人工智能 I

杨杰

yangjie@seu.edu.cn

2025/12/22，五四楼-303，9:50~11:25

本周四实验课安排

周次	14	
日期	2025/12/25	
星期	星期四	
时间	16:40-18:15	19:00-20:35
机房地点	五五楼342 软实五	中心楼实验室 B区、D区
实验任务	强化学习基础算法	

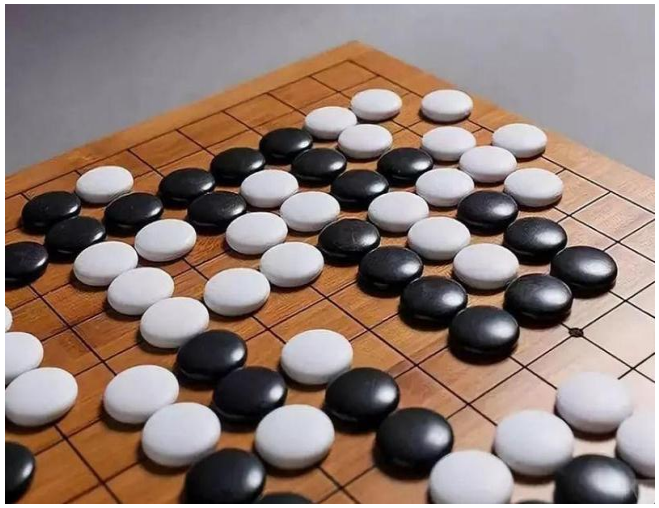
注意两个时间段是不同机房！请勿跑错！
注意18:15机房电脑自动关机！请及时保存！

上节课回顾

- 马尔科夫决策过程 MDP：$< S, A, P, R >$
- 状态离散且有限、动作离散且有限
  - 环境模型P已知
    - ✓ 值迭代、策略迭代
  - 环境模型P未知
    - ✓ 基于模型估计的方法
    - ✓ 免模型方法：TD、Q-Learning、蒙特卡罗法

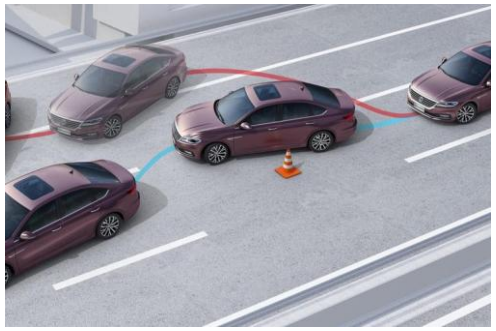
背景

- 前面所学方法针对离散有限状态集：
  - 难以应对状态数很多的情况



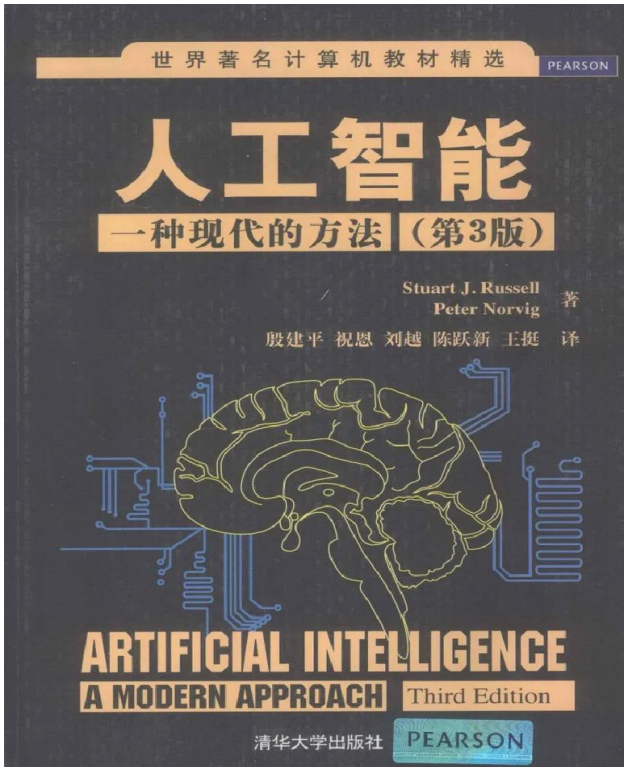
连续状态空间

  - 没有泛化能力，不能处理未见状态



本节课安排

- 连续状态强化学习
  - 值函数近似
  - 策略搜索



状态连续的序列决策问题

- 马尔可夫决策过程（MDP）：

$$< S, A, P, R >$$

  - 状态空间$S \subseteq \mathbb{R}^d$，状态由向量表示：

$$\mathbf{x} \in \mathbb{R}^d$$

  - 动作集A离散且有限
  - 状态转移模型：

$$P(\mathbf{y}|\mathbf{x}, a), \qquad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$$

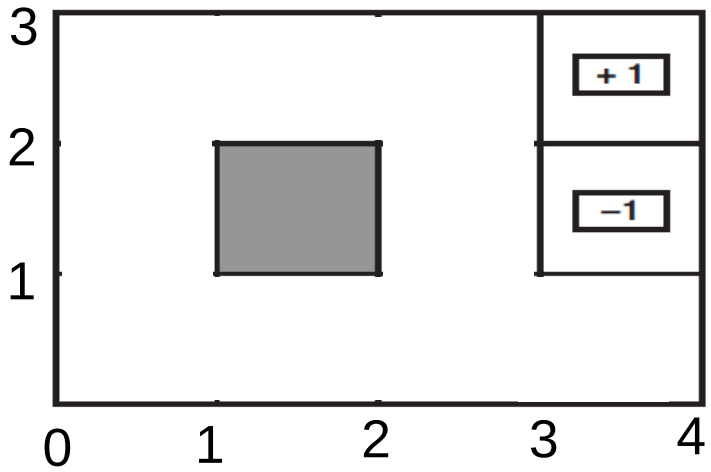
  - 奖赏函数：$R(\mathbf{x})$表示到达状态$\mathbf{x}$时获得的奖励

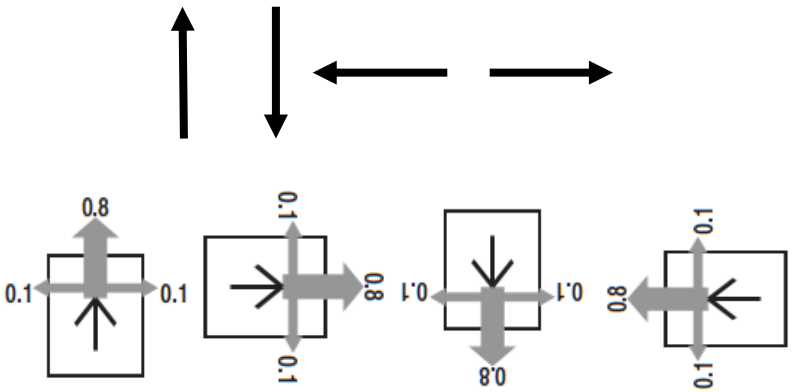
- 问题的解：策略

$$\pi: S \to A$$

  - $\pi(\mathbf{x})$给出状态$\mathbf{x}$下执行的动作，$\forall \mathbf{x} \in S$

一个例子





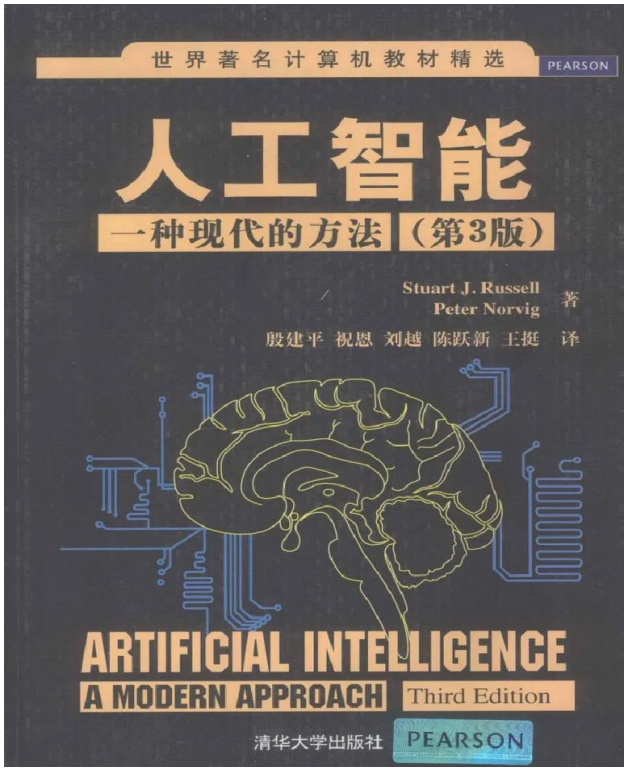
- 状态空间：

$$S = \{(x, y): 0 \le x \le 4, 0 \le y \le 3\} \setminus \{(x, y): 1 \le x \le 2, 1 \le y \le 2\}$$

- 状态转移模型：以状态(x, y)下执行up动作为例
  - 以概率0.8，进入状态$(x, y + 0.1)$
  - 以概率0.1，进入状态$(x - 0.1, y)$　超界则状态不变
  - 以概率0.1，进入状态$(x + 0.1, y)$
- 奖赏函数：除终止状态外，$R((x, y)) = -0.001$

本节课安排

- 连续状态
  - 值函数近似
  - 策略搜索



值函数

- 状态值函数：

S_t是d元随机变量

$$U_T^{\pi}(\mathbf{x}) = \mathbf{E}_{\boldsymbol{S}|S_0=\mathbf{x}}\left[\frac{1}{T}\sum_{t=1}^{T} R(S_t)\right]$$

T步平均奖赏

$$U^{\pi}(\mathbf{x}) = \mathbf{E}_{\boldsymbol{S}|S_0=\mathbf{x}}\left[\sum_{t=0}^{+\infty} \gamma^t R(S_t)\right]$$

累加奖赏

- Q函数（状态-动作值函数）：

$$Q_T^{\pi}(\mathbf{x}, a) = \mathbf{E}_{\boldsymbol{S}|(S_0=\mathbf{x},a_0=a)}\left[\frac{1}{T}\sum_{t=1}^{T} R(S_t)\right], \forall a \in A$$

$$Q^{\pi}(\mathbf{x}, a) = \mathbf{E}_{\boldsymbol{S}|(S_0=\mathbf{x},a_0=a)}\left[\sum_{t=0}^{+\infty} \gamma^t R(S_t)\right], \forall a \in A$$

关键问题：如何计算值函数？

一个直接的想法

- 以状态值函数为例（T步平均奖赏）

$$U_T^{\pi}(\mathbf{x}) = \mathbf{E}_{\boldsymbol{S}|S_0=\mathbf{x}}\left[\frac{1}{T}\sum_{t=1}^{T} R(S_t)\right]$$

- 思路： 值函数近似
  - 设计模型$U_{\boldsymbol{\theta}}(\mathbf{x})$：形式已知、参数未知
  - 通过蒙特卡罗采样，可获得 $U_T^{\pi}(\mathbf{x})$ 在有限个状态（样本）下的值（估计值）

$$(\mathbf{x}_1, u_1)$$
$$(\mathbf{x}_2, u_2)$$
$$\cdots\cdots$$
$$(\mathbf{x}_n, u_n)$$

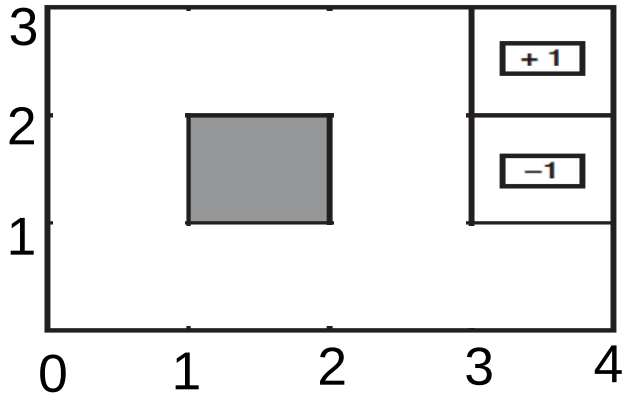
u_i为$U_T^{\pi}(\mathbf{x}_i)$的估计值

  - 寻找参数$\boldsymbol{\theta}$，使得

$$U_{\boldsymbol{\theta}}(\mathbf{x}_i) \approx u_i, i = 1, \cdots, n$$

回顾：学过的模型

- 线性模型： $U_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$



$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad U_{\mathbf{w}}(\mathbf{x}) = w_1 x + w_2 y + b$$

  - 广义线性模型
    - 联系函数
- 非线性模型
  - 核模型
  - 决策树
  - 神经网络

如何获得数据?

数据获取：蒙特卡罗采样

- 目标：获得状态值函数$U_T^{\pi}(\mathbf{x})$在有限个状态上的估计值
  - 在环境中执行策略π^{ϵ}(随机选初始状态)，获得N条轨迹

$$\tau_i: < \mathbf{x}_0^i, r_0^i, a_0^i, \mathbf{x}_1^i, r_1^i, a_1^i, \mathbf{x}_2^i, r_2^i, \cdots, a_{2T-2}^i, \mathbf{x}_{2T-1}^i, r_{2T-1}^i >$$
$$i = 1, \cdots, N$$

  - 计算T步平均奖赏

$$u_j^i = \frac{1}{T}\sum_{t=j+1}^{T+j} r_t^i$$
$$j = 0, \cdots, T-1, i = 1, \cdots, N$$

  - u_j^i为$U_T^{\pi}(\mathbf{x}_j^i)$的估计值

$$(\mathbf{x}_j^i, u_j^i)$$
$$i = 1, \ldots, N$$
$$j = 0,1, \cdots, T-1$$

拟合 ⟸ 回归模型

$T \times N$个训练样本

探索与利用

- 比非0初始化更一般的方法：探索与利用
- 探索（Exploration）
  - 跳出现有策略π的框架，尝试$\pi(s)$以外的动作
    - ✓ 随机探索：随机选择动作$a \in A(s)$
    - ✓ 理论上，只要随机探索次数足够多，可得到完整的真实模型$P(s'|s,a)$
- 利用（Exploitation）
  - 利用学到的模型，评估与改进现有的策略
- 只利用：陷入老套路，无法获得改进；只探索：无意义
- 必须在探索与利用之间进行权衡
  - 方法（一）：
    - 以ϵ概率随机选择动作$a \in A(s)$
    - 以$1-\epsilon$概率选择动作$\pi(s)$

π^{ϵ}

Q函数近似

- 为了得到最优策略，需对Q函数（状态-动作值函数）进行近似
- 思路与状态值函数近似类似
- 算法：

$$Q_T^{\pi}(\mathbf{x}, a) = \mathbf{E}_{\boldsymbol{S}|(S_0=\mathbf{x}, a_0=a)}\left[\frac{1}{T}\sum_{t=1}^{T} R(S_t)\right], \forall a \in A$$

蒙特卡罗法 + 线性回归（MC-LR）

Q函数近似模型

- 与状态值函数不同，Q函数无法由单一的线性模型近似

分别对应每个动作

$$W = [\mathbf{w}_1, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$$

$$Q_{\mathbf{w}_i}(\mathbf{x}, a_i) = \mathbf{w}_i^T \mathbf{x} \xrightarrow{\text{近似}} Q_T^\pi(\mathbf{x}, a_i)$$

$$i = 1, \cdots, m$$

- 动作编码

第i个动作a_i

$$\mathbf{e}_i \in \mathbb{R}^m, i = 1, \cdots m$$
$$[\mathbf{e}_i]_j = \delta(j - i), j = 1, \cdots, m$$

- 近似模型

$$Q_W(\mathbf{x}, a) = \mathbf{e}^T W^T \mathbf{x} \xrightarrow{\text{近似}} Q_T^\pi(\mathbf{x}, a), \forall a \in A$$

$\mathbf{e}$为动作a的编码

数据获取：蒙特卡罗采样

- 目标：获得Q函数$Q_T^{\pi}(\mathbf{x},a)$在有限状态上的估计值
  - 在环境中执行策略π^{ϵ}，获得N条轨迹

$$\tau_i: < \mathbf{x}_0^i, r_0^i, a_0^i, \mathbf{x}_1^i, r_1^i, a_1^i, \mathbf{x}_2^i, r_2^i, \cdots, a_{2T-2}^i, \mathbf{x}_{2T-1}, r_{2T-1} >$$
$$i = 1, \cdots, N$$

  - 计算T步平均奖赏

$$u_j^i = \frac{1}{T}\sum_{t=j+1}^{T+j} r_t^i$$
$$j = 0, \cdots, T-1, i = 1, \cdots, N$$

  - u_j^i为$Q_T^{\pi}\left(\mathbf{x}_j^i, a_j^i\right)$的估计值

$$(\mathbf{x}_j^i, a_j^i, u_j^i)$$
$$i = 1, \ldots, N$$
$$j = 0,1, \cdots, T-1$$

$T \times N$个训练样本

拟合

模型

$$Q_W(\mathbf{x},a) = \mathbf{e}^T W^T \mathbf{x}$$

MC-LR算法

- 注意：策略由Q函数决定

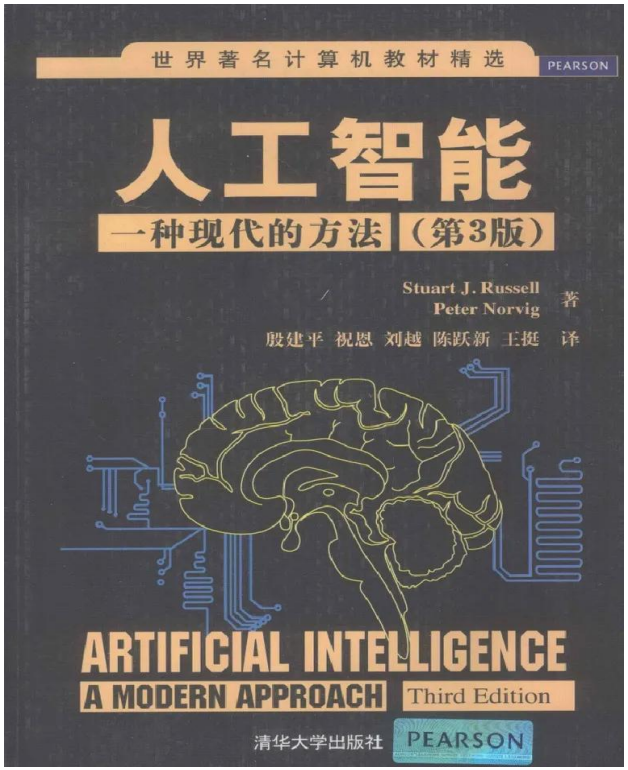
$$\pi(\mathbf{x}) = \arg\max_{a \in A} Q_W(\mathbf{x}, a), \forall \mathbf{x}$$

$$Q_W(\mathbf{x}, a) = \mathbf{e}^T W^T \mathbf{x}$$

- 算法框架
  - 初始化W
  - ① 在环境中执行策略π^ϵ，获得N条轨迹
  - ② 抽取数据
  - ③ 求解线性回归问题，更新参数W
  - 重复步骤1—3

本节课安排

- 连续状态
  - 值函数近似
  - 策略搜索



策略搜索

- MC-LR等值函数近似方法的共同点：
  - 设计模型$Q_{\boldsymbol{\theta}}(\mathbf{x}, a)$，从数据中学习出参数$\boldsymbol{\theta}$，从而得到真实最优Q函数$Q^*(\mathbf{x}, a)$的近似
  - 运用MEU准则，得到好的策略

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}) = \arg\max_{a \in A} Q_{\boldsymbol{\theta}}(\mathbf{x}, a)$$

- 策略搜索(Policy Search) 的思路：
  - 根据策略的表现直接求出参数$\boldsymbol{\theta}$
  - 对应的Q函数不一定与真实最优Q函数接近，比如：

$$Q_{\boldsymbol{\theta}}(\mathbf{x}, a) = \frac{Q^*(\mathbf{x}, a)}{10}$$

- 由于$\pi_{\boldsymbol{\theta}}(\mathbf{x})$是$\boldsymbol{\theta}$的非连续函数，需使用随机性策略
  - softmax函数

$$\pi_{\boldsymbol{\theta}}(\mathbf{x}, a) = \frac{e^{Q_{\boldsymbol{\theta}}(\mathbf{x}, a)}}{\sum_{a'} e^{Q_{\boldsymbol{\theta}}(\mathbf{x}, a')}}$$

策略搜索：算法框架

- 如何衡量策略的表现？
  - 策略评估：计算状态值函数

$$U_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})$$

  - 综合考虑所有状态，可得策略价值函数(policy value)

$$\rho(\boldsymbol{\theta}) = \mathbf{E}_{\mathbf{x}}\left[U_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x})\right] = \mathbf{E}_{\mathbf{x}}\left[\sum_{a\in A}\pi_{\boldsymbol{\theta}}(\mathbf{x},a)Q_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},a)\right]$$

- 参数寻优

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta})$$

- 随机梯度上升（stochastic gradient-ascent，SGA）

$$J(\theta) = \sum_{a\in A}\pi_{\boldsymbol{\theta}}(\mathbf{x},a)Q_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x},a)$$

$$\theta \leftarrow \theta + \alpha\Delta J(\theta),\ \ \Delta J(\theta)\text{为近似梯度}$$

REINFORCE算法

$$J(\theta) = \sum_{a \in A} \pi_{\boldsymbol{\theta}}(\mathbf{x}, a) Q_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, a)$$

$$\Delta J(\theta) = \sum_{a \in A} (\Delta \pi_{\boldsymbol{\theta}}(\mathbf{x}, a)) Q_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, a)$$

$$= \sum_{a \in A} \frac{\Delta \pi_{\boldsymbol{\theta}}(\mathbf{x}, a)}{\pi_{\boldsymbol{\theta}}(\mathbf{x}, a)} \pi_{\boldsymbol{\theta}}(\mathbf{x}, a) Q_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, a)$$

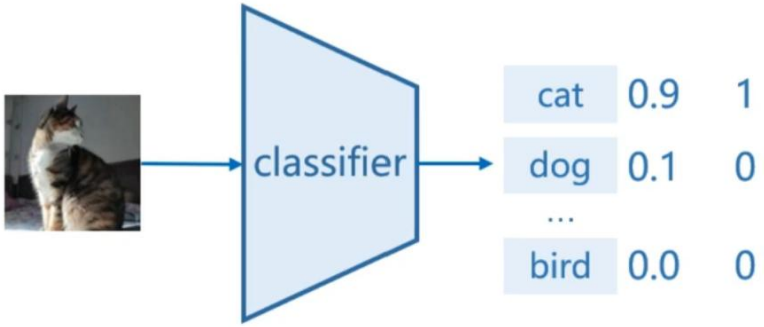
$$= \sum_{a \in A} \pi_{\boldsymbol{\theta}}(\mathbf{x}, a) Q_T^{\pi_{\boldsymbol{\theta}}}(\mathbf{x}, a) \Delta \log(\pi_{\boldsymbol{\theta}}(\mathbf{x}, a))$$

可用蒙特卡罗方法获得

强化学习小结

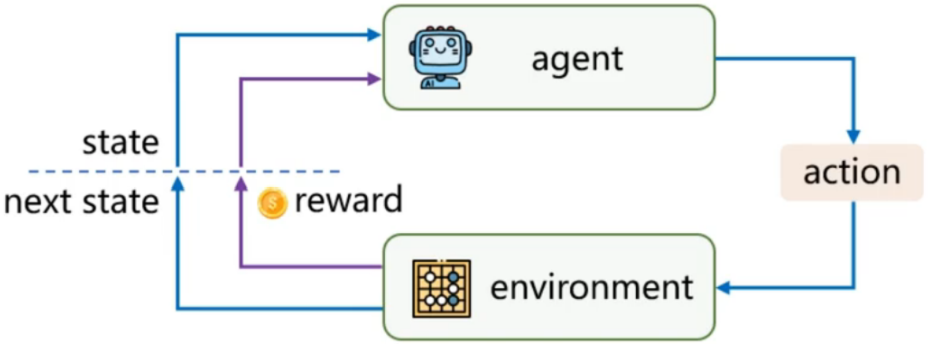
- 马尔可夫决策过程 MDP：$< S, A, P, R >$（数学基础）
- 状态离散且有限、动作离散且有限
  - 环境模型P已知
    - ✓ 值迭代、策略迭代
  - 环境模型P未知
    - ✓ 基于模型估计的方法
    - ✓ 免模型方法：TD、Q-Learning、蒙特卡罗法
- 状态空间连续、动作离散且有限、环境模型P未知
  - 值函数近似：MC-LR
  - 策略搜索：REINFORCE

监督学习 vs 强化学习



监督学习

寻找一个模型来
拟合映射关系



强化学习

寻找一个策略来
最大化累计奖励

2025 Turing Award

Dr. Richard Sutton

Andrew Barto

ACM A.M. TURING AWARD HONORS TWO RESEARCHERS WHO LED THE DEVELOPMENT OF CORNERSTONE AI TECHNOLOGY

Andrew Barto and Richard Sutton Recognized as Pioneers of Reinforcement Learning

ACM, the Association for Computing Machinery, today named Andrew Barto and Richard Sutton as the recipients of the 2024 ACM A.M. Turing Award for developing the conceptual and algorithmic foundations of reinforcement learning. In a series of papers beginning in the 1980s, Barto and Sutton introduced the main ideas, constructed the mathematical foundations, and developed important algorithms for reinforcement learning—one of the most important approaches for creating intelligent systems.

Barto is Professor Emeritus of Information and Computer Sciences at the University of Massachusetts, Amherst. Sutton is a Professor of Computer Science at the University of Alberta and a Research Scientist at Keen Technologies.

The ACM A.M. Turing Award, often referred to as the "Nobel Prize in Computing," carries a $1 million prize with financial support provided by Google, Inc. The award is named for Alan M. Turing, the British mathematician who articulated the mathematical foundations of computing.