强化学习2022
第2节

涉及知识点：
马尔可夫决策过程、基于动态规划的强化学习、
基于模型的强化学习

马尔可夫决策过程

张伟楠 – 上海交通大学

课程大纲

强化学习基础部分

1. 强化学习、探索与利用
2. MDP和动态规划
3. 值函数估计
4. 无模型控制方法
5. 规划与学习
6. 参数化的值函数和策略
7. 深度强化学习价值方法
8. 深度强化学习策略方法

强化学习前沿部分

9. 基于模型的深度强化学习
10. 模仿学习
11. 离线强化学习
12. 参数化动作空间
13. 目标导向的强化学习
14. 多智能体强化学习
15. 强化学习大模型
16. 技术交流与回顾

马尔可夫决策过程

讲师：张伟楠 – 上海交通大学

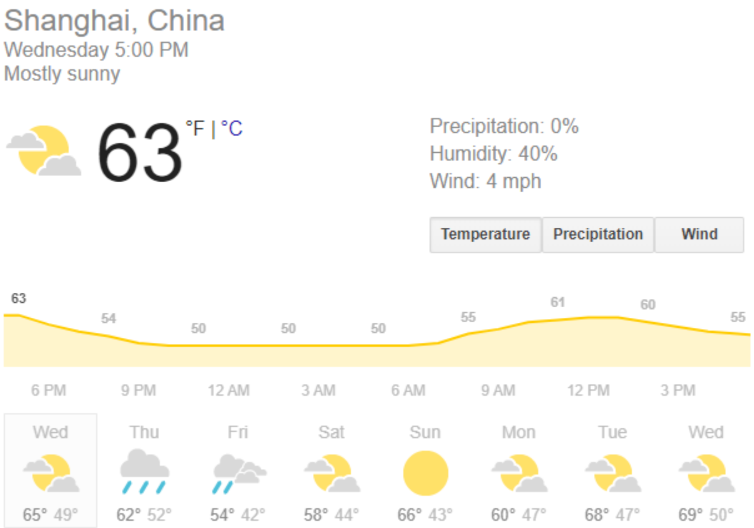
随机过程

- 随机过程是一个或多个事件、随机系统或者随机现象随时间发生演变的过程

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

  - 概率论研究静态随机现象的统计规律
  - 随机过程研究动态随机现象的统计规律

布朗运动

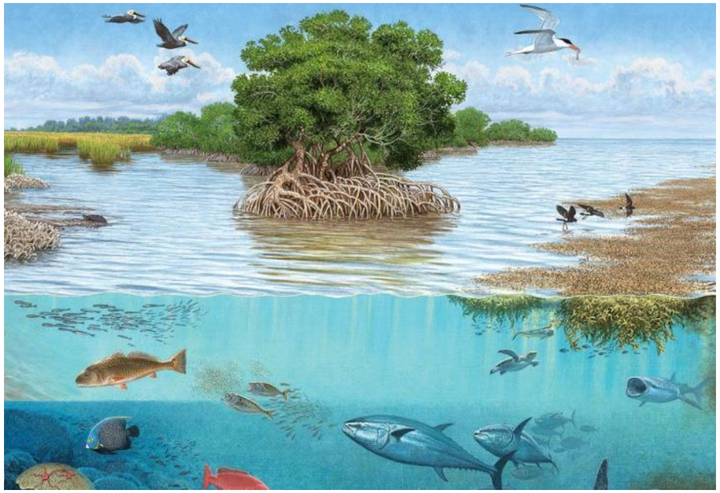


天气变化

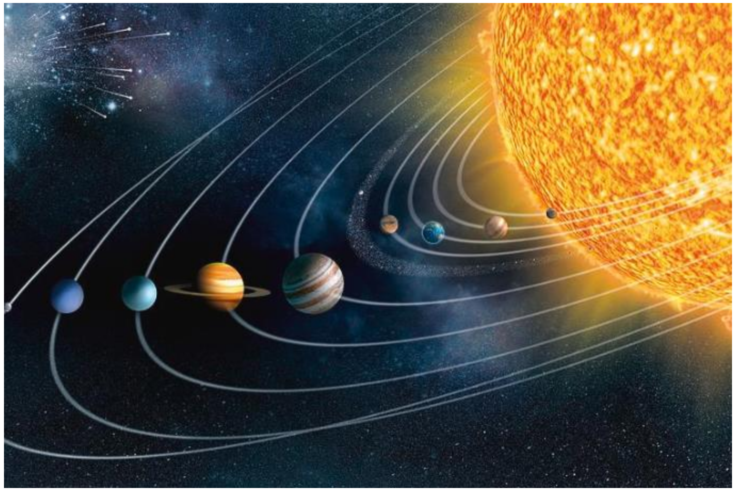
随机过程

足球比赛

城市交通



生态系统



星系

马尔可夫过程

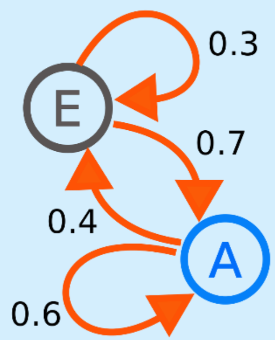
- 马尔可夫过程（Markov Process）是具有马尔可夫性质的随机过程

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

- 定义：
  - 状态S_t是马尔可夫的，当且仅当

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

- 性质：
  - 状态从历史（history）中捕获了所有相关信息
  - 当状态已知的时候，可以抛开历史不管
  - 也就是说，当前状态是未来的充分统计量



一个具有两个转换状态的马尔可夫链

马尔可夫决策过程

- 马尔可夫决策过程（Markov Decision Process，MDP）
  - 提供了一套为在结果部分随机、部分在决策者的控制下的决策过程建模的数学框架

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

$$\mathbb{P}[S_{t+1}|S_t, A_t]$$

- MDP形式化地描述了一种强化学习的环境
  - 环境完全可观测
  - 即，当前状态可以完全表征过程（马尔可夫性质）

MDP五元组

- MDP可以由一个五元组表示 $(S, A, \{P_{sa}\}, \gamma, R)$
  - S是状态的集合
    - 比如，迷宫中的位置，Atari游戏中的当前屏幕显示
  - A是动作的集合
    - 比如，向N、E、S、W移动，手柄操纵杆方向和按钮
  - P_{sa}是状态转移概率
    - 对每个状态$s \in S$和动作$a \in A$，P_{sa}是下一个状态在S中的概率分布
  - $\gamma \in [0,1]$是对未来奖励的折扣因子
  - $R: S \times A \longmapsto \mathbb{R}$ 是奖励函数
    - 有时奖励只和状态相关

MDP的动态

- MDP的动态如下所示：
  - 从状态s_0开始
  - 智能体选择某个动作$a_0 \in A$
  - 智能体得到奖励$R(s_0, a_0)$
  - MDP随机转移到下一个状态$s_1 \sim P_{s_0 a_0}$
  - 这个过程不断进行

$$s_0 \xrightarrow{a_0, R(s_0, a_0)} s_1 \xrightarrow{a_1, R(s_1, a_1)} s_2 \xrightarrow{a_2, R(s_2, a_2)} s_3 \quad \cdots$$

  - 直到终止状态s_T出现为止，或者无止尽地进行下去
  - 智能体的总回报为

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots$$

MDP的动态性

- 在大部分情况下，奖励只和状态相关
  - 比如，在迷宫游戏中，奖励只和位置相关
  - 在围棋中，奖励只基于最终所围地盘的大小有关
- 这时，奖励函数为 $R(s): S \mapsto \mathbb{R}$
- MDP的过程为

$$s_0 \xrightarrow{a_0, R(s_0)} s_1 \xrightarrow{a_1, R(s_1)} s_2 \xrightarrow{a_2, R(s_2)} s_3 \quad \cdots$$

- 累积奖励为

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

REVIEW: 在与动态环境的交互中学习

有监督、无监督学习

Model ←

Fixed Data

强化学习

Agent ↔

Dynamic Environment

和动态环境交互产生的数据分布

- 给定同一个动态环境（即MDP），不同的策略采样出来的(状态-行动)对的分布是不同的

- 占用度量（Occupancy Measure）

$$\rho^{\pi}(s,a) = \mathbb{E}_{a\sim\pi(s),\, s'\sim p(s,a)}\left[\sum_{t=0}^{T} \gamma^{t}\, p(s_t = s, a_t = a)\right]$$

占用度量和策略

- 占用度量（Occupancy Measure）

$$\rho^{\pi}(s,a) = \mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t = s, a_t = a)\right]$$

- 定理1：和同一个动态环境交互的两个策略π_1和π_2得到的占用度量ρ^{π_1}和ρ^{π_2}满足

$$\rho^{\pi_1} = \rho^{\pi_2} \ \text{ iff } \ \pi_1 = \pi_2$$

- 定理2：给定一占用度量ρ，可生成该占用度量的唯一策略是

$$\pi_{\rho} = \frac{\rho(s,a)}{\sum_{a'}\rho(s,a')}$$

U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. ICML 2008.

占用度量和策略

- 占用度量（Occupancy Measure）

$$\rho^{\pi}(s,a)=\mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t=s,a_t=a)\right]$$

- 状态占用度量

$$\rho^{\pi}(s)=\mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t=s)\right]$$

$$=\mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t=s)\sum_{a'}p(a_t=a'|s_t=s)\right]$$

$$=\sum_{a'}\mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t=s,a_t=a'))\right]$$

$$=\sum_{a'}\rho^{\pi}(s,a')$$

U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. ICML 2008.

占用度量和累计奖励

- 占用度量（Occupancy Measure）

$$\rho^{\pi}(s,a) = \mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t = s, a_t = a)\right]$$

- 策略的累积奖励为

$$V(\pi) = \mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}[R(s_0,a_0) + \gamma R(s_1,a_1) + \gamma^2 R(s_2,a_2) + \cdots]$$

$$= \sum_{s,a} \mathbb{E}_{a\sim\pi(\cdot|s),\, s'\sim p(\cdot|s,a)}\left[\sum_{t=0}^{T}\gamma^{t}\, p(s_t = s, a_t = a)\right] R(s,a)$$

$$= \sum_{s,a}\rho^{\pi}(s,a)R(s,a) \;=\; \mathbb{E}_{\pi}[R(s,a)]$$

强化学习中的简写

U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. ICML 2008.

基于动态规划的强化学习

讲师：张伟楠 – 上海交通大学

MDP目标和策略

- 目标：选择能够最大化累积奖励期望的动作

$$\mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots]$$

  - $\gamma \in [0,1]$是未来奖励的折扣因子，使得和未来奖励相比起来智能体更重视即时奖励
    - 以金融为例，今天的\$1比明天的\$1更有价值
- 给定一个特定的策略 $\pi(s): S \rightarrow A$
  - 即，在状态 s 下采取动作 $a = \pi(s)$
- 给策略π定义价值函数

$$V^{\pi}(s) = \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots | s_0 = s, \pi]$$

  - 即，给定起始状态和根据策略π采取动作时的累积奖励期望

价值函数的Bellman等式

- 给策略π定义价值函数

$$V^{\pi}(s) = \mathbb{E}[R(s_0) + \underbrace{\gamma R(s_1) + \gamma^2 R(s_2) + \cdots}_{\gamma V^{\pi}(s_1)} | s_0 = s, \pi]$$

$$= R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

Bellman等式

立即奖励

状态转移

时间折扣

下一个状态的价值

最优价值函数

- 对状态s来说的最优价值函数是所有策略可获得的最大可能折扣奖励的和

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

- 最优价值函数的Bellman等式

$$V^*(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s')$$

- 最优策略

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V^*(s')$$

- 对状态s和策略π

$$V^*(s) = V^{\pi^*}(s) \geq V^{\pi}(s)$$

价值迭代和策略迭代

- 价值函数和策略相关

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s')V^{\pi}(s')$$

$$\pi(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s')V^{\pi}(s')$$

- 可以对最优价值函数和最优策略执行迭代更新
  - 价值迭代
  - 策略迭代

价值迭代

- 对于一个动作空间和状态空间有限的MDP

$$|S| < \infty, |A| < \infty$$

- 价值迭代过程

1. 对每个状态s，初始化 $V(s) = 0$
2. 重复以下过程直到收敛 {

   对每个状态，更新

$$V(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s')V(s')$$

}

注意：在以上的计算中没有明确的策略

价值迭代收敛性证明：https://towardsdatascience.com/mathematical-analysis-of-reinforcement-learning-bellman-equation-ac9f0954e19f

同步 vs. 异步价值迭代

- 同步的价值迭代会储存两份价值函数的拷贝
  1. 对S中的所有状态s

$$V_{new}(s) \leftarrow \max_{a \in A}\left(R(s) + \gamma \sum_{s' \in S} P_{sa}(s')V_{old}(s')\right)$$

  2. 更新 $V_{old}(s) \leftarrow V_{new}(s)$

- 异步价值迭代只储存一份价值函数
  1. 对S中的所有状态s

$$V(s) \leftarrow \max_{a \in A}\left(R(s) + \gamma \sum_{s' \in S} P_{sa}(s')V(s')\right)$$

价值迭代例子：最短路径

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

策略迭代

- 对于一个动作空间和状态空间有限的MDP

$$|S| < \infty, |A| < \infty$$

- 策略迭代过程

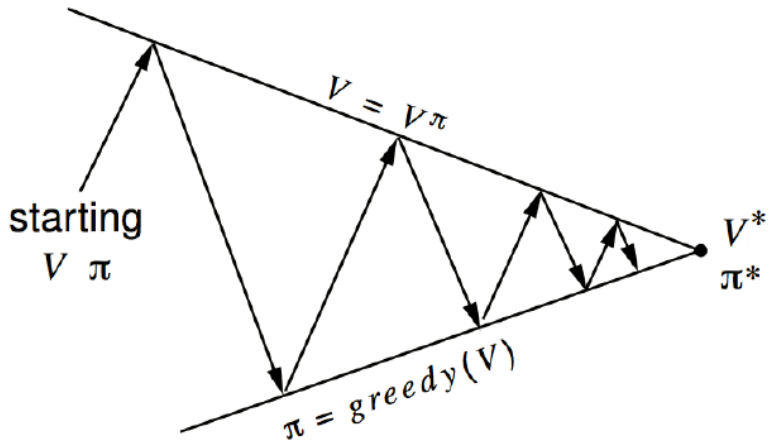
1. 随机初始化策略 π
2. 重复以下过程直到收敛{
   a) 让 $V := V^{\pi}$
   b) 对每个状态，更新

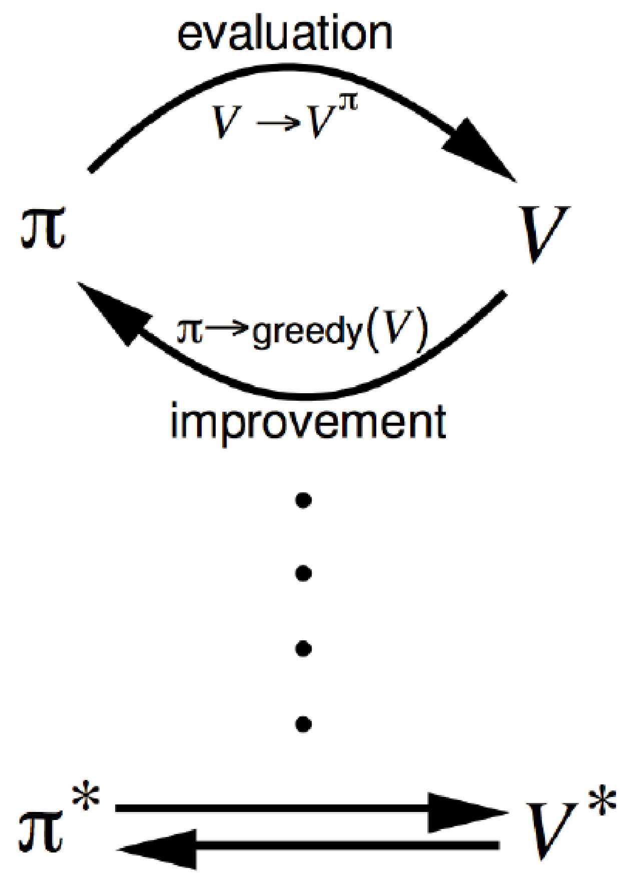
$$\pi(s) = arg \max_{a \in A} \sum_{s' \in S} P_{sa}(s') V(s')$$

}

更新价值函数会很耗时

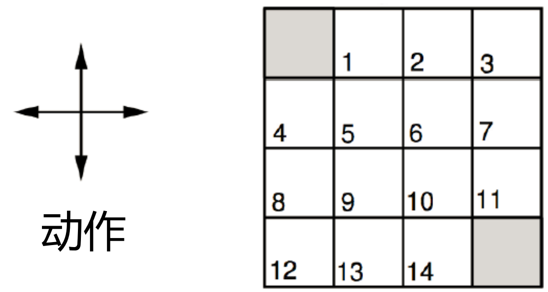
策略迭代





- 策略评估
  - 估计V^{π}
  - 迭代的评估策略
- 策略改进
  - 生成 $\pi' \geq \pi$
  - 贪心策略改进

举例：策略评估



- 非折扣MDP（$\gamma = 1$）
- 非终止状态：1, 2, ...,14
- 两个终止状态（灰色方格）
- 如果动作指向所有方格以外，则这一步不动
- 奖励均为-1，直到到达终止状态
- 智能体的策略为均匀随机策略

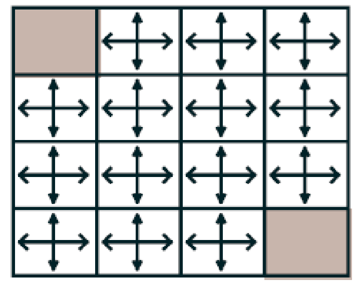
$$\pi(n|\cdot) = \pi(e|\cdot) = \pi(s|\cdot) = \pi(w|\cdot) = 0.25$$

举例：策略评估

随机策略的 V_k　　　V_k对应的贪心策略

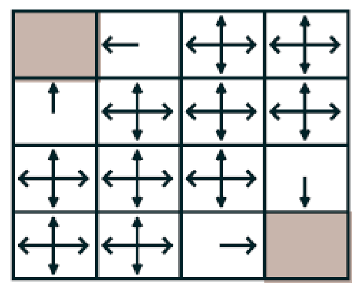
K=0

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



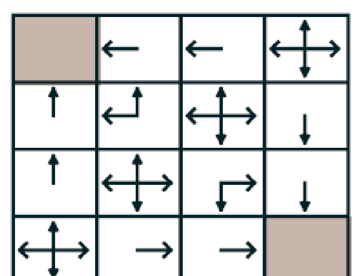
K=1

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0



K=2

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

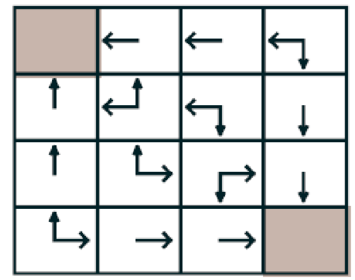


举例：策略评估

随机策略的 V_k | V_k对应的贪心策略

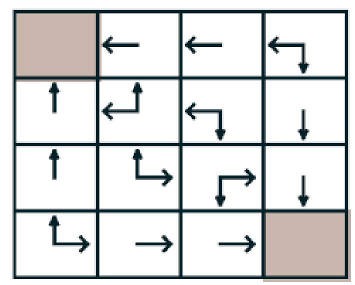
K=3

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0



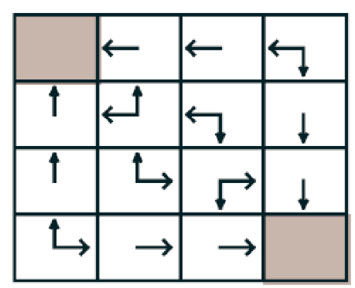
K=10

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0



K=∞

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



$V := V^{\pi}$
最优策略

价值迭代 vs. 策略迭代

价值迭代

1. 对每个状态s，初始化 $V(s) = 0$
2. 重复以下过程直到收敛 {
   对每个状态，更新
   $$V(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s')V(s')$$
   }

策略迭代

1. 随机初始化策略 π
2. 重复以下过程直到收敛 {
   a) 让 $V := V^{\pi}$
   b) 对每个状态，更新
   $$\pi(s) = arg \max_{a \in A} \sum_{s' \in S} P_{sa}(s')V(s')$$
   }

备注：

1. 价值迭代是贪心更新法
2. 策略迭代中，用Bellman等式更新价值函数代价很大
3. 对于空间较小的MDP，策略迭代通常很快收敛
4. 对于空间较大的MDP，价值迭代更实用（效率更高）
5. 如果没有状态转移循环，最好使用价值迭代

基于模型的强化学习

讲师：张伟楠 – 上海交通大学

学习一个MDP模型

- 目前我们关注在给出一个已知MDP模型后：（也就是说，状态转移 $P_{sa}(s')$和奖励函数$R(s)$明确给定后）
  - 计算最优价值函数
  - 学习最优策略
- 在实际问题中，状态转移和奖励函数一般不是明确给出的
  - 比如，我们只看到了一些episodes

$$\text{Episode1:}\quad s_0^{(1)} \xrightarrow{a_0^{(1)}, R(s_0)^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)}, R(s_1)^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}, R(s_2)^{(1)}} s_3^{(1)} \cdots s_T^{(1)}$$

$$\text{Episode2:}\quad s_0^{(2)} \xrightarrow{a_0^{(2)}, R(s_0)^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)}, R(s_1)^{(2)}} s_2^{(2)} \xrightarrow{a_2^{(2)}, R(s_2)^{(2)}} s_3^{(2)} \cdots s_T^{(2)}$$

学习一个MDP模型

Episode1： $s_0^{(1)} \xrightarrow{a_0^{(1)},R(s_0)^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)},R(s_1)^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)},R(s_2)^{(1)}} s_3^{(1)} \cdots s_T^{(1)}$

Episode2： $s_0^{(2)} \xrightarrow{a_0^{(2)},R(s_0)^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)},R(s_1)^{(2)}} s_2^{(2)} \xrightarrow{a_2^{(2)},R(s_2)^{(2)}} s_3^{(2)} \cdots s_T^{(2)}$

$\vdots \qquad\qquad\qquad\qquad \vdots$

- 从“经验”中学习一个MDP模型
  - 学习状态转移概率$P_{sa}(s')$

$$P_{sa}(s') = \frac{\text{在}s\text{下采取动作}a\text{并转移到}s'\text{的次数}}{\text{在}s\text{下采取动作}a\text{的次数}}$$

  - 学习奖励函数$R(s)$，也就是立即奖赏期望

$$R(s) = average\{R(s)^{(i)}\}$$

学习模型&优化策略

算法

1. 随机初始化策略π

2. 重复以下过程直到收敛 {

   a) 在MDP中执行π，收集经验数据

   b) 使用MDP中的累积经验更新对P_{sa}和R的估计

   c) 利用对P_{sa}和R的估计执行价值迭代，得到新的估计价值函数V

   d) 根据V更新策略π为贪心策略

}

学习一个MDP模型

- 在实际问题中，状态转移和奖励函数一般不是明确给出的
  - 比如，我们只看到了一些episodes

Episode1： $s_0^{(1)} \xrightarrow{a_0^{(1)}, R(s_0)^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)}, R(s_1)^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}, R(s_2)^{(1)}} s_3^{(1)} \cdots s_T^{(1)}$

Episode2： $s_0^{(2)} \xrightarrow{a_0^{(2)}, R(s_0)^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)}, R(s_1)^{(2)}} s_2^{(2)} \xrightarrow{a_2^{(2)}, R(s_2)^{(2)}} s_3^{(2)} \cdots s_T^{(2)}$

- 另一种解决方式是不学习MDP，从经验中直接学习价值函数和策略
  - 也就是模型无关的强化学习（Model-free Reinforcement Learning）

马尔可夫决策过程总结

- MDP由一个五元组构成 $(S, A, \{P_{sa}\}, \gamma, R)$，其中状态转移$P$和奖励函数$R$构成了动态系统

- 动态系统和策略交互的占用度量

$$\rho^{\pi}(s,a) = \mathbb{E}_{a\sim\pi(s),\, s'\sim p(s,a)}\left[\sum_{t=0}^{T} \gamma^{t}\, p(s_t = s, a_t = a)\right]$$

- 一个白盒环境给定的情况下，可用动态规划的方法求解最优策略
  - 值迭代和策略迭代

- 如果环境是黑盒的，可以根据统计信息来拟合出动态环境P和R，然后做动态规划求解最优策略

THANK YOU