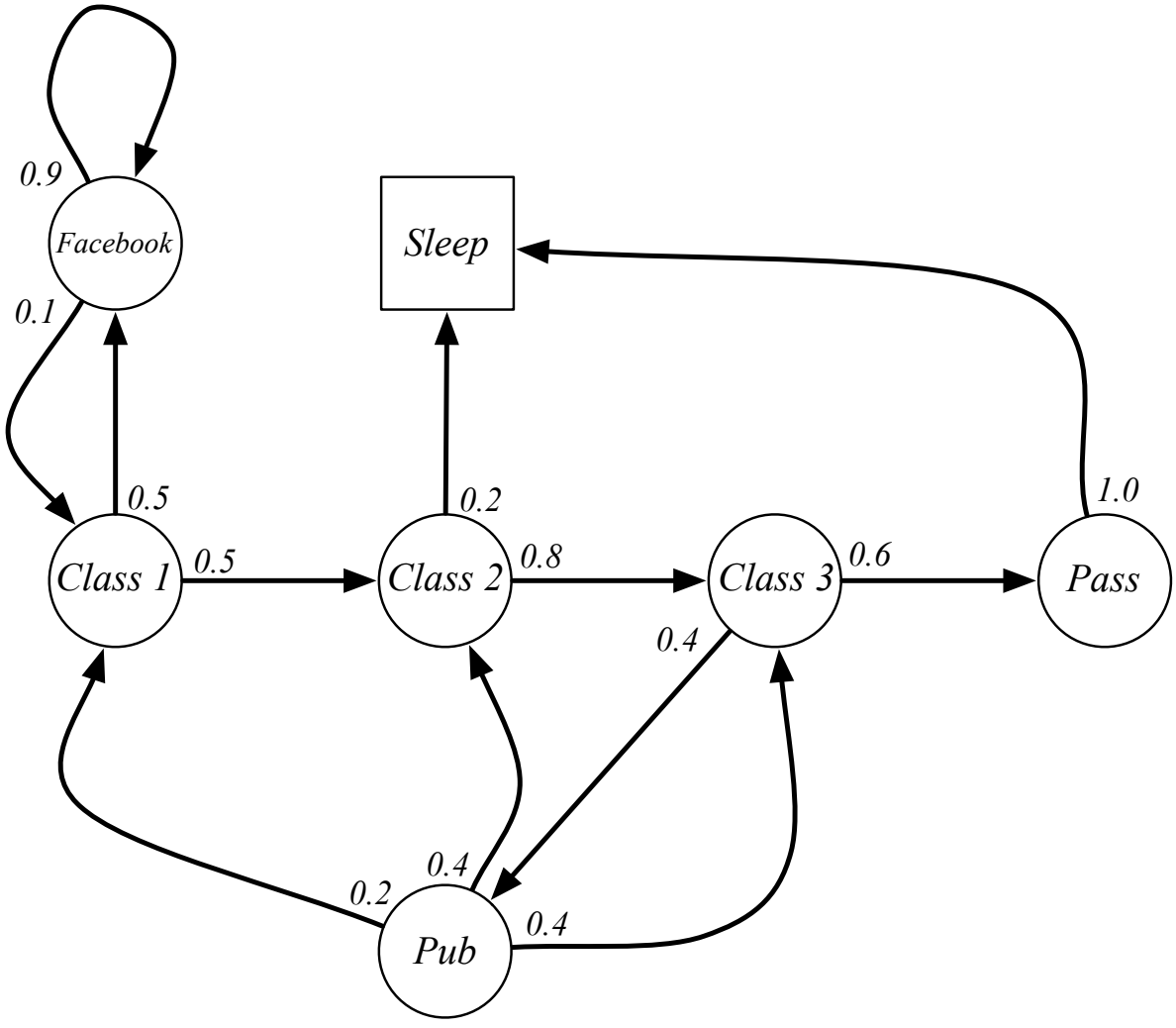
一个马尔可夫过程是一个 无记忆的随机过程，即一组具有马尔可夫性的随机状态序列 $s_1, s_2, \ldots$

定义

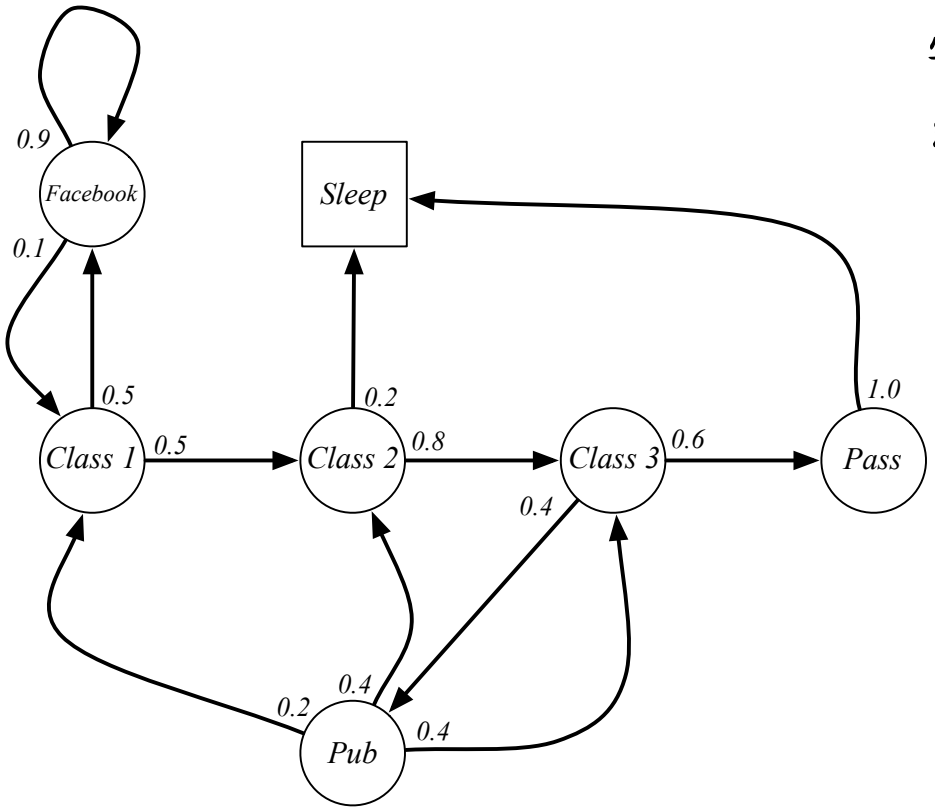
一个马尔可夫过程（或马尔可夫链）可以用一组 $\langle \mathcal{S}, \mathcal{P} \rangle$ 表示

- $\mathcal{S}$ 是（有限）状态集
- $\mathcal{P}$ 是状态转移概率矩阵 $\mathcal{P}_{ss'} = \mathbb{P}[s_{t+1} = s' | s_t = s]$

举例：学生 MP


0.9
Facebook
0.1
Sleep
0.5
0.2
1.0
Class 1
0.5
Class 2
0.8
Class 3
0.6
Pass
0.4
0.4
0.2
0.4
Pub

举例：学生 MP 轨迹

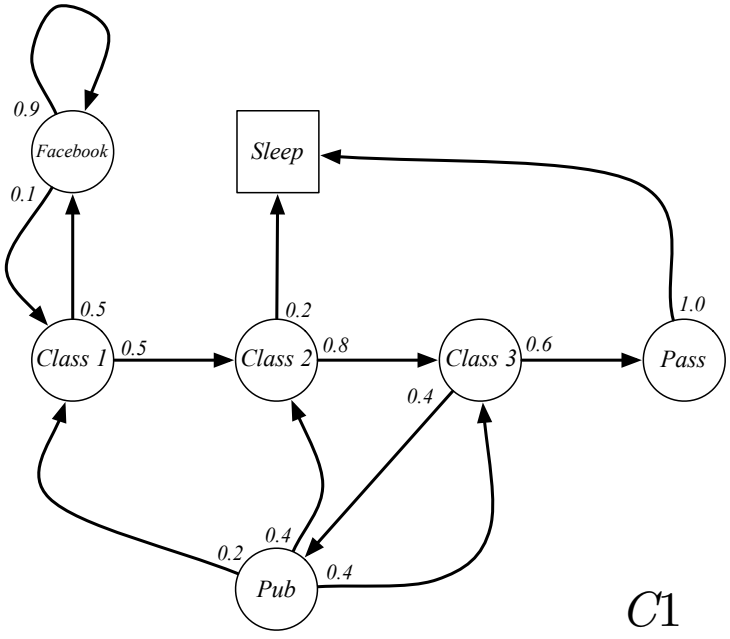


学生 MP 从初始状态 $s_1 = C_1$ 出发可能发生的 轨迹

$$s_1, s_2, \ldots, s_T$$

- 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

举例：学生 MP 转移矩阵



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

马尔可夫奖励过程

马尔可夫奖励过程 Markov Reward Process

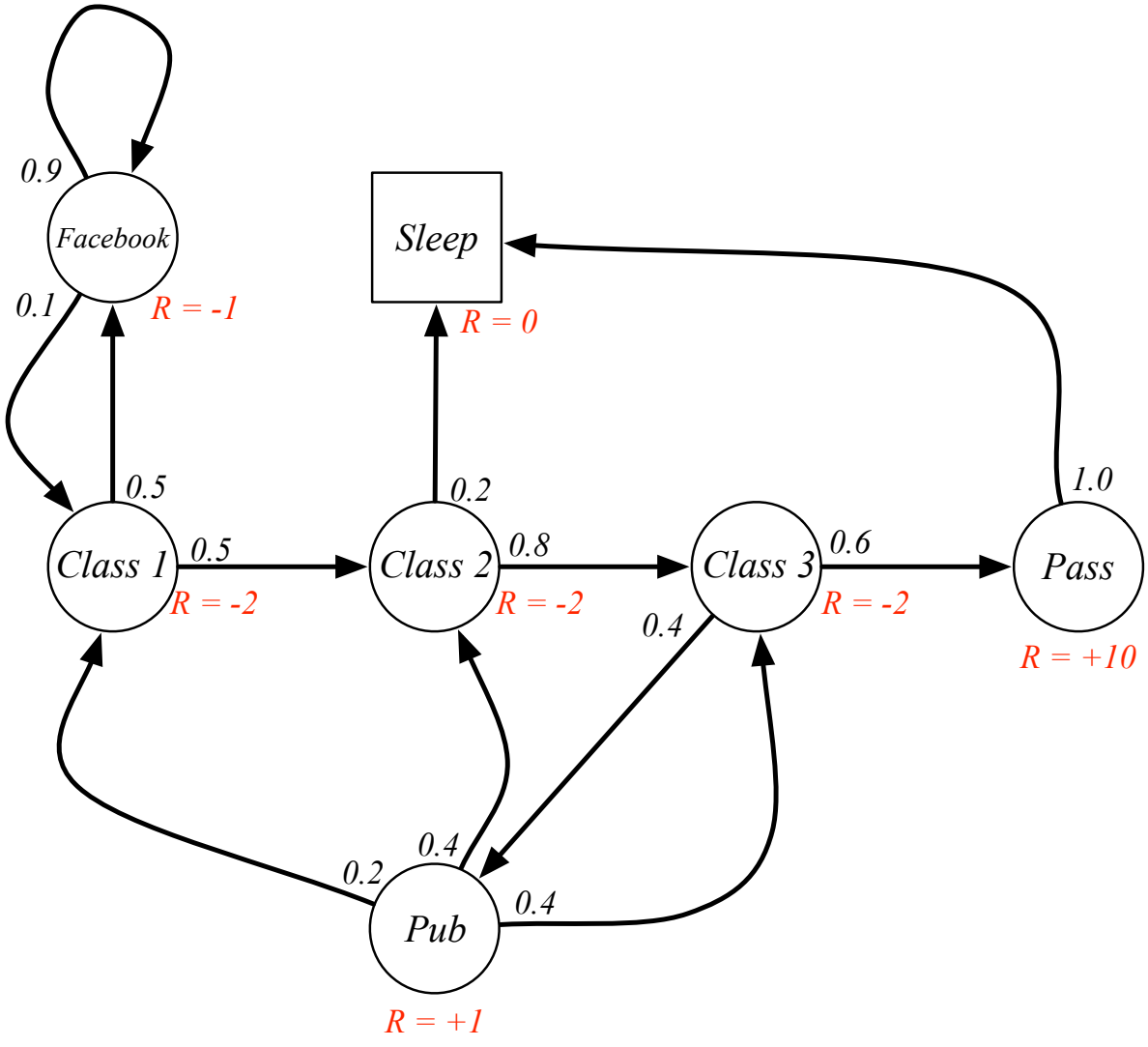
- 一个马尔可夫奖励过程是一个马尔可夫链加上奖励

定义

一个马尔可夫奖励过程 由一组 $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 构成

- $\mathcal{S}$ 是一组有限状态集
- $\mathcal{P}$ 是状态转移概率矩阵 $\mathcal{P}_{ss'} = \mathbb{P}[s_{t+1} = s' | s_t = s]$
- $\mathcal{R}$ 是奖励函数，$\mathcal{R}_s = \mathcal{R}(s) = \mathbb{E}[r_{t+1} | s_t = s]$
- γ 是折扣因子，$\gamma \in [0, 1]$

举例：学生 MRP


0.9
Facebook
0.1
R = -1
Sleep
R = 0
0.5
0.2
1.0
Class 1
0.5
R = -2
Class 2
0.8
R = -2
Class 3
0.6
R = -2
Pass
R = +10
0.4
0.4
0.2
0.4
Pub
R = +1

回报 Return 和价值 Value

回报定义

回报 G_t 代表在 t 时刻之后轨迹的累加奖励

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

价值定义

一个马尔可夫奖励过程的 价值 V 等于从状态 s 出发的期望回报

$$V(s) = \mathbb{E}[G_t | s_t = s]$$

- 回报对应某一 具体 的轨迹
- 价值代表所有轨迹的 期望

为什么要用折扣因子

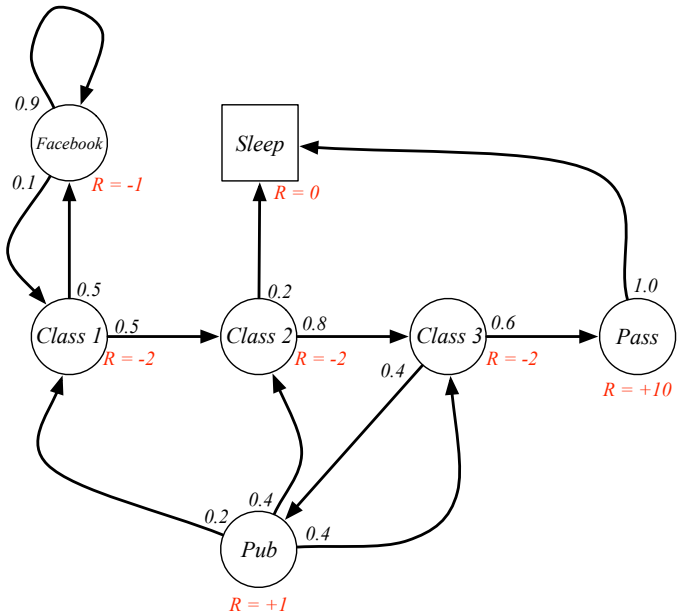
大部分的马尔可夫奖励和决策过程都使用折扣因子 $0 < \gamma < 1$，为什么？

- 如果想要调整奖励的重要性，数学上很方便
- 在连续的 MDPs 问题中能避免无穷回报
- 重视近期的奖励，但是有可能会忽视未来奖励
- 自然界的人类或动物行为模式更倾向于近期奖励
- 有时候也会使用无折扣的马尔可夫奖励过程 (即 $\gamma = 1$)，例如所有的轨迹序列都有终止状态，e.g. 围棋

举例：学生 MRP 回报

从状态 $s_1 = C1$ 出发学生马尔可夫奖励过程可能的回报，$\gamma = \frac{1}{2}$

$$G_1 = r_2 + \gamma r_3 + \cdots + \gamma^{T-2} r_T$$

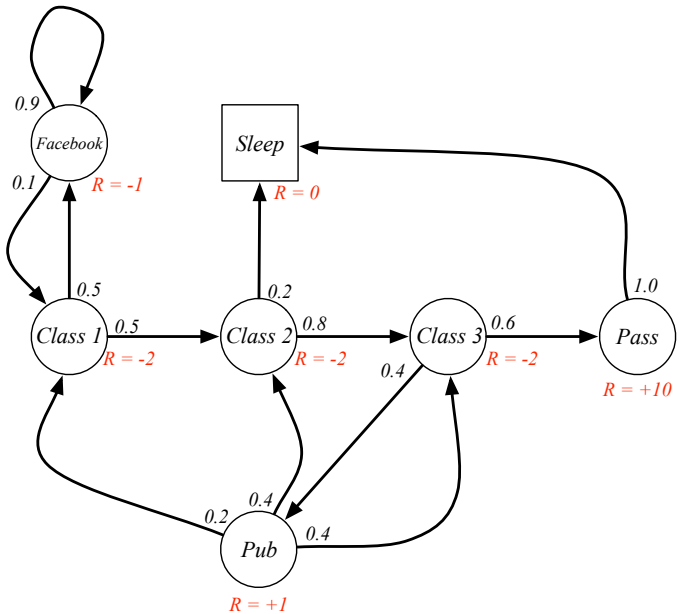


- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

举例：学生 MRP 回报

从状态 $s_1 = C1$ 出发学生马尔可夫奖励过程可能的回报，$\gamma = \frac{1}{2}$

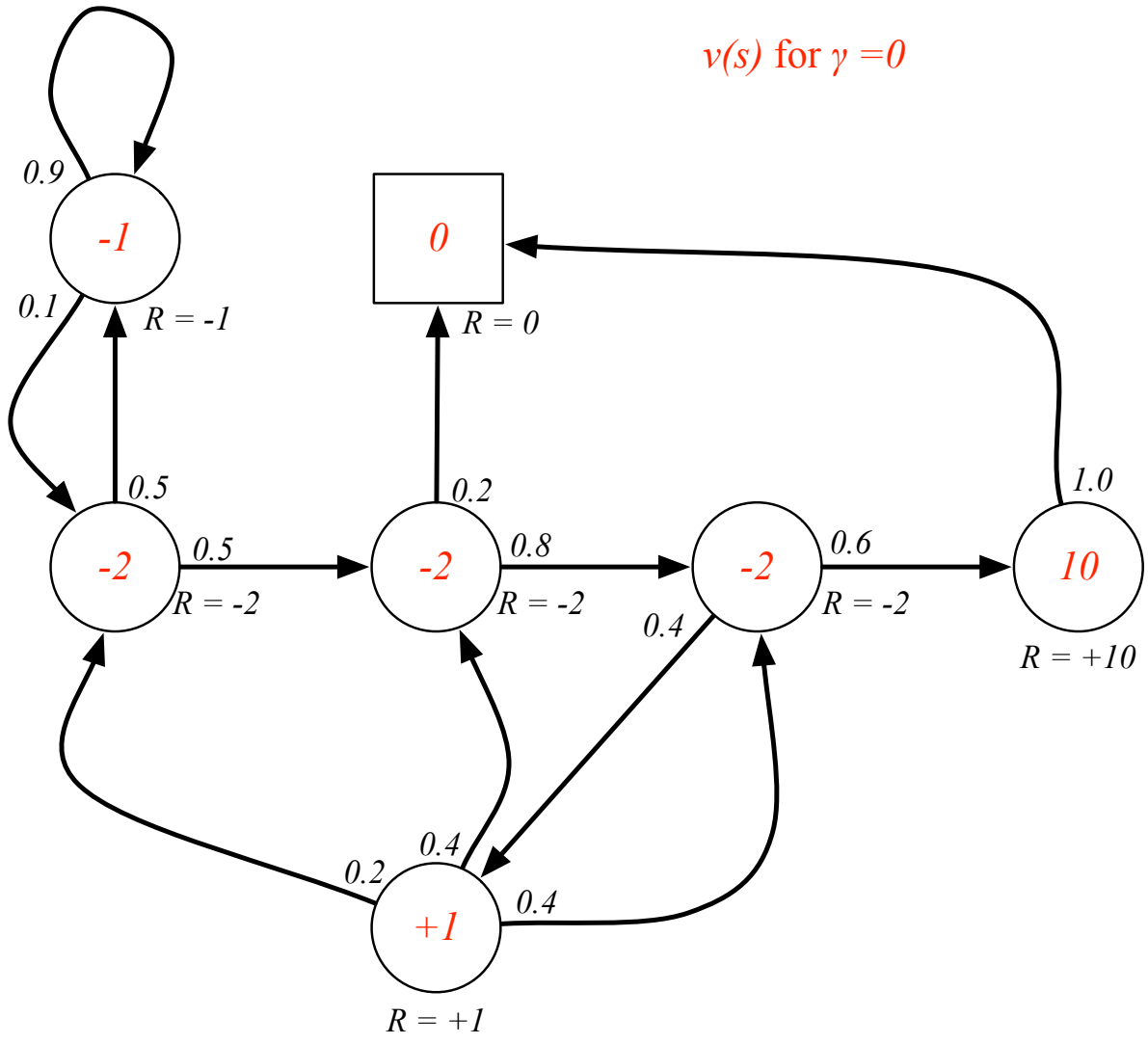
$$G_1 = r_2 + \gamma r_3 + \cdots + \gamma^{T-2} r_T$$



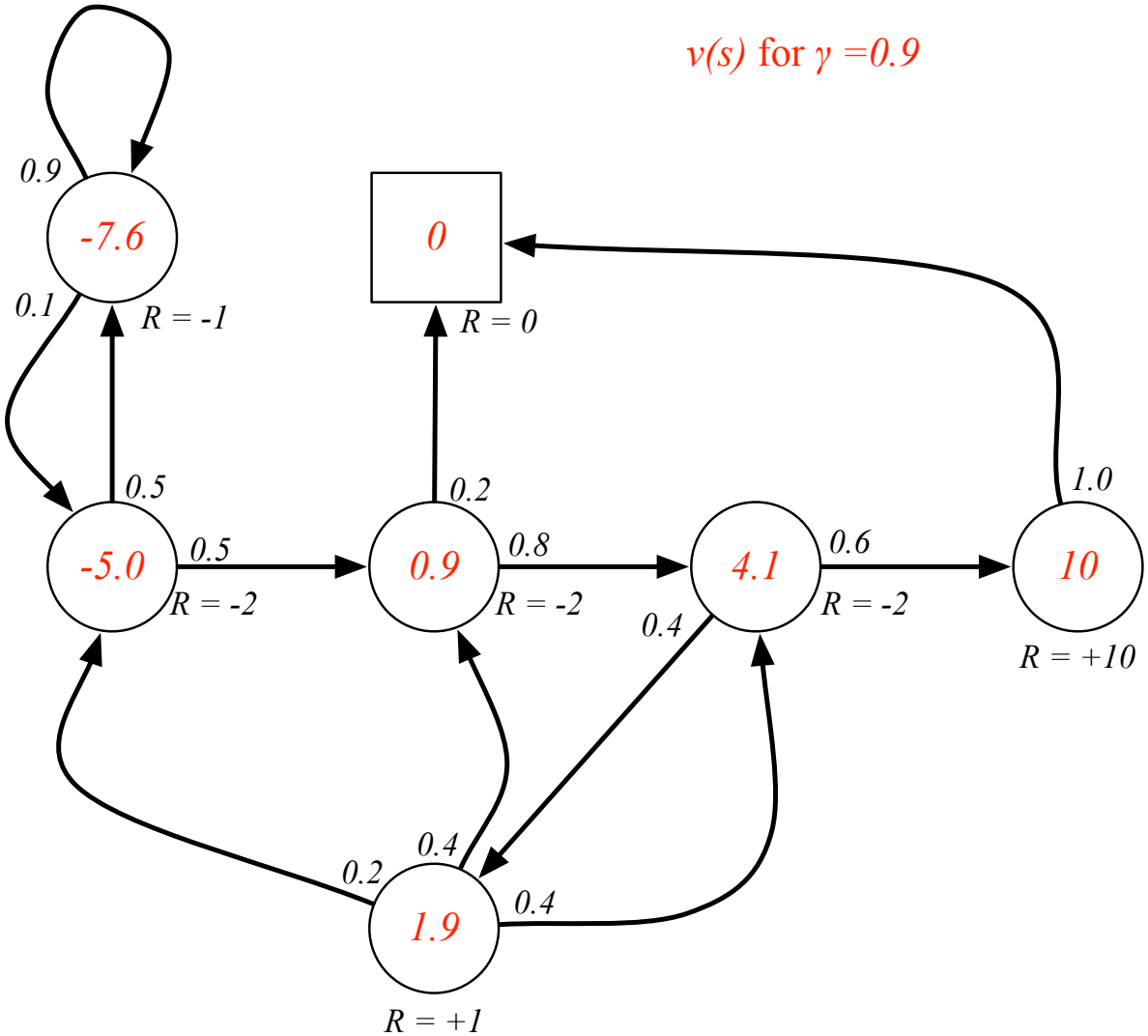
- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

- $G_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8} = -2.25$
- $G_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} = -3.125$
- $G_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \cdots = -3.41$
- $G_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \cdots = -3.20$

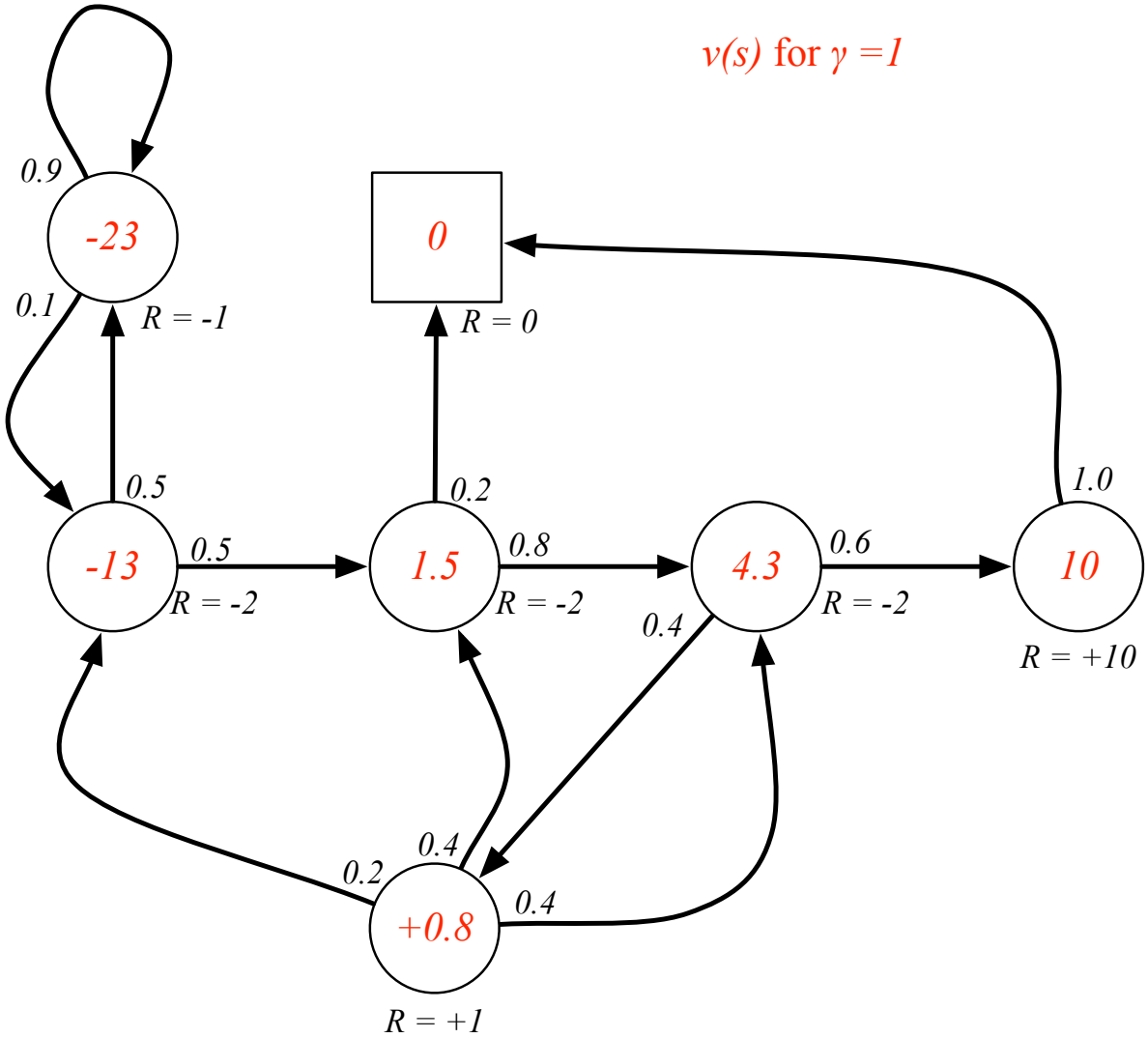
举例：学生 MRP 的价值 (1)



举例：学生 MRP 的价值 (2)



举例：学生 MRP 的价值 (3)



MRPs 的贝尔曼方程 Bellman equation

将价值函数拆分成两部分

$$
\begin{aligned}
V(s) &= \mathbb{E}[G_t | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s_t = s]
\end{aligned}
$$

MRPs 的贝尔曼方程 Bellman equation

将价值函数拆分成两部分

- 瞬间奖励 r_{t+1}

$$
\begin{aligned}
V(s) &= \mathbb{E}[G_t | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma \underline{(r_{t+2} + \gamma r_{t+3} + \ldots)} | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma \underline{G_{t+1}} | s_t = s]
\end{aligned}
$$

MRPs 的贝尔曼方程 Bellman equation

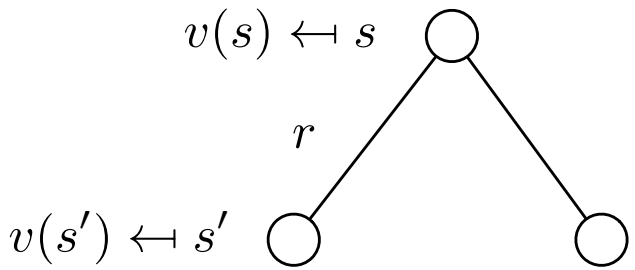
将价值函数拆分成两部分

- 瞬间奖励 r_{t+1}
- 后继状态的折扣价值 $\gamma V(s_{t+1})$

$$
\begin{aligned}
V(s) &= \mathbb{E}[G_t | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma \underline{(r_{t+2} + \gamma r_{t+3} + \ldots)} | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma \underline{G_{t+1}} | s_t = s] \\
&= \mathbb{E}[r_{t+1} + \gamma V(s_{t+1}) | s_t = s]
\end{aligned}
$$

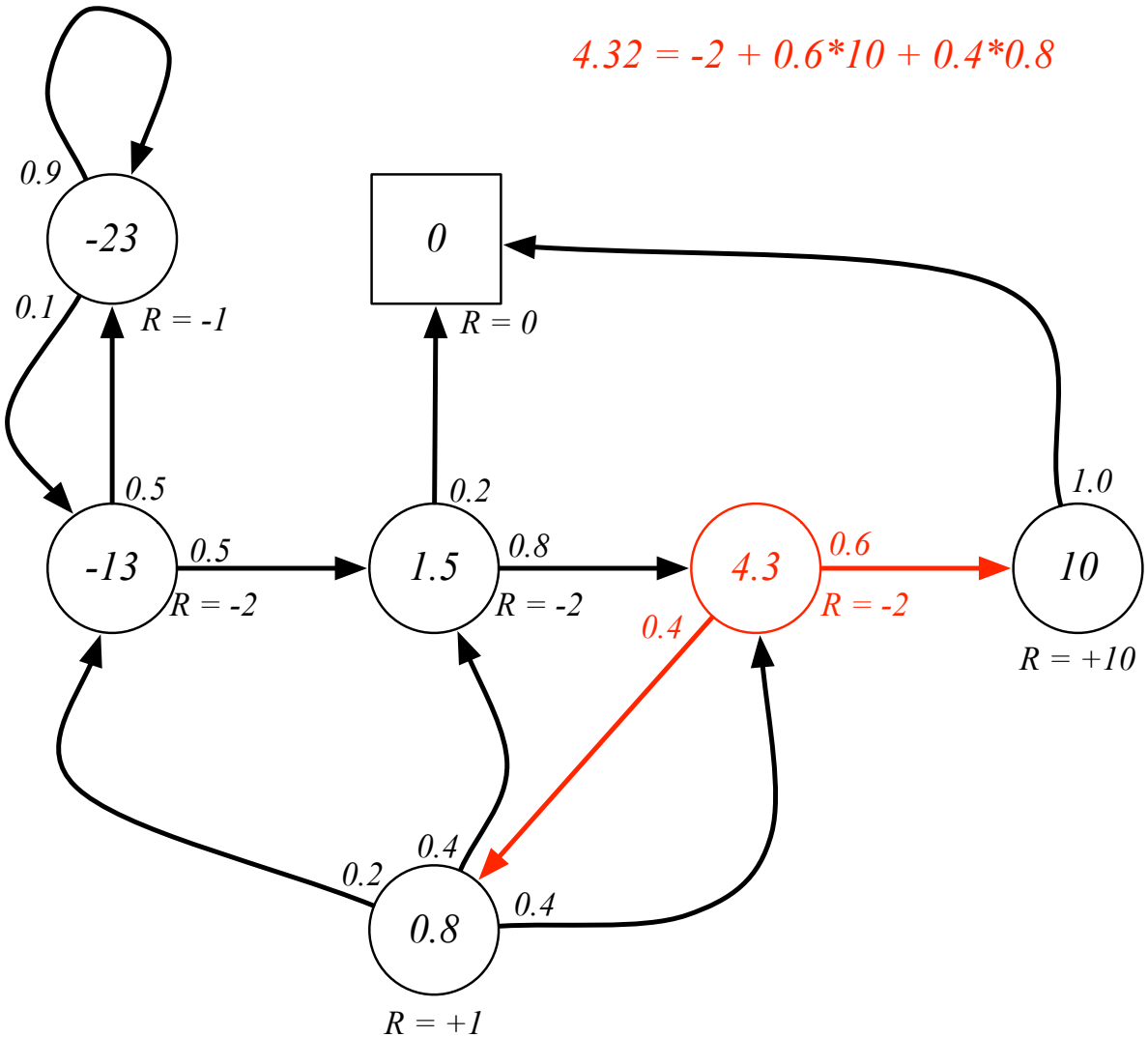
MRPs 的贝尔曼方程 (2)

$$V(s) = \mathbb{E}[r_{t+1} + \gamma V(s_{t+1}) | s_t = s]$$



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

举例: 学生 MRP 的贝尔曼方程



矩阵形式的贝尔曼方程

- 对 $\mathcal{S}$ 中所有状态 $\{S_1, \ldots, S_n\}$，贝尔曼方程都有

$$
\begin{aligned}
V(S_1) &= \mathcal{R}(S_1) + \gamma \left[\mathcal{P}_{S_1 S_1} V(S_1) + \cdots + \mathcal{P}_{S_1 S_n} V(S_n) \right] \\
&\vdots \\
V(S_n) &= \mathcal{R}(S_n) + \gamma \left[\mathcal{P}_{S_n S_1} V(S_1) + \cdots + \mathcal{P}_{S_n S_n} V(S_n) \right]
\end{aligned}
$$

矩阵形式的贝尔曼方程

- 对 $\mathcal{S}$ 中所有状态 $\{S_1, \ldots, S_n\}$，贝尔曼方程都有

$$V(S_1) = \mathcal{R}(S_1) + \gamma \left[\mathcal{P}_{S_1 S_1} V(S_1) + \cdots + \mathcal{P}_{S_1 S_n} V(S_n)\right]$$

$$\vdots$$

$$V(S_n) = \mathcal{R}(S_n) + \gamma \left[\mathcal{P}_{S_n S_1} V(S_1) + \cdots + \mathcal{P}_{S_n S_n} V(S_n)\right]$$

- 用矩阵形式表示贝尔曼方程

$$\begin{bmatrix} V(S_1) \\ \vdots \\ V(S_n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}(S_1) \\ \vdots \\ \mathcal{R}(S_n) \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{S_1 S_1} & \cdots & \mathcal{P}_{S_1 S_n} \\ \vdots & \ddots & \vdots \\ \mathcal{P}_{S_n S_1} & \cdots & \mathcal{P}_{S_n S_n} \end{bmatrix} \begin{bmatrix} V(S_1) \\ \vdots \\ V(S_n) \end{bmatrix}$$

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

求解贝尔曼方程

- 贝尔曼方程是 线性方程
- 可以直接求解:

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

- n 个状态下的计算复杂度 $O(n^3)$
- 直接求解法只适用于问题较小 (状态空间小) 的 MRPs
- 对于大规模的 MRPs 问题, 可以使用迭代或基于数据的方法, 例如
  - 动态规划
  - 蒙特卡洛估计
  - 时间差分学习

马尔可夫决策过程

马尔可夫决策过程 Markov Decision Process, MDP

- 马尔可夫决策过程是马尔可夫奖励过程加上智能体的决策
- 问题的所有状态都具有马尔可夫性

定义

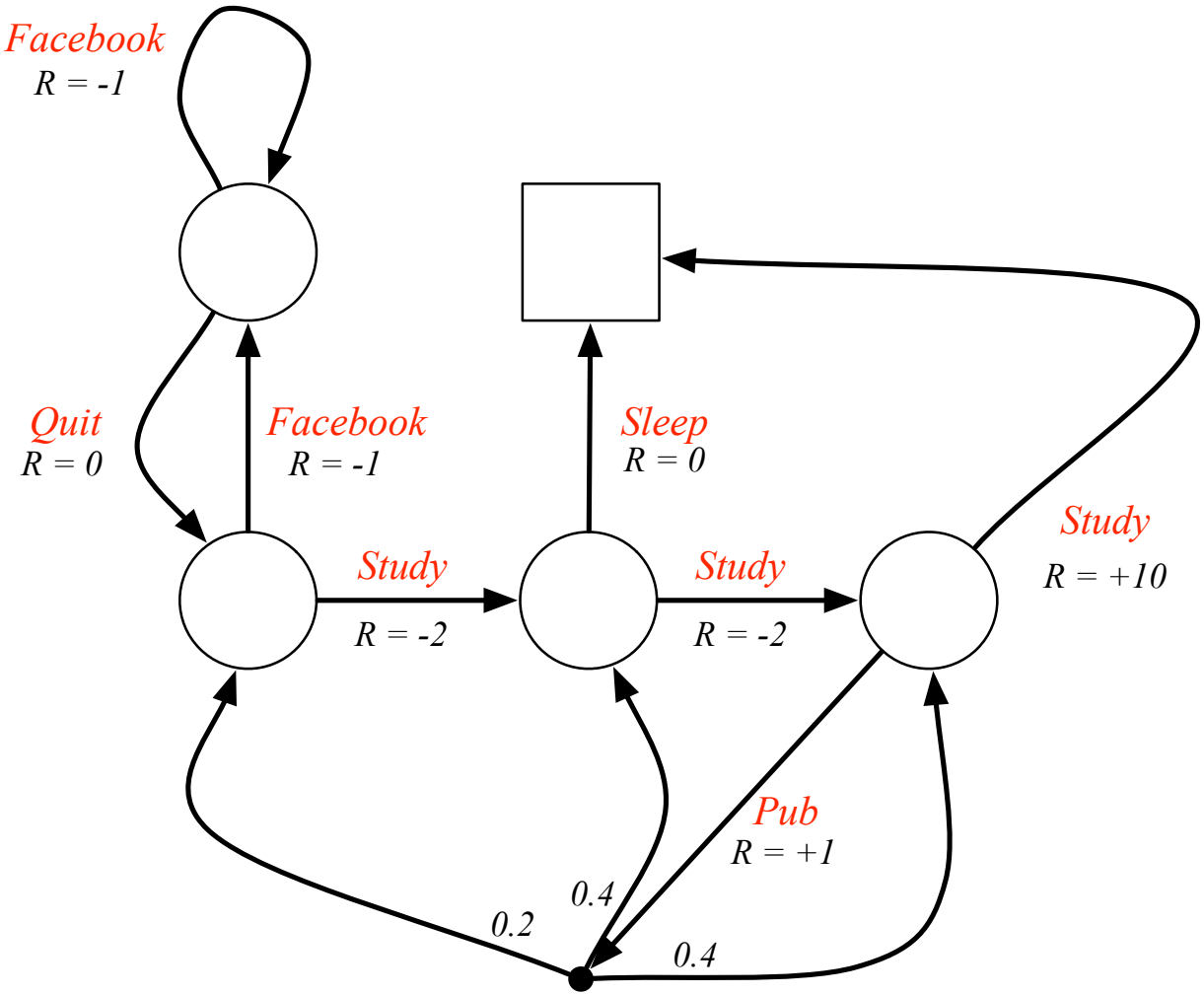
一个马尔可夫决策过程 由 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 组成

- $\mathcal{S}$ 是有限状态集
- $\mathcal{A}$ 是有限动作集
- $\mathcal{P}$ 是状态转移概率矩阵

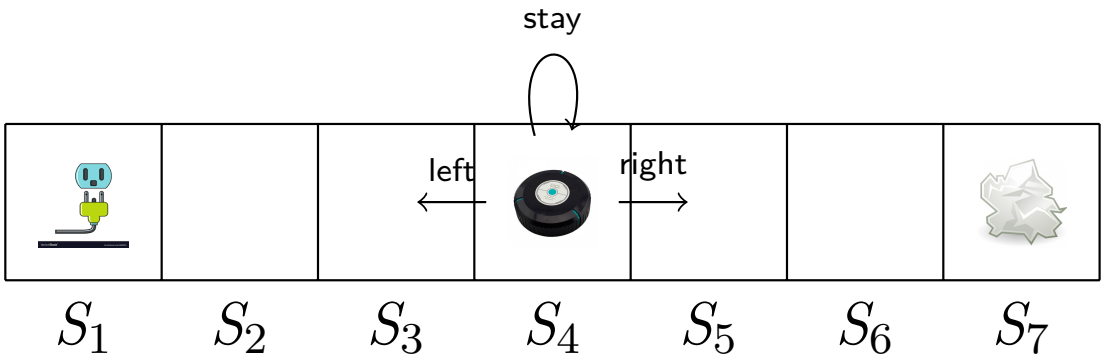
$$\mathcal{P}_{ss'}^{a} = \mathbb{P}[s_{t+1} = s' | s_t = s, a_t = a]$$

- $\mathcal{R}$ 是奖励函数, $\mathcal{R}_s^a = \mathbb{E}[r_{t+1} | s_t = s, a_t = a]$
- γ 是折扣因子 $\gamma \in [0, 1]$

举例: 学生 MDP

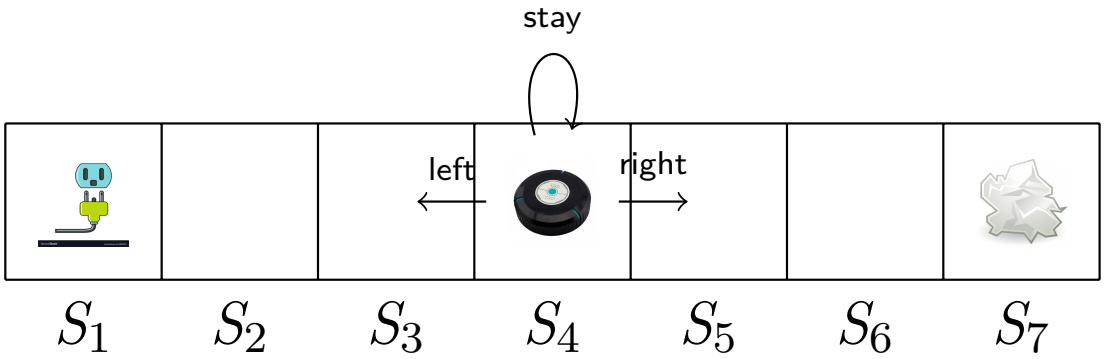

Facebook
R = -1
Quit
R = 0
Facebook
R = -1
Sleep
R = 0
Study
R = -2
Study
R = -2
Study
R = +10
Pub
R = +1
0.2
0.4
0.4

练习: 机器人 MDP(1)



- 扫地机器人
  1. 在一条直线上行走, 所处的位置分别是 $\{S_1, S_2, \ldots, S_7\}$
  2. S_1 是充电区域, 能够获得奖励 $+1$; S_7 上有一团废纸, 走上去清扫能获得奖励 $+10$; 其它位置奖励 0
  3. 可以选择向左或向右行走
  4. 由于机械精度原因, 选择的动作只有 80% 的可能正确执行, 10% 保持不动, 10% 向反方向移动
  5. 超出边界的动作 90% 概率留在边界, 10% 概率反向移动

练习: 机器人 MDP(2)



- MDP 表示
  - $\mathcal{S}$: $\{S_1, S_2, \ldots, S_7\}$
  - $\mathcal{A}$: $\{A_{left}, A_{right}\}$
  - $\mathcal{R}$: $\mathcal{R}(S_1) = +1, \mathcal{R}(S_7) = +10, \mathcal{R}(S_i) = 0, i = 2, \ldots, 6$
  - G: $r_1 + \gamma r_2 + \ldots$

练习: 机器人 MDP(3)

状态转移

- $s = S_2, \ldots, S_6$

$$\mathcal{P}_{ss'}^{a} = \begin{cases} 0.8 & (a = A_{left}, s' = s-1) || (a = A_{right}, s' = s+1) \\ 0.1 & s' = s \\ 0.1 & (a = A_{left}, s' = s+1) || (a = A_{right}, s' = s-1) \end{cases}$$

- $s = S_1$

$$\mathcal{P}_{S_1,S_1}^{A_{left}} = 0.9, \ \mathcal{P}_{S_1,S_2}^{A_{left}} = 0.1$$

$$\mathcal{P}_{S_1,S_1}^{A_{right}} = 0.2, \ \mathcal{P}_{S_1,S_2}^{A_{right}} = 0.8$$

- $s = S_7$

$$\mathcal{P}_{S_7,S_6}^{A_{left}} = 0.8, \ \mathcal{P}_{S_7,S_7}^{A_{left}} = 0.2$$

$$\mathcal{P}_{S_7,S_6}^{A_{right}} = 0.1, \ \mathcal{P}_{S_7,S_7}^{A_{right}} = 0.9$$

策略与价值

策略

定义

策略 π 是状态到动作的一种分布

$$\pi(a|s) = \mathbb{P}[a_t = a|s_t = s]$$

- 一个策略定义了一个智能体的行为
  - 可以是确定性的，即 $\pi(\cdot|s)$ 只在某一个动作下概率是 1，其它动作概率是零，也可写成 $\pi(s) = a$
  - 或是随机性的，在多个动作下概率都大于零
- MDP 问题动作的选择只取决于当前状态：$a_t \sim \pi(s_t)$
  - 与历史无关，马尔可夫性
- 我们统一 $\pi(s)$ 代表状态 s 下动作的概率分布，$\pi(s, a)$ 或 $\pi(a|s)$ 代表状态 s 下选择动作 a 的概率值

- 给定一个 MDP 问题 $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 和策略 π
- 智能体的状态轨迹 $s_1, s_2, \ldots$ 是一个 **马尔可夫过程** $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$
- 状态和奖励轨迹 $s_1, r_2, s_2, r_3, \ldots$ 是一个 **马尔可夫奖励过程** $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$
- 其中

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

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

价值

定义

MDP 的 (状态) 价值 $V_\pi(s)$ 定义为从状态 s 出发, 在策略 π 作用下的期望回报

$$V_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s]$$

其中

$$a_k \sim \pi(s_k), s_{k+1} \sim \mathcal{P}(s_k, a_k), r_{k+1} \sim \mathcal{R}(s_k, a_k), \forall k \geq t$$

定义

动作 -价值 $Q_\pi(s, a)$ 是智能体从状态 s 出发, 首先执行动作 a, 然后按照策略 π 的期望回报

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a]$$

- $V_\pi(s)$ 的轨迹

$$s_t = s, a_t \sim \pi, r_{t+1} \sim \mathcal{R}, s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi, \ldots$$

- $Q_\pi(s, a)$ 的轨迹

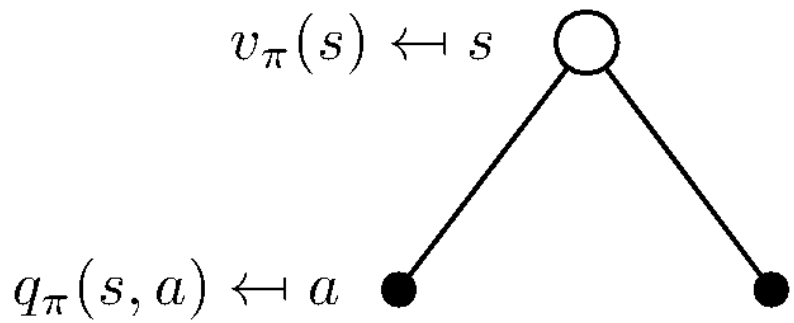
$$s_t = s, a_t = a, r_{t+1} \sim \mathcal{R}, s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi, \ldots$$

- $V_\pi(s)$ 的轨迹

$$s_t = s, a_t \sim \pi, r_{t+1} \sim \mathcal{R}, s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi, \ldots$$

- $Q_\pi(s, a)$ 的轨迹

$$s_t = s, a_t = a, r_{t+1} \sim \mathcal{R}, s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi, \ldots$$



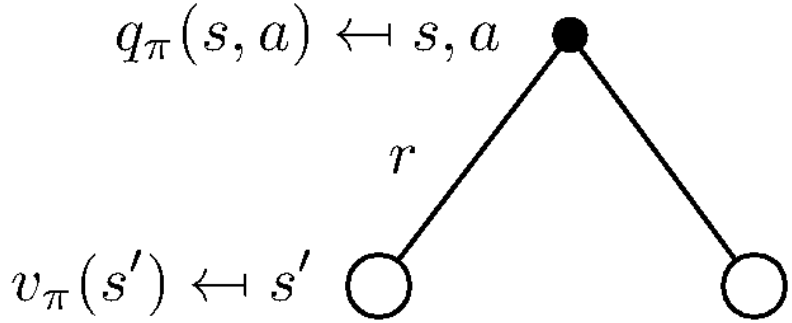
$$\boxed{V_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q_\pi(s, a)}$$

- $Q_\pi(s,a)$ 的轨迹

$$s_t = s, a_t = a, r_{t+1} \sim \mathcal{R}, s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi, r_{t+2} \sim \mathcal{R}, \ldots$$

- $Q_\pi(s,a)$ 的轨迹

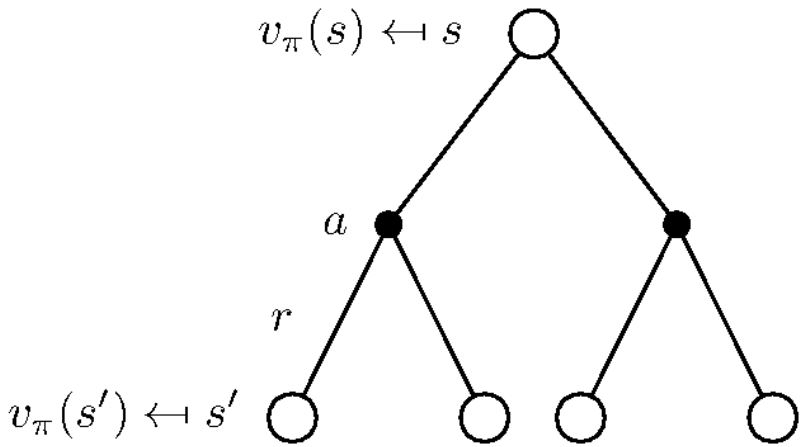
$$s_t = s, a_t = a, r_{t+1} \sim \mathcal{R}, s_{t+1} \sim \mathcal{P}, \underbrace{a_{t+1} \sim \pi, r_{t+2} \sim \mathcal{R}, \ldots}_{V(s_{t+1})}$$



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

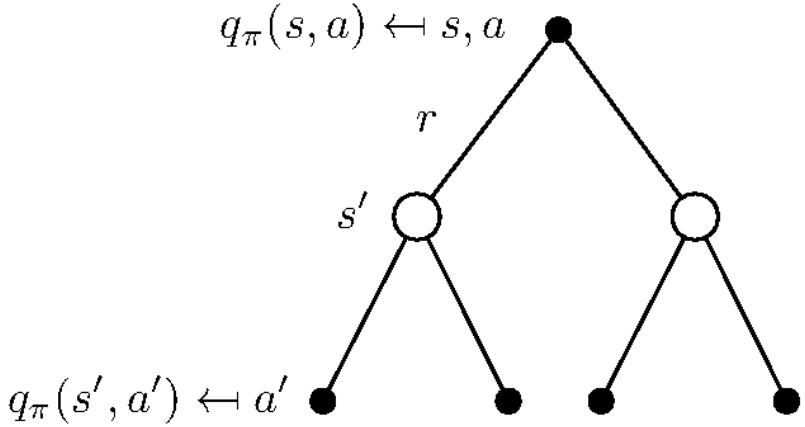
贝尔曼期望方程 Bellman Expectation Eq, V_π

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



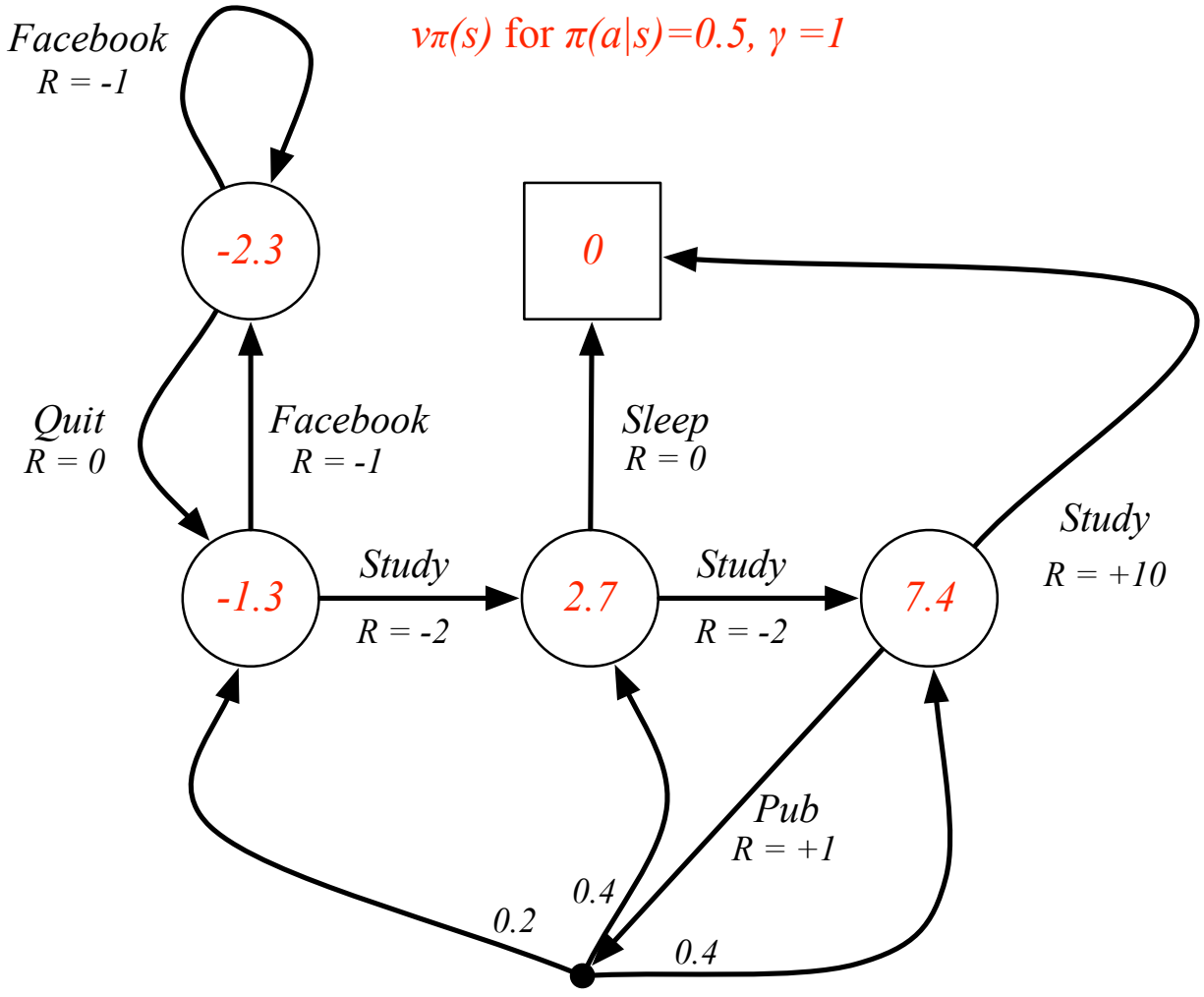
$$V_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}_{ss'}^a V_\pi(s') \right)$$

贝尔曼期望方程 Bellman Expectation Eq, Q^{π}

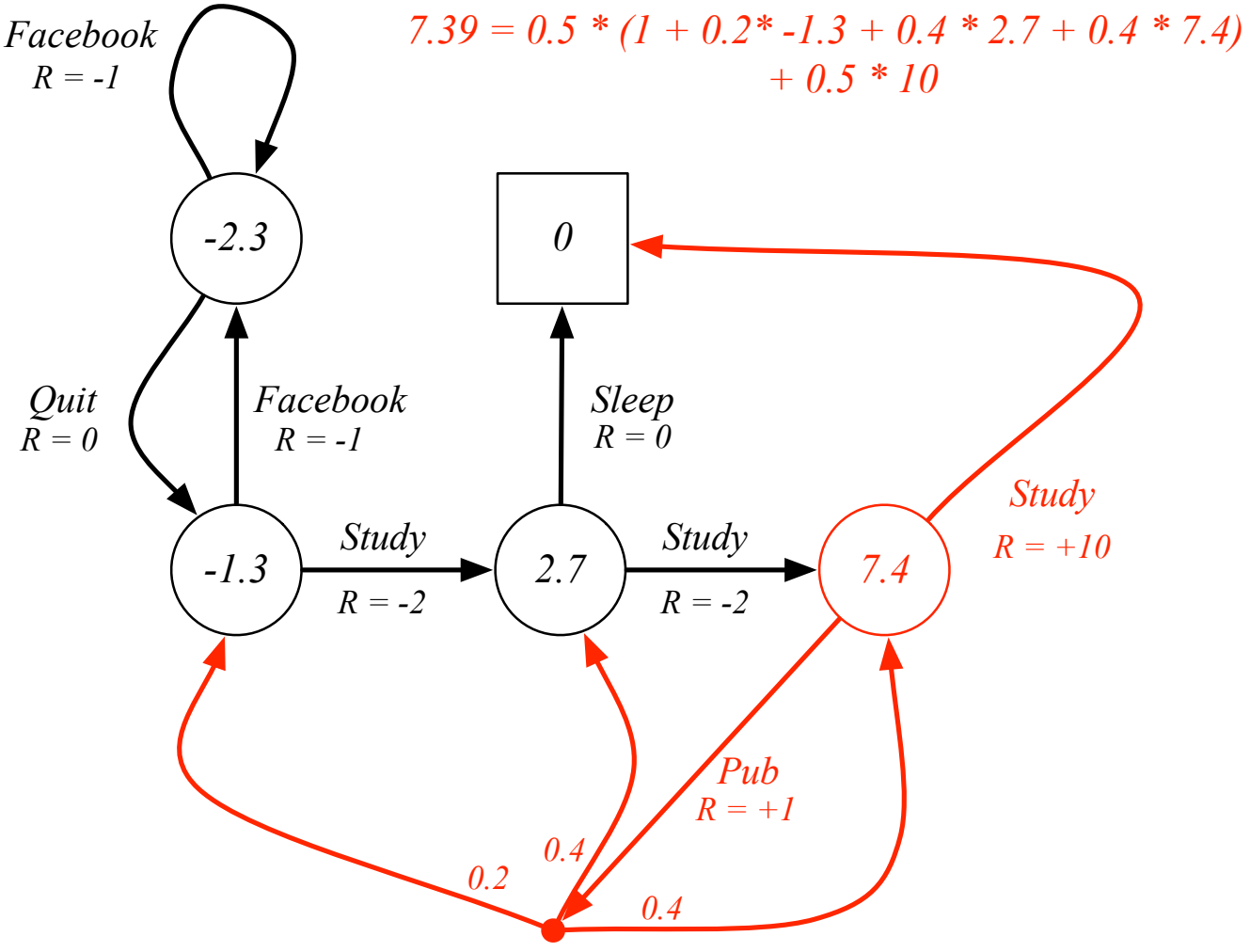


$$Q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') Q_\pi(s',a')$$

举例: 学生 MDP 的价值


Facebook
R = -1
vπ(s) for π(a|s)=0.5, γ =1
-2.3
0
Quit
R = 0
Facebook
R = -1
Sleep
R = 0
Study
R = +10
-1.3
Study
R = -2
2.7
Study
R = -2
7.4
Pub
R = +1
0.4
0.2
0.4

举例: 学生 MDP 贝尔曼期望方程



贝尔曼期望方程 (矩阵形式)

- 贝尔曼期望方程可以在对应的马尔可夫奖励过程下表示为矩阵形式

$$\mathcal{V}_\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi(\mathcal{V}_\pi)$$

- 方程的解

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

最优价值 Optimal value

定义

在所有策略中价值最大的称为<u>最优 (状态) 价值</u>

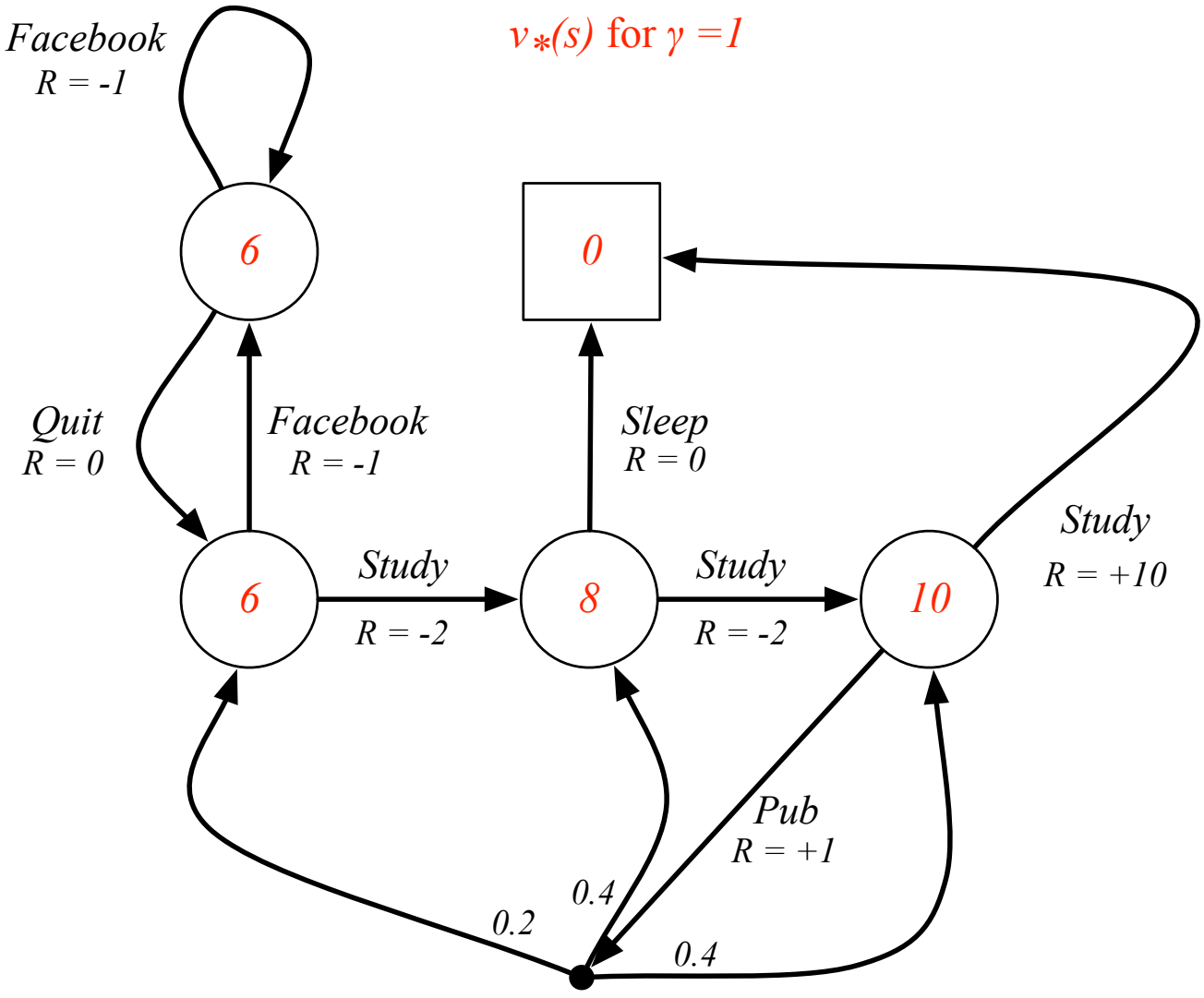
$$V_*(s) = \max_\pi V_\pi(s), \forall s \in \mathcal{S}$$

在所有策略中动作价值最大的称为<u>最优动作 - 价值</u>

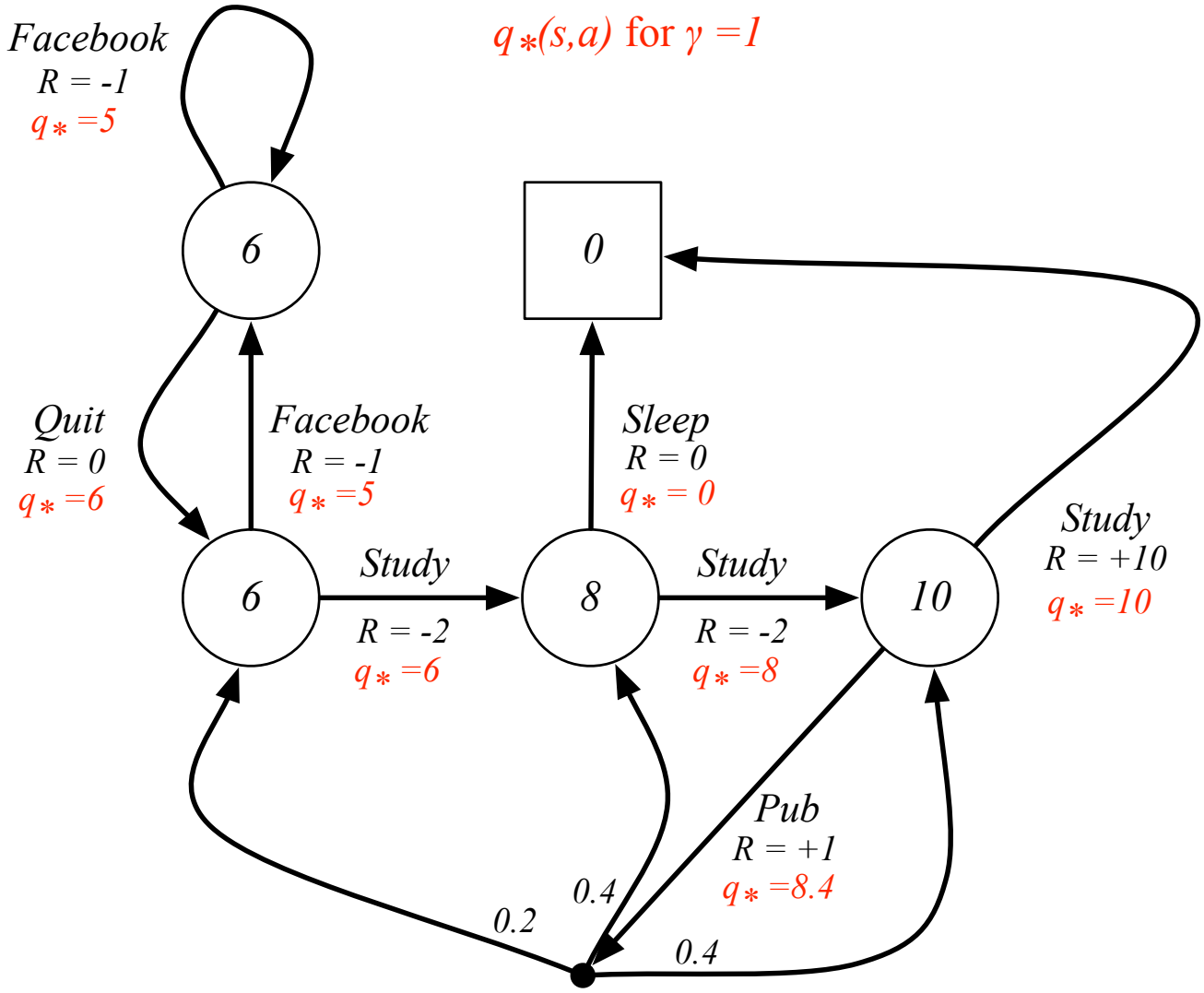
$$Q_*(s,a) = \max_\pi Q_\pi(s,a)$$

- 最优价值代表了智能体能获得的最大期望累加奖励/回报

举例: 学生 MDP 的最优价值



举例: 学生 MDP 的最优动作 -价值



最优策略

- 定义一个关于策略的比较操作

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

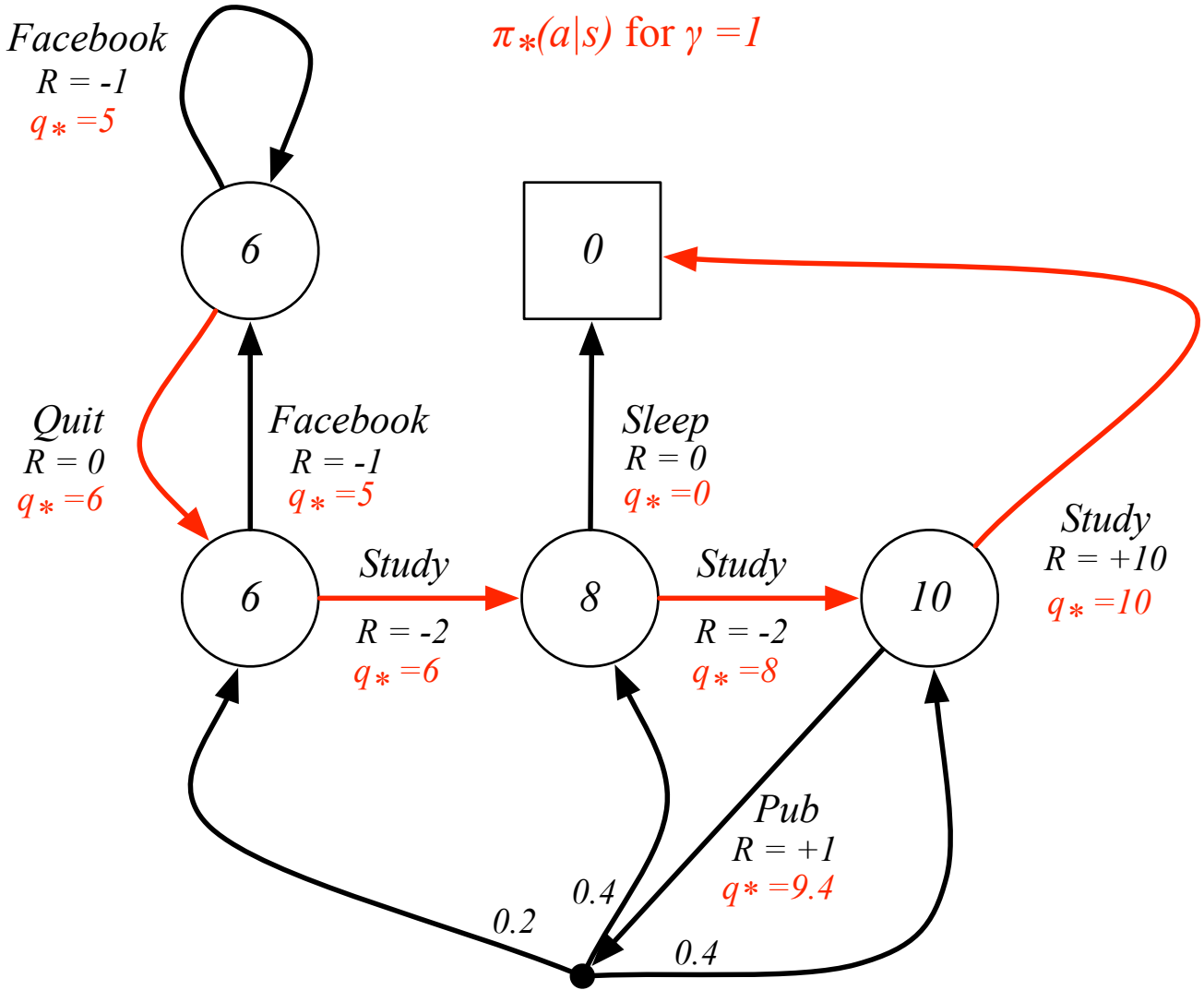
定理

对任意马尔可夫决策过程

- 总是存在一个最优策略 π_* 优于或至少等于其它所有策略，$\pi_* \geq \pi, \forall \pi$
- 所有的最优策略 的价值都是相同的，并且等于最优价值，$V_{\pi_*}(s) = V_*(s)$
- 所有的最优策略 的动作 -价值都是相同的，并且等于最优动作 -价值，$Q_{\pi_*}(s,a) = Q_*(s,a)$

- 智能体的强化学习目标就是要找到最优策略

举例: 学生 MDP 的最优策略



最优化原理

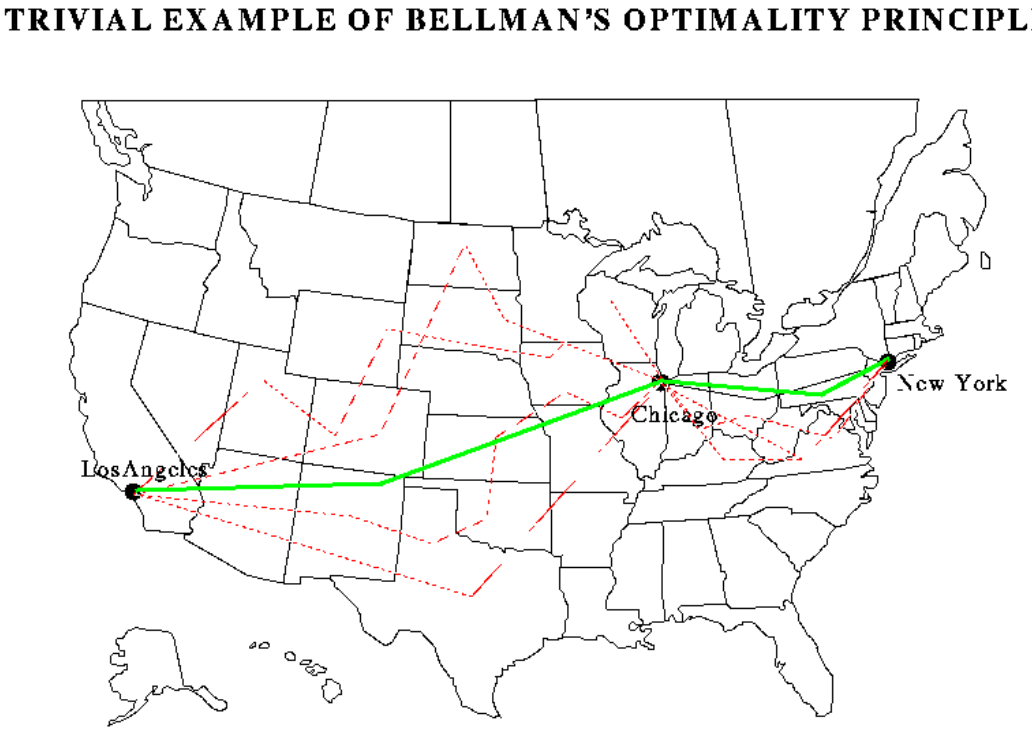
贝尔曼最优化原理

Principle of Optimality (See Bellman, 1957, Chap. III.3)

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

- 最优策略具有如下性质：不论初始状态和初始决策如何，余下的决策依然是余下问题的最优策略

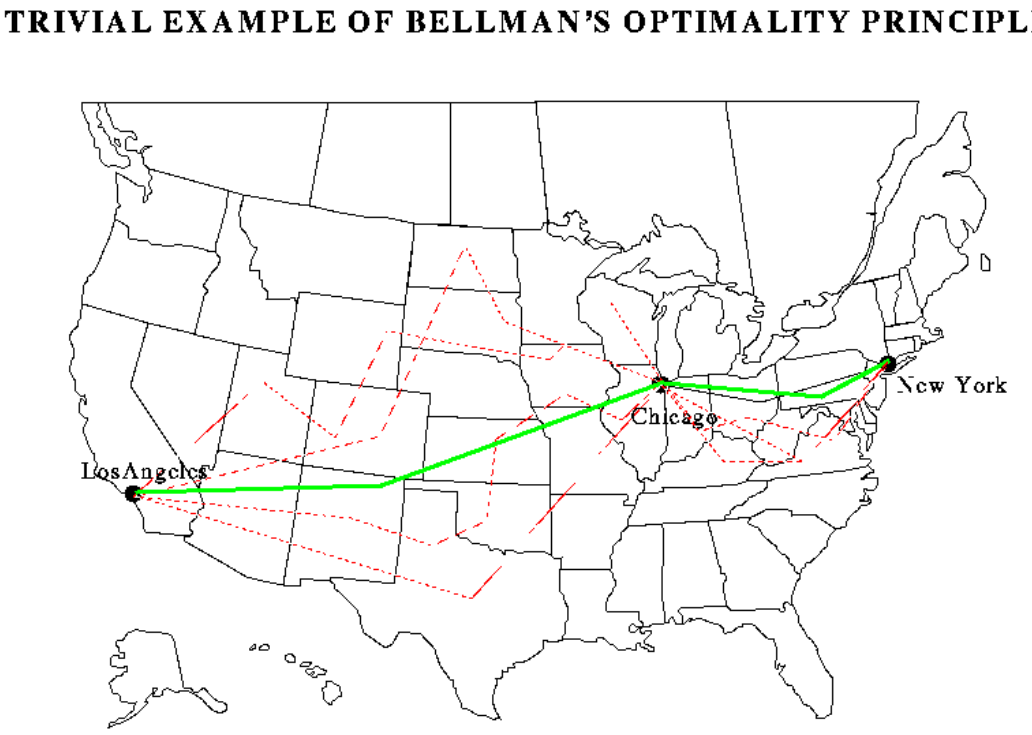
举例



- 从 LA 到 NY 最快的火车路线

  LA — city 1 — $\cdots$ — Chi — city k $\cdots$ — city m — NY

举例



- 从 LA 到 NY 最快的火车路线

  $$\text{LA} \text{ — city 1 — } \cdots \text{ — Chi — city k } \cdots \text{ — city m — NY}$$

- 假如现在已经到达了 Chi，那么剩下的路线

  $$\text{LA — city 1 — } \cdots \text{ — Chi — city k } \cdots \text{ — city m — NY}$$

  依然是从 Chi 到 NY 最快的火车路线

- 最优价值 V_*

$$V_*(s_0) = \mathbb{E}\left[\max_{a_0, a_1, \ldots} \left(r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots\right)\right]$$

- 最优价值 V_*

$$
\begin{aligned}
V_*(s_0) &= \mathbb{E}\left[\max_{a_0, a_1, \ldots} \left(r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots\right)\right] \\
&= \mathbb{E}\left[\max_{a_0} \left(r_1 + \gamma \max_{a_1, a_2, \ldots} \left(r_2 + \gamma r_3 + \ldots\right)\right)\right]
\end{aligned}
$$

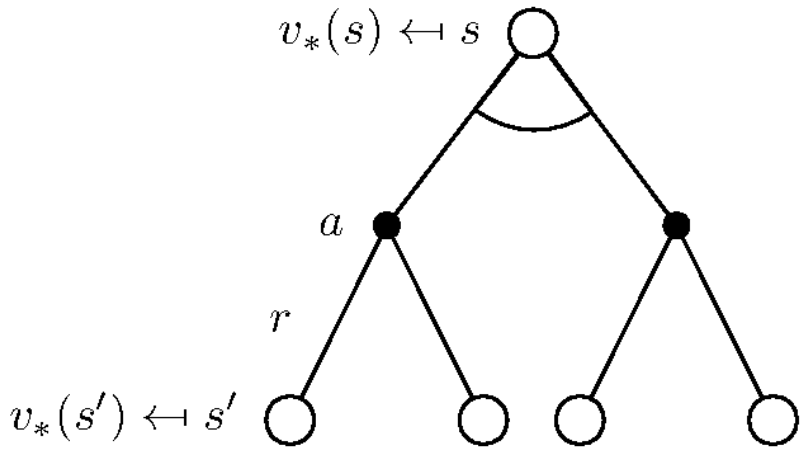
- 最优价值 V_*

$$
\begin{aligned}
V_*(s_0) &= \mathbb{E}\left[\max_{a_0, a_1, \ldots} \left(r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots\right)\right] \\
&= \mathbb{E}\left[\max_{a_0}\left(r_1 + \gamma \max_{a_1, a_2, \ldots} \left(r_2 + \gamma r_3 + \ldots\right)\right)\right] \\
&= \mathbb{E}\left[\max_{a_0}\left(r_1 + \gamma \sum_{s_1} \mathcal{P}^{a_0}_{s_0 s_1} V_*(s_1)\right)\right]
\end{aligned}
$$

- 最优价值 V_*

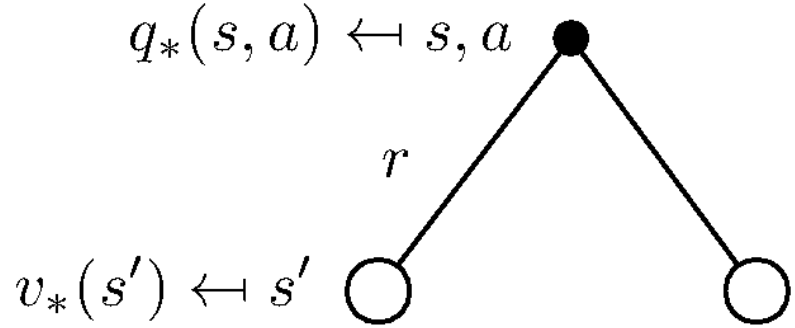
$$
\begin{aligned}
V_*(s_0) &= \mathbb{E}\left[\max_{a_0,a_1,\ldots}\left(r_1+\gamma r_2+\gamma^2 r_3+\ldots\right)\right] \\
&= \mathbb{E}\left[\max_{a_0}\left(r_1+\gamma\max_{a_1,a_2,\ldots}\left(r_2+\gamma r_3+\ldots\right)\right)\right] \\
&= \mathbb{E}\left[\max_{a_0}\left(r_1+\gamma\sum_{s_1}\mathcal{P}^{a_0}_{s_0 s_1}V_*(s_1)\right)\right] \\
&= \max_{a_0}\left(\mathcal{R}(s_0,a_0)+\gamma\sum_{s_1}\mathcal{P}^{a_0}_{s_0 s_1}V_*(s_1)\right)
\end{aligned}
$$

贝尔曼最优方程 Bellman optimal eq V_*

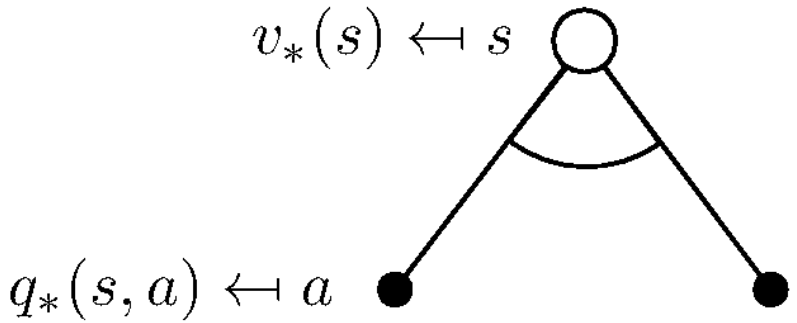


$$V_*(s) = \max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s') \right)$$

- 最优动作 -价值 Q_*

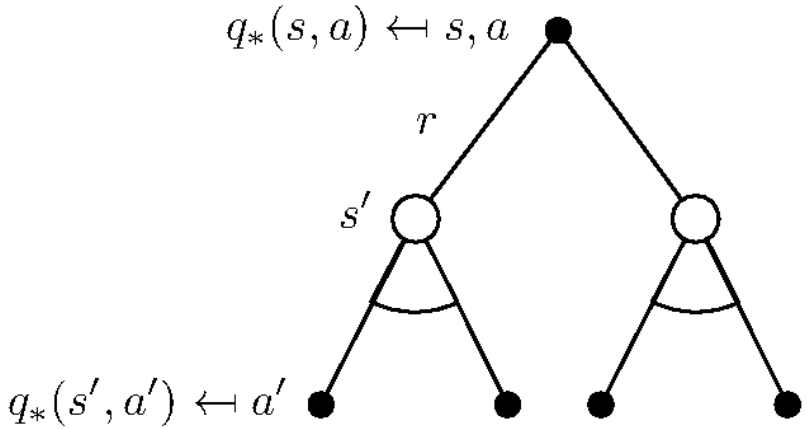


$$Q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s')$$



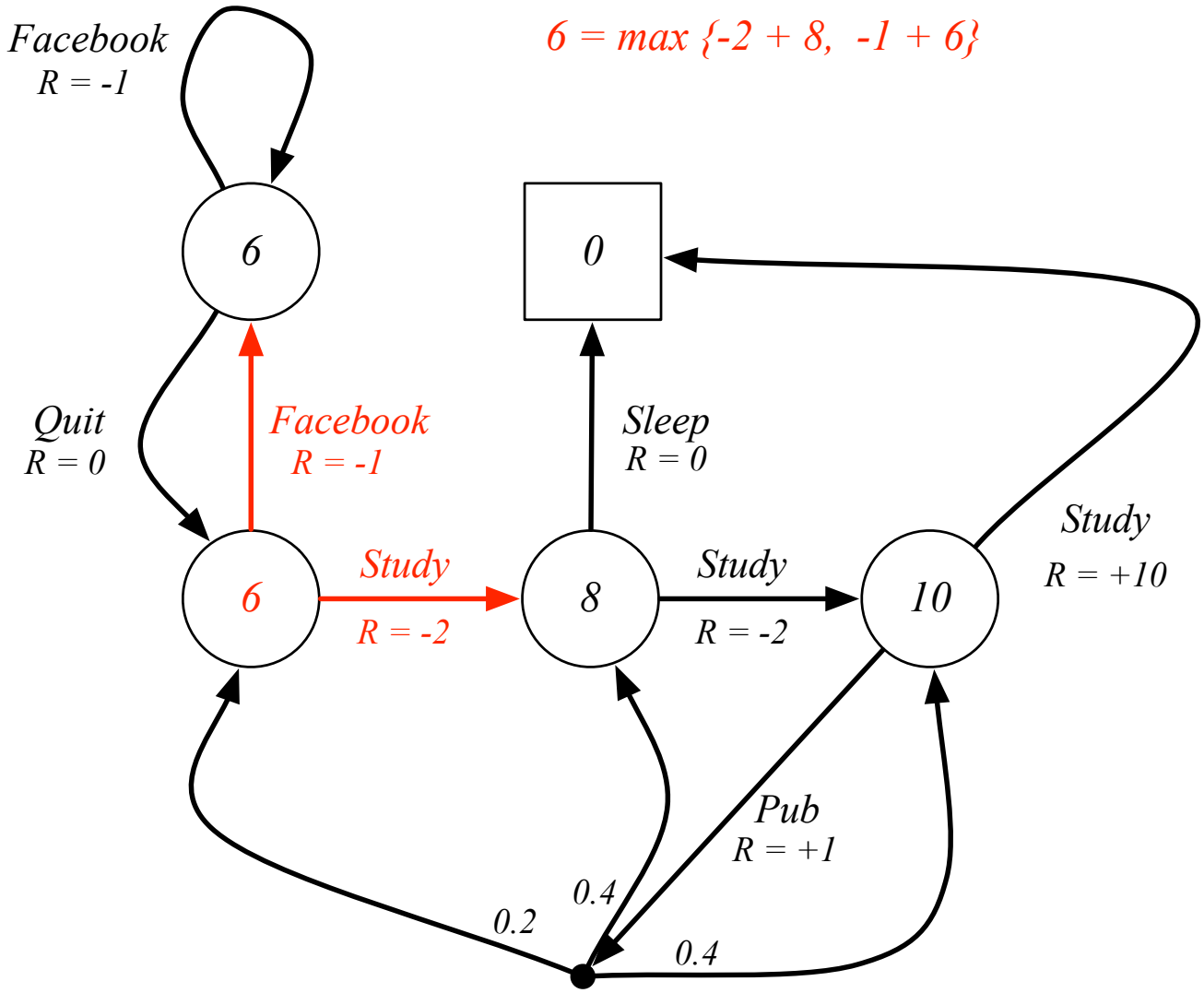
$$\boxed{V_*(s) = \max_a Q_*(s,a)}$$

贝尔曼最优方程 Q_*



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

举例: 学生 MDP 的贝尔曼最优方程



寻找最优策略

$$
\begin{aligned}
V_*(s) = & \max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s') \right) \\
\geq & \mathcal{R}_s^b + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^b V_*(s'), \forall b \in \mathcal{A}
\end{aligned}
$$

寻找最优策略

$$
\begin{aligned}
V_*(s) = &\max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s') \right) \\
\geq &\mathcal{R}_s^b + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^b V_*(s'), \forall b \in \mathcal{A}
\end{aligned}
$$

- 最优策略可以基于 V_* 和模型计算最优动作

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

寻找最优策略

$$
\begin{aligned}
V_*(s) = &\max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a V_*(s') \right) \\
\geq &\mathcal{R}_s^b + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^b V_*(s'), \forall b \in \mathcal{A}
\end{aligned}
$$

- 最优策略可以基于 V_* 和模型计算最优动作

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

- 也可以直接最大化 $Q_*(s, a)$ 得到最优策略

$$
\pi_*(s) = \arg\max_a Q_*(s, a)
$$

确定性最优策略

- 对某一 s，假如存在两个 a_1，a_2 同时最大化 Q_*

$$Q_*(s, a_1) = Q_*(s, a_2) \geq Q_*(s, a), \forall a \in \mathcal{A} \setminus \{a_1, a_2\}$$

确定性最优策略

- 对某一 s，假如存在两个 a_1, a_2 同时最大化 Q_*

$$Q_*(s, a_1) = Q_*(s, a_2) \geq Q_*(s, a), \forall a \in \mathcal{A} \setminus \{a_1, a_2\}$$

- 对任意 $p \in [0, 1]$ 构造策略

$$\pi_p(s) = \begin{cases} a_1 & \text{with prob } p \\ a_2 & \text{with prob } 1 - p \end{cases}$$

π_p 都是最优策略

$$\sum_a \pi_p(a|s) Q_*(s, a) = pQ_*(s, a_1) + (1 - p) Q_*(s, a_2) = V_*(s)$$

确定性最优策略

- 对某一 s，假如存在两个 a_1, a_2 同时最大化 Q_*

$$Q_*(s, a_1) = Q_*(s, a_2) \geq Q_*(s, a), \forall a \in \mathcal{A} \setminus \{a_1, a_2\}$$

- 对任意 $p \in [0, 1]$ 构造策略

$$\pi_p(s) = \begin{cases} a_1 & \text{with prob } p \\ a_2 & \text{with prob } 1 - p \end{cases}$$

π_p 都是最优策略

$$\sum_a \pi_p(a|s) Q_*(s, a) = pQ_*(s, a_1) + (1 - p) Q_*(s, a_2) = V_*(s)$$

- 回顾：最优策略不是唯一的，但是最优价值是唯一的

确定性最优策略

- 对某一 s，假如存在两个 a_1, a_2 同时最大化 Q_*

$$Q_*(s, a_1) = Q_*(s, a_2) \geq Q_*(s, a), \forall a \in \mathcal{A} \setminus \{a_1, a_2\}$$

- 对任意 $p \in [0, 1]$ 构造策略

$$\pi_p(s) = \begin{cases} a_1 & \text{with prob } p \\ a_2 & \text{with prob } 1 - p \end{cases}$$

π_p 都是最优策略

$$\sum_a \pi_p(a|s) Q_*(s, a) = pQ_*(s, a_1) + (1 - p) Q_*(s, a_2) = V_*(s)$$

  - 回顾：最优策略不是唯一的，但是最优价值是唯一的

- 对智能体来说选择任意形式的 π_p 都能最大化期望累加奖励
- 为了方便，通常选择一个确定性的最优策略

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_a Q_*(s, a) \\ 0 & \text{otherwise} \end{cases}$$

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

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

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

$$\pi_*(s) = \arg\max_a Q_*(s,a)$$

- 求解贝尔曼最优方程需要：
  1. 求解非线性算子 max
  2. 模型已知
  3. 足够的计算空间

MDPs 扩展

MDPs 扩展

回顾: 马尔可夫决策过程用 5 个元素表示 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$: 有限状态空间
  - 离散集合, $\{S_1, S_2, \ldots, S_N\}$
- $\mathcal{A}$: 有限动作空间
  - 离散集合, $\{A_1, A_2, \ldots, A_M\}$
- $\mathcal{P}$: 状态转移函数
  - 离散时间, $s_{t+1} \sim \mathcal{P}(s_t, a_t)$
- $\mathcal{R}$: 奖励函数
  - $r_{t+1} \sim \mathcal{R}(s_t, a_t)$
- γ: 衰减因子
  - 无穷时域的累加奖励, $G_t = r_{t+1} + \gamma r_{t+2} + \ldots$

系统模型的不同类型 (1)

连续状态/动作 vs 离散状态/动作

1. 连续状态: $s_t \in \mathbb{R}^n$
   e.g. 车速
2. 连续动作: $a_t \in \mathbb{R}^m$
   e.g. 油门深度
3. 离散状态: $s_t \in \{S_0, S_1, \ldots, S_N\}$
   e.g. 棋盘局面
4. 离散动作: $a_t \in \{A_0, A_1, \ldots, A_M\}$
   e.g. 下棋位置

系统模型的不同类型 (2)

确定模型 vs 随机模型

1. 确定模型: $s_{t+1} = f(s_t, a_t)$
   e.g. $1 + 1 = 2$
2. 随机模型: $s_{t+1} \sim \mathcal{P}(s_t, a_t)$
   e.g. $1 \underbrace{+1}_{\text{command via wireless network}} =?$

系统模型的不同类型 (3)

连续时间 vs 离散时间

1. 连续时间模型: $\mathrm{d}s/\mathrm{d}t = f(s(t), a(t)) + \varepsilon$, 噪声 $\varepsilon \sim \mathcal{N}(0, \Sigma)$:
   e.g. 现实世界运动规律
   - 确定性: $\mathrm{d}s/\mathrm{d}t = f(s(t), a(t))$
   - 随机性: $\mathrm{d}s/\mathrm{d}t = f(s(t), a(t)) + \varepsilon$, 噪声 $\varepsilon \sim \mathcal{N}(0, \Sigma)$
2. 离散时间模型: e.g. 数字计算机
   - 确定性: $s_{t+1} = f(s_t, a_t)$
   - 随机性: $s_{t+1} \sim \mathcal{P}(s_t, a_t)$

奖励 vs 惩罚

- 奖励函数反映了智能体在某一状态时的好坏
  - 正的奖励, 智能体目标是最大化期望回报

$$\max \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \dots]$$

  - 负的奖励 (惩罚), 智能体目标是最小化期望回报

$$\min \mathbb{E}[c_{t+1} + \gamma c_{t+2} + \dots]$$

  - 通过正负号可以将两种情况相互转换

$$\boxed{\mathcal{C}(s) = (-1) * \mathcal{R}(s)}$$

回报的不同形式 (1)

- 无限时域的累加奖励

$$G(s_0) = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$$

  - γ 折扣因子，$\gamma \in [0, 1]$，反映对未来奖励的重视程度，γ 越小，未来奖励对当前回报的影响越小
  - 终止状态: 智能体状态不再变化, 轨迹结束

$$s_0, s_1, \ldots, s_T = \text{terminal state}$$

$$G_0 = r_1 + \gamma r_2 + \cdots + \gamma^T r_{T+1}$$

  - 当 $\gamma = 1$ 时，为了保证 $G(s_0)$ 的有效性，智能体要么达到终止状态, 要么在无穷时刻奖励为 0

回报的不同形式 (2)

- 无限时域的平均奖励

$$G(s_0) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t$$

  - 智能体在 MDP 的状态空间是各态遍历的
  - 不存在终止状态, MDP 的状态分布 $d^{\pi}(s)$ 达到平衡

$$d^{\pi}(s) = \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} d^{\pi}(s')$$

  - 各态遍历的 MDP 问题, 当 $T \to \infty$ 时平均奖励与状态无关

$$G^{\pi} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} r_t$$

回报的不同形式 (3)

- 有限时域的累加奖励和

$$G(s_0) = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$$

  - $T < \infty$
  - 只考虑固定时长的累加奖励和，$r_1, r_2, \ldots, r_T$
  - 比如股票未来 7 天的收益

- 有限时域的平均奖励

$$G(s_0) = \frac{1}{T} \sum_{t=1}^{T} r_t$$

总结

马尔可夫性

马尔可夫过程

马尔可夫奖励过程

马尔可夫决策过程

策略与价值

最优化原理

MDPs 扩展