强化学习（Reinforcement Learning）

作者：Calvin
QQ: 179209347
Mail: 179209347@qq.com

介绍

笔记简介：

- 面向对象：深度学习初学者
- 依赖课程：**线性代数，统计概率**，优化理论，图论，离散数学，微积分，信息论

知乎专栏：

https://zhuanlan.zhihu.com/p/693738275

Github & Gitee 地址：

https://github.com/mymagicpower/AIAS/tree/main/deep_learning

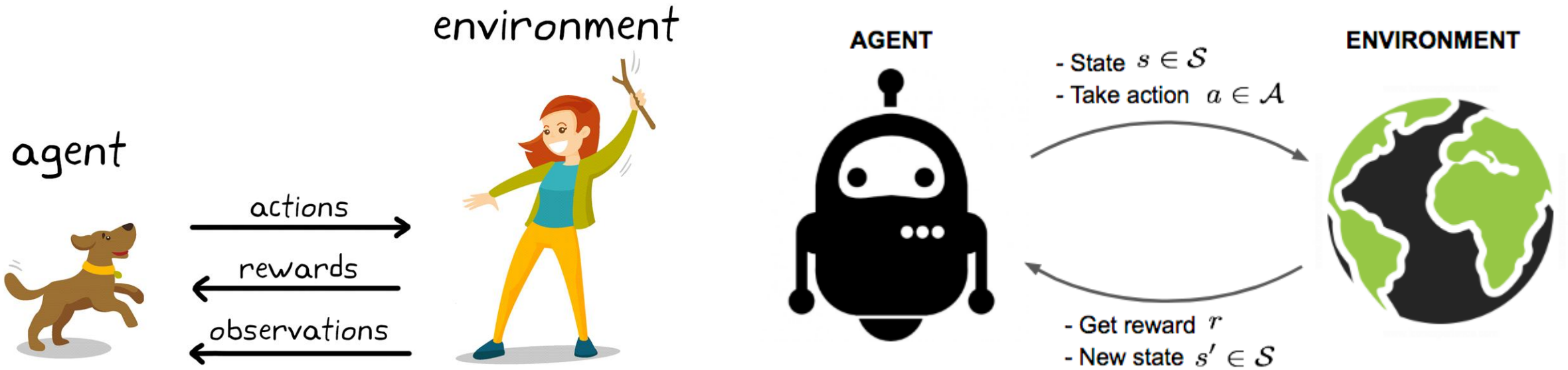
https://gitee.com/mymagicpower/AIAS/tree/main/deep_learning

*** 版权声明：**

- 仅限用于个人学习
- 禁止用于任何商业用途

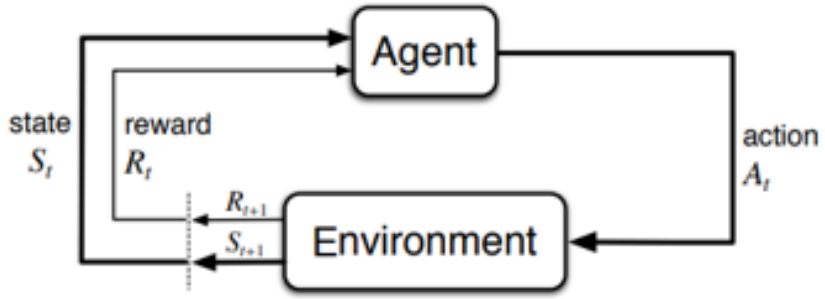
强化学习（Reinforcement Learning）

强化学习（Reinforcement Learning）是一种机器学习方法，旨在让智能体通过与环境的交互学习如何做出决策以达到既定的目标。在强化学习中，智能体通过尝试不同的行动来最大化累积的奖励，而不是依赖标记的数据进行学习。



马尔可夫决策过程（Markov Decision Process，MDP）

马尔可夫决策过程（MDP）是一个具有马尔可夫状态的环境；马尔可夫状态满足马尔可夫性质：该状态包含过去预测未来的所有相关信息。 MDP通常用于强化学习领域，其中智能体需要在不确定性环境中做出决策以最大化长期累积奖励。



名称	符号	说明
状态 状态空间	$\mathcal{S} = \{s_1, s_2, \cdots, s_\tau\}$	状态是对环境的描述，在智能体做出动作后，状态会发生变化，且演变具有马尔可夫性质。MDP所有状态的集合是状态空间。状态空间可以是离散或连续的。
动作 动作空间	$\mathcal{A} = \{a_1, a_2, \cdots, a_\tau\}$	动作是对智能体行为的描述，是智能体决策的结果。MDP所有可能动作的集合是动作空间。动作空间可以是离散或连续的。
策略	$\pi(a\|s) = p(a\|s)$	MDP的策略是按状态给出的，动作的条件概率分布，在强化学习的语境下属于随机性策略。
奖励	$R = R(s_t, a_t, s_{t+1})$	智能体给出动作后环境对智能体的反馈。是当前时刻状态、动作和下个时刻状态的标量函数。
回报	$G = \sum_{t=0}^{\tau-1} R_{t+1}$	回报是奖励随时间步的积累，在引入轨迹的概念后，回报也是轨迹上所有奖励的总和。

马尔可夫性质（Markov property）

马尔可夫性质是指，未来只与当前状态有关，与过去无关。例如，等红绿灯时，假设你每一秒看一次灯，有了当前这一秒看到红绿灯的状态，就不需要前面看到的所有状态了。其数学定义为：

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

对于一个马尔可夫状态 s 及其后续状态 s'，状态转移概率 定义为：

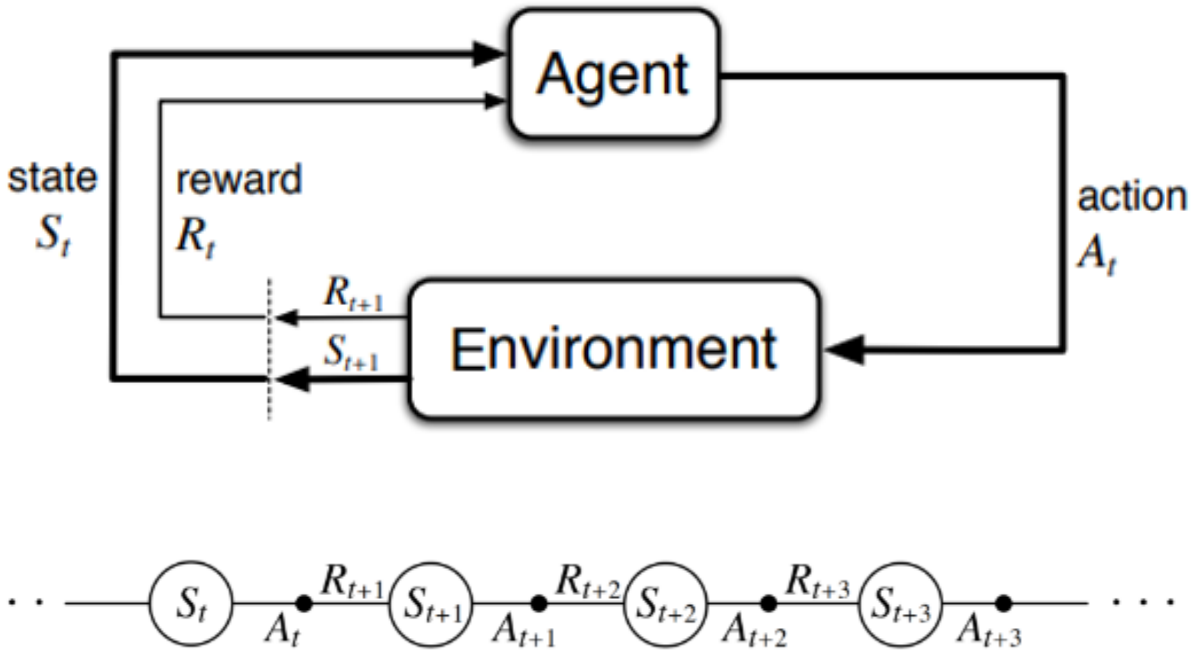
$$P_{ss'} = P[S_{t+1} = s'|S_t = s]$$

状态转移概率矩阵 P 定义为从所有的状态 s 到所有的后续状态 s' 的转移概率：

$$\boldsymbol{P} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & & & \\ \dots & & & \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix}$$

矩阵中行号表示当前状态 s，列号表示到达的后续状态 s'。每一行的和为 1。

马尔可夫决策过程（Markov Decision Process，MDP）



MDP 的正式定义为：

马尔可夫决策过程是一个 5 元组S, A, P, R, γ:

- S 是一个有限状态集
- A 是一个有限动作集
- $P_a(s, s') = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ 是行动的概率，也就是在动作 a 和状态 s 共同作用下的状态转移矩阵。
- R 是从状态转换后收到的立即奖励（或预期立即奖励）

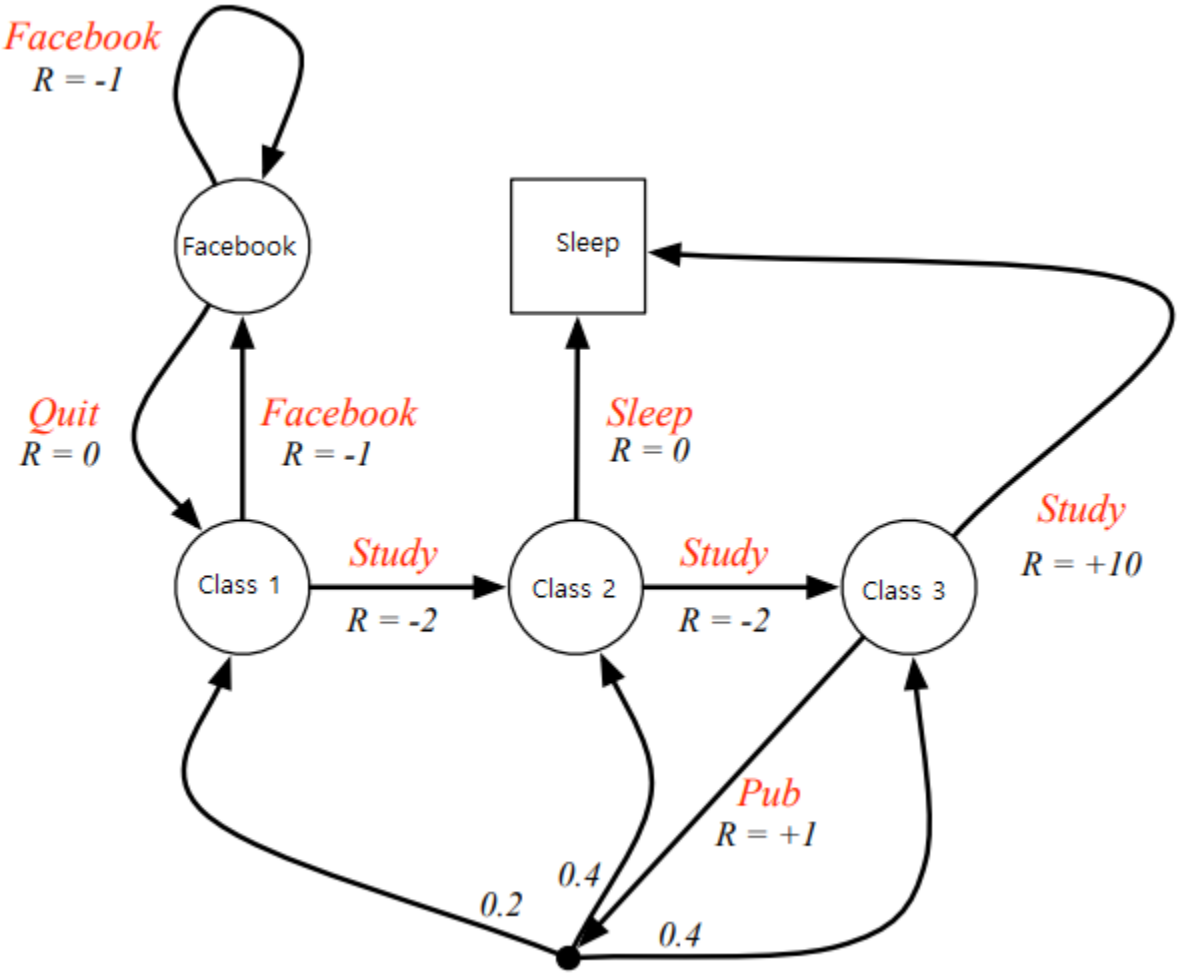
$$R(s_t = s, a_t = a) = E[r_t | s_t = s, a_t = a]$$

- 折扣因子 $\gamma \in [0,1]$

MDP – 例子

图中描绘了一名学生一天在学校的场景：
圆圈和正方形代表可以处于的状态，红色文字是根据所处的状态您可以采取的操作，例如：

在状态 1 中，可以选择是否要学习或查看 Facebook，并根据采取的操作给予数字奖励。
还有一个动作节点（图中的后点），根据转移概率，可以在其中结束不同的状态；例如，当决定从 Class 3 去 Pub 后，有 0.2 的概率进入 Class 1 。这个节点显示了你无法控制的环境的随机性。在所有其他情况下，转移概率为 1，



如果折扣因子为 1，则 MDP 可以定义为：

(S, A, R, P, γ)

S: <Facebook, Class 1, Class 2, Class 3, Sleep>

A: <Study, Facebook, Sleep, Pub>

R: <-2, -1, 0, +1, +10>

P: <0.2,0.4,1>

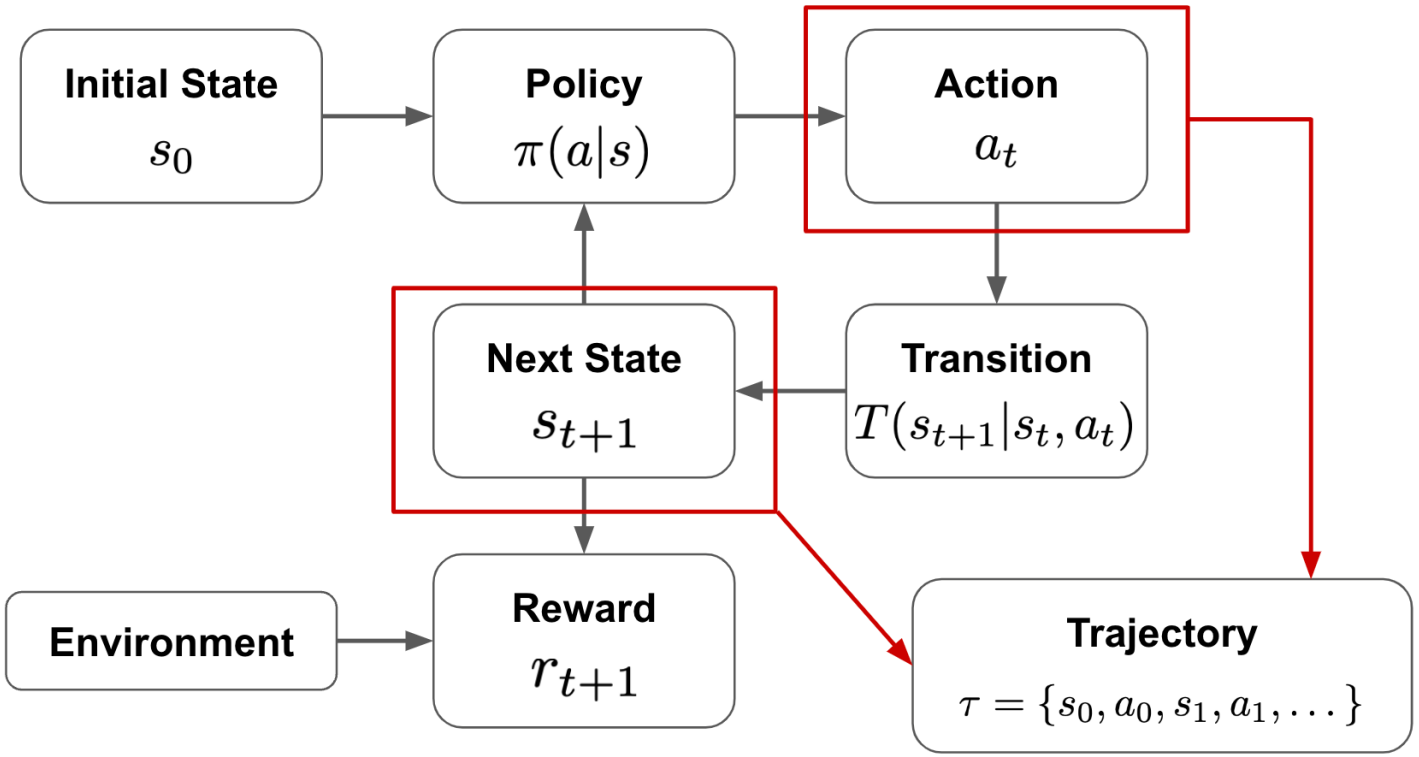
γ: 1

MDP – 策略

策略告诉您要采取哪些行动。它定义为：

$$\pi(a|s) = P(a_t| = a \mid s_t = s)$$

策略 π 本质上是在某一状态 s 时，指定一个 action 的概率分布，即在某一状态 s_t 下采取可能的行为 $action$ 的概率。
对于 MDP 来说，策略仅取决于当前状态。



MDP – 轨迹

在此基础上，类比马尔可夫链中的样本轨道（sample path），可定义MDP的轨迹（trajectory）

$$A_\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_{\tau-1}, a_{\tau-1}, r_{\tau-1}, s_\tau r_\tau\}$$

即环境由初始状态 s_0 按给定策略 $\pi(a|s)$ 演进至当前状态 s_t 的所有动作、状态和奖励的集合。由于MDP的策略和状态转移具有随机性，因此其模拟得到的轨迹是随机的，且该轨迹出现的概率有如下表示:

$$p(A_\tau) = p(s_0)\prod_{i=0}^{\tau-1} p(a_i \mid s_i)p(s_{i+1} \mid s_i, a_i)$$

一般地，MDP中两个状态间的轨迹可以有多条，此时由Chapman-Kolmogorov等式可知，两个状态间的n步转移概率是所有轨迹出现概率的和。

MDP – 总回报

给定策略 $\pi(a|s)$，智能体和环境一次交互过程的轨迹 τ 所收到的累积奖励为总回报（return）：

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

γ ∈ [0,1] 是折扣率。当 γ 接近于 0 时，智能体更在意短期回报；而当 γ 接近于 1 时，长期回报变得更重要。
环境中有一个或多个特殊的终止状态（terminal state）

基于值函数的策略学习 - 如何评估策略 $\pi(a|s)$？

两个值函数：

- 状态值函数:状态价值函数告诉您您所处的状态“有多好”
- 状态-动作值函数：告诉您在特定状态下采取特定行动“有多好”

状态（或状态-动作对）的“多好”是根据预期的未来奖励来定义的。

一个策略π期望回报可以分解为：

$$\mathbb{E}_{\tau\sim p(\tau)}[G(\tau)] = \mathbb{E}_{s\sim p(s_0)}\left[\mathbb{E}_{\tau\sim p(\tau)}\left[\sum_{t=0}^{T-1} \middle| \gamma^t r_{t+1}\tau_{s_0} = s\right]\right] = \mathbb{E}_{s\sim p(s_0)}[V^{\pi}(s)]$$

状态值函数定义为：

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

$$V^{\pi}(s) = \mathbb{E}_{\tau\sim p(\tau)}\left[\sum_{t=0}^{T-1} \middle| \gamma^t r_{t+1}\tau_{s_0} = s\right]$$

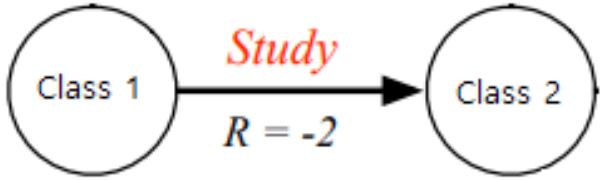
从状态s开始，执行策略π得到的期望总回报

状态-动作值函数定义为：

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

贝尔曼方程 （Bellman）

上面定义的价值函数满足**贝尔曼方程**；它指出： “开始状态的值必须等于预期下一个状态的值，加上一路上预期的奖励。”



例如，如果我们采用从 1 类到 2 类的路径，那么我们可以按以下方式编写贝尔曼方程：

$$V_\pi(\text{Class1}) = \text{R} + \text{y}V_\pi(\text{Class2})$$

价值函数的贝尔曼方程

贝尔曼最优方程可以用类似的方式写成：

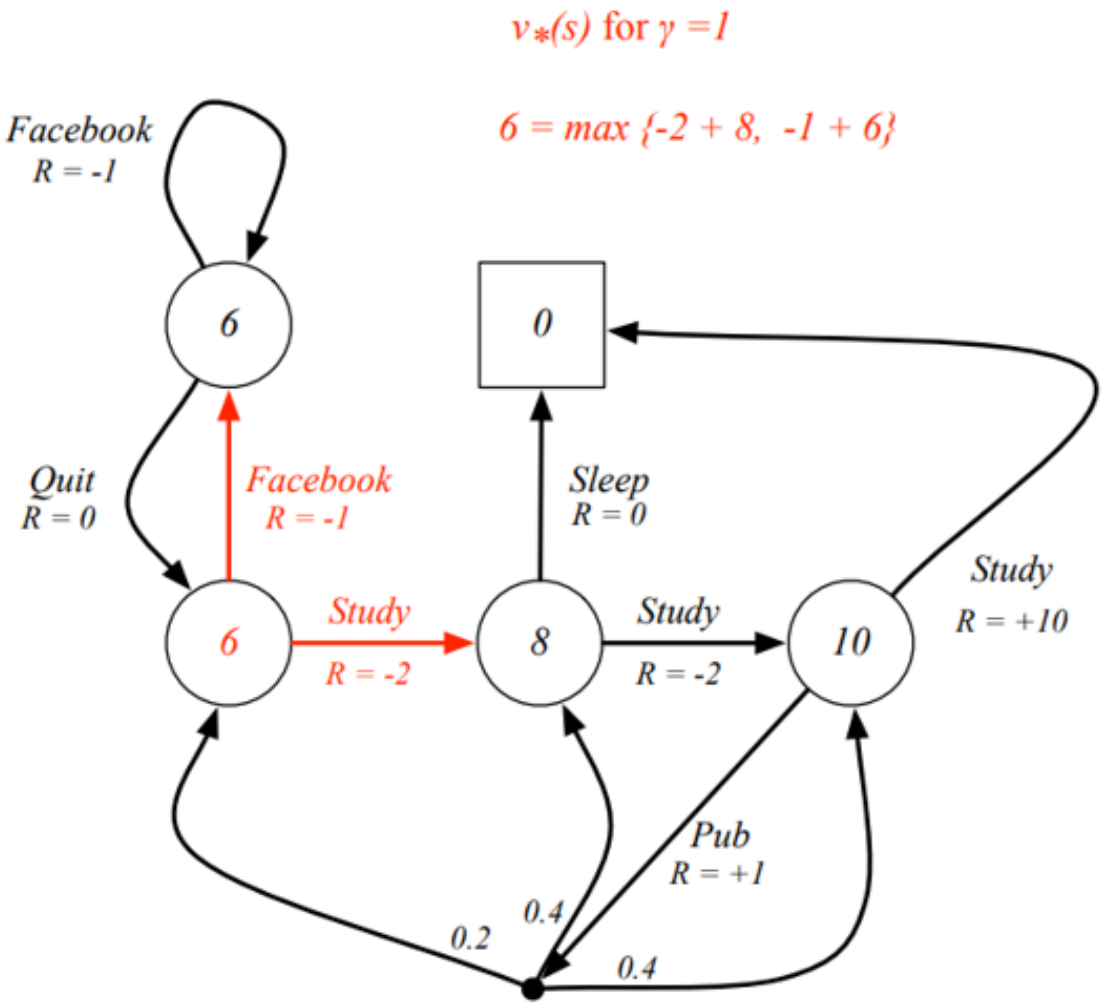
$$V_*(\text{Class1}) = \text{R} + \text{y}V_*(\text{Class2})$$

这些概念可以轻松扩展到多条路径，并对不同状态采取不同的操作。在这种情况下，贝尔曼最优性方程为：

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

MDP – 最优值函数

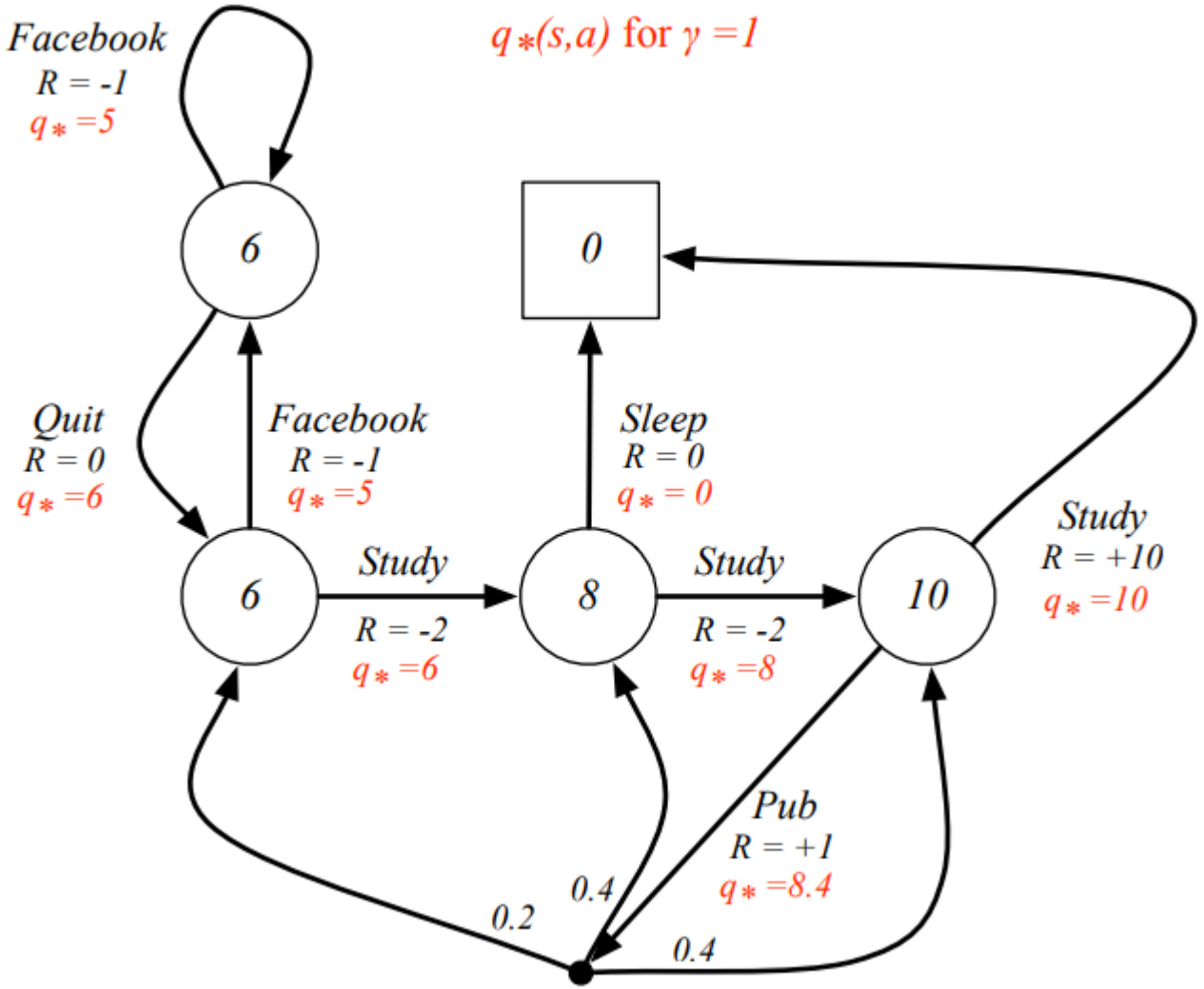
使用上面的方程，我们可以找到学生 MDP 示例中每个状态的最优值函数。



MDP – 最佳动作值

最佳动作值可以用类似的方式表示：

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

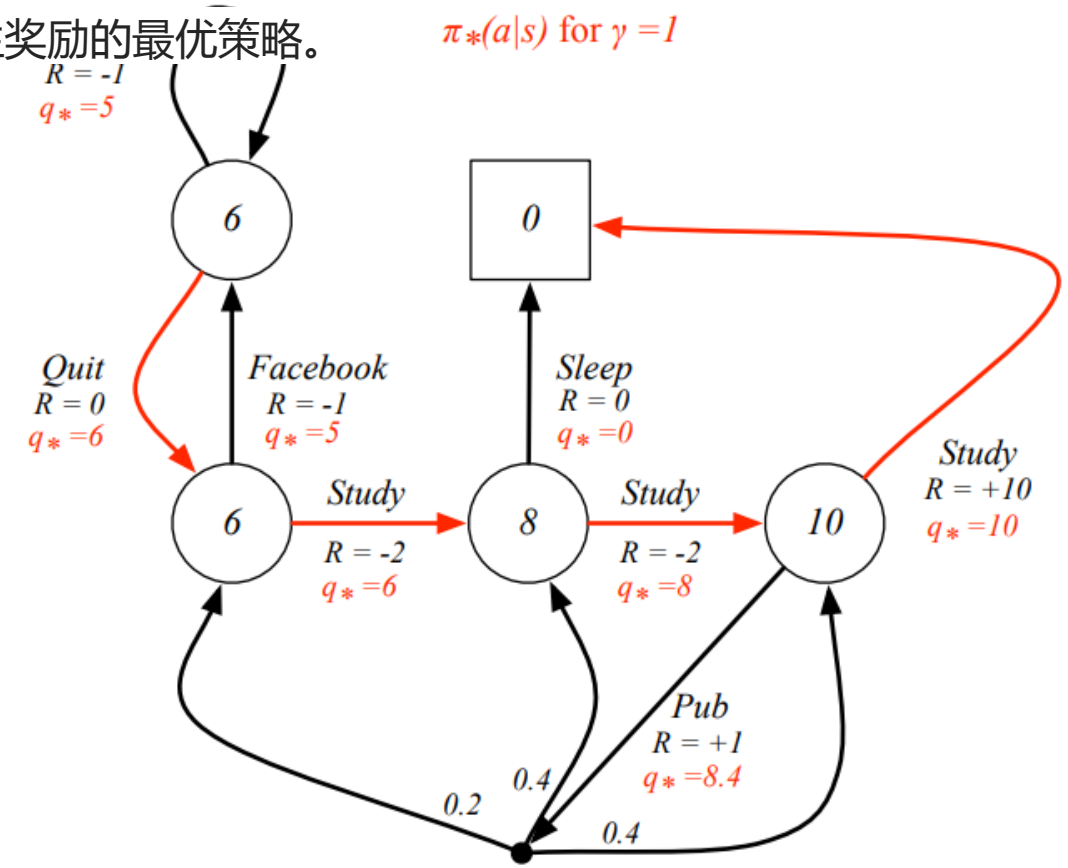


MDP – 最优策略

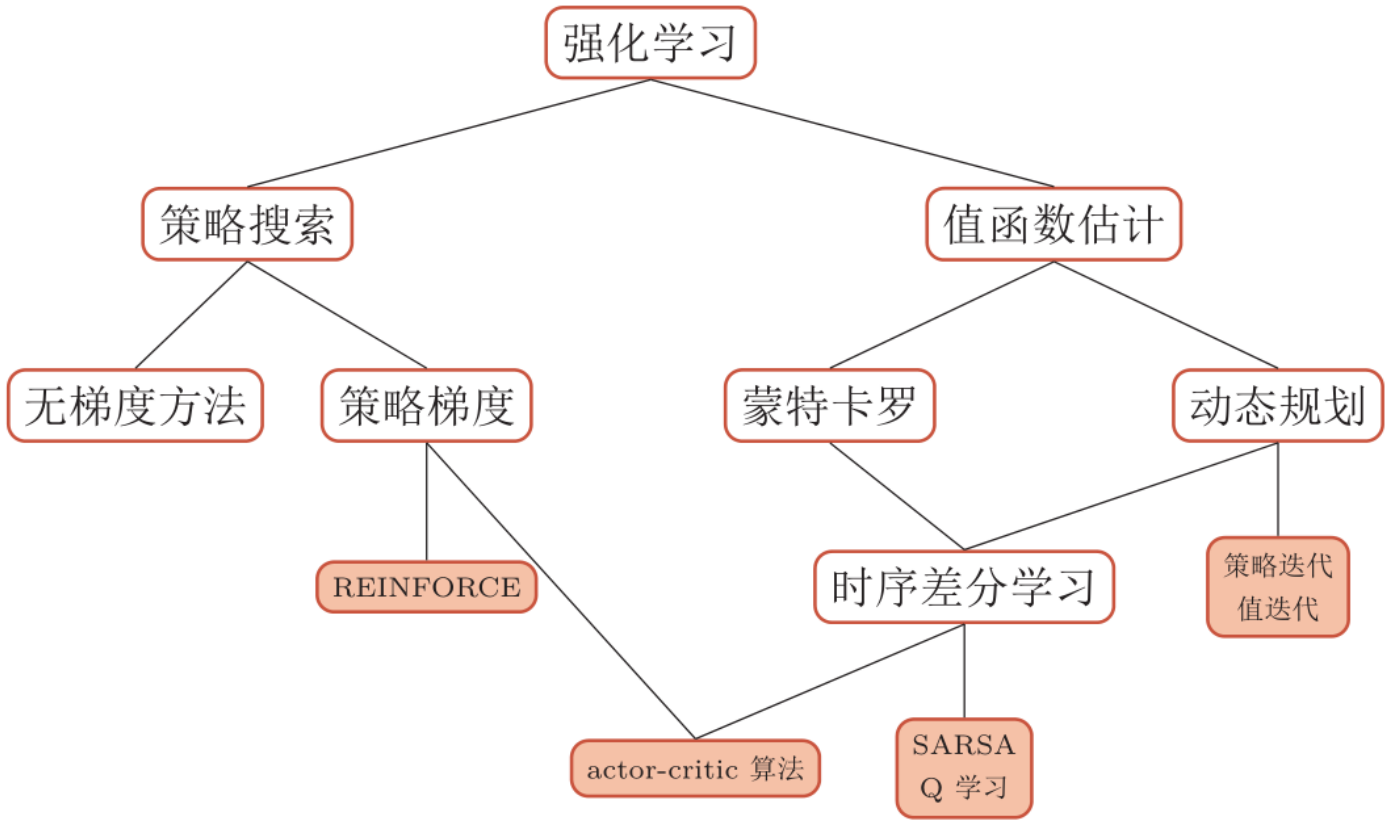
一旦我们有了行动价值函数，我们就可以通过取最大值来找到最优策略：

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

红色弧线显示了最大化学生奖励的最优策略。



不同强化学习算法之间的关系




Thank
You