



《强化学习》课程之第四讲（2021年春季研究生）

动态规划法

苏州大学计算机科学与技术学院

主讲：刘全

目 录

4

前言

4.1

策略迭代

4.2

值迭代

4.3

广义策略迭代

4.4

小结

4. 前言 (1)

- 在强化学习中，**动态规划法**（dynamic programming, DP）主要用于求解有模型的MDP问题。
- 尽管在现实任务中难以获得完备的环境模型，且**动态规划**需要消耗大量的**计算资源**，但是作为强化学习的基础，动态规划法仍然具有非常重要的理论意义。

前言 (2)

➤ 事实上，所有其他强化学习方法，如**蒙特卡洛法**（Monte Carlo, MC）、**时序差分法**（Temporal Difference, TD）等，都是对动态规划法的一种近似，只是在学习过程中不再需要完整的环境模型，而且在计算资源的消耗方面也可以大幅度减少。

前言 (3)

➤ 动态规划法：利用值函数来评价策略。

✓ 基于模型的策略迭代

✓ 基于模型的值迭代

目 录

4

前言

4.1

策略迭代

4.2

值迭代

4.3

广义策略迭代

4.4

小结

4.1 策略迭代 (1)

➤ 策略迭代

通过构建策略的值函数（状态值函数或动作值函数）来评估当前策略，并利用这些值函数给出改进的新策略。策略迭代由策略评估（PE）和策略改进（PI）两部分组成。

4.1 策略迭代 (2)

➤ 策略评估:

每一次策略评估都是一个迭代过程，对于一个给定的策略 π ，评估在该策略下，所有状态 s (或状态-动作对 (s, a)) 的值函数 $v_\pi(s)$ 或 $q_\pi(s, a)$ 。

➤ 策略改进:

在策略评估的基础上，直接利用策略 π 的动作值函数，然后通过贪心策略 (或 ε -贪心策略) 对策略进行改进。

4.1 策略迭代 (3)

➤ 链式关系

根据策略 π 的值函数 v_π (或 q_π) 产生一个更优策略 π' ,
再根据策略 π' 的值函数 $v_{\pi'}$ (或 $q_{\pi'}$) 得到一个更优策略 π'' ,
以此类推, 通过这样的链式方法可以得到一个关于策略和
值函数的更新序列, 并且能够保证每一个新策略都比前一个
策略更优 (除非前一个策略已是最优策略)。

4.1 策略迭代 (4)

➤ 链式关系

在有限MDP中，策略有限，所以在多次迭代后，一定能收敛到最优策略和最优值函数。

$$\pi_0 \xrightarrow{\text{PE}} v_{\pi_0} \xrightarrow{\text{PI}} \pi_1 \xrightarrow{\text{PE}} v_{\pi_1} \xrightarrow{\text{PI}} \dots \xrightarrow{\text{PI}} \pi_* \xrightarrow{\text{PE}} v_*$$

注：策略 π 的下标 $0, 1, 2, \dots$ 表示迭代更新的次序。

4.1 策略迭代 (5)

➤4.1.1 策略评估

✓1. 基于状态值函数的策略评估(v 值)

每一次策略评估都是一个迭代过程，对于一个给定的策略 π ，评估在该策略下，所有状态 s 的状态值函数或动作值函数。

4.1 策略迭代 (6)

➤ 基于状态值函数的策略评估迭代式

可以将状态值函数的**贝尔曼方程**转化为迭代式：

$$\begin{aligned} v_{\tau}(s) &= \mathbb{E}_{\pi} \left(R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s \right) \\ &= \sum_a \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a) [r + \gamma v_{\tau-1}(s')] \end{aligned}$$

初始值 v_0 可以为任意值（通常设为0）。在使用 $v_{\tau}(s)$ 这样的迭代式时，默认采用同一策略 π ，为了突出迭代关系，可以省略 $v_{\pi}(s)$ 的下标 π 。

4.1 策略迭代 (7)

➤ 提前结束迭代的两种方法

- ✓ 直接设置迭代次数。只要达到预期的迭代次数，即可停止迭代；
- ✓ 设定较小的阈值（次优界限） θ 。当 $\left|v_{\tau}(s) - v_{\tau-1}(s)\right|_{\infty} < \theta$ 时，停止迭代。比较两次迭代的状态值函数差的绝对值 Δ （或差的平方），当 Δ 最大值小于阈值时，终止迭代。

4.1 策略迭代 (8)

➤ 期望更新法 (expected update)

$$\begin{aligned} v_{\tau}(s) &= \mathbb{E}_{\pi} (R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s) \\ &= \sum_a \pi(a | s) \sum_{s', r} p(s', r | s, a) [r + \gamma v_{\tau-1}(s')] \end{aligned}$$

根据给定的策略（动作分布），采取可能的动作，得到**单步转移**后的所有状态 s' 和奖励 r ，并利用下一状态 s' 的值函数 $v_{\tau-1}(s')$ 。通过分布的期望值，更新状态 s 的值函数 $v_{\tau}(s)$ 。

4.1 策略迭代 (9)

算法 4.1 基于状态值函数的策略评估算法	
输 入	初始策略 $\pi(a s)$, 动态性 p , 奖赏函数 r , 折扣因子 γ
初始化	<ol style="list-style-type: none">1. 对任意 $s \in \mathcal{S}$, 初始化状态值函数, 如 $V(s)=0$2. 阈值 θ 设置为一个较小的实数值, 如 $\theta=0.01$
策 略 评 估	<ol style="list-style-type: none">3. repeat 对每一轮策略评估 $\tau=1, 2, \dots$4. $\delta \leftarrow 0$5. for 每个状态 s do6. $v \leftarrow V(s)$7. $V(s) \leftarrow \sum_a \pi(a s) \sum_{s',r} p(s',r s,a) [r + \gamma V(s')]$8. $\delta \leftarrow \max(\delta, v - V(s))$9. end for10. until $\delta < \theta$
输 出	$v_\pi = V$

4.1 策略迭代 (10)

➤ 异步计算方式:

在相邻的两个迭代轮次 τ 和 $\tau - 1$ ，保存同一组状态值函数 $V(s)$ 。在 $V(s)$ 中，存储两轮混合的函数值。因此在每次计算中，如果状态 s 的值函数已被更新，那么当用到 $V(s)$ 时，就使用已经更新过的数据。

评估过程中，中间结果与状态评估的先后次序密切相关。

4.1 策略迭代 (11)

➤ 同步计算方式:

在每一迭代轮次 τ ，都保存相邻两轮的状态值函数: $V_\tau(s)$ 和 $V_{\tau-1}(s)$ 。在计算 $V_\tau(s)$ 过程中，使用的全部是上一轮的 $V_{\tau-1}(s)$ 值。

评估过程中，中间结果与状态评估的先后次序无关。

- ✓ 在相同情况下，利用两种迭代方式评估，收敛后结果是相同的。
- ✓ 但收敛速度方面相比较，通常异步计算方式收敛速度会更快。

4.1 策略迭代 (12)

- 算法4.1可用于有穷状态空间的确定MDP问题和随机MDP问题，这具体体现在MDP的状态转移动态： $p(s', r | s, a)$
- 对于确定MDP问题，在当前状态 s 下采取动作 a ，到达下一状态 s' 的概率为1，而到达其他状态的概率均为0。
- 对于随机MDP问题，在当前状态 s 下采取动作 a ，到达下一状态 s' 是随机的。

4.1 策略迭代 (13)

- ✓ 因此利用算法4.1解决确定MDP问题和随机MDP时，只是状态转移动态的变化，而算法本身不需要改变。
- ✓ 这也体现出确定MDP是随机MDP的特例。

4.1 策略迭代 (14)

➤例4.1 对确定环境扫地机器人任务进行策略评估。

- ✓ 机器人在非终止状态（除位置0、12、19）均采取等概率策略：

$$\pi(a|s) = 1/|\mathcal{A}(s)|$$

- ✓ 扫地机器人最多可以采取4个动作。

$$\{Up, Down, Left, Right\}$$

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

- ✓ 奖赏：

$$r(s,a) = \begin{cases} +3, & \text{如果 } s \neq [5,4] \text{ 且 } s+a=[5,4] \\ +1, & \text{如果 } s \neq [1,1] \text{ 且 } s+a=[1,1] \\ -10, & \text{如果 } s+a=[3,3] \\ 0, & \text{其他} \end{cases}$$

4.1 策略迭代 (15)

➤ 评估过程 (1)

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$\tau = 0$

0.01	-0.13	-0.73	-0.27	1.39
0.02	-0.49	-2.60	-2.29	0.00
0.09	-2.46		-2.60	0.27
0.33	0.13		-0.49	-0.13
0.00	0.33	0.09	0.02	0.01

$\tau = 1$

$\tau = 1$:

$$\begin{aligned}
 V(S_{24}) &= \frac{1}{2} * 1 * (r_1 + \gamma V(S_{19})) + \frac{1}{2} * 1 * (r_2 + \gamma V(S_{23})) \\
 &= \frac{1}{2} * 1 * (3 + 0.8 * 0) + \frac{1}{2} * 1 * (0 + 0.8 * (-0.27)) \\
 &\approx 1.39
 \end{aligned}$$

$$\begin{aligned}
 V(S_{17}) &= \frac{1}{4} * 1 * (r_1 + \gamma V(S_{22})) + \frac{1}{4} * 1 * (r_2 + \gamma V(S_{17})) \\
 &\quad + \frac{1}{4} * 1 * (r_3 + \gamma V(S_{16})) + \frac{1}{4} * 1 * (r_4 + \gamma V(S_{18})) \\
 &= 0.25 * (0 + 0) + 0.25 * (-10 + 0) + 0.25 * (0 + (-0.49)) + 0.25 * (0 + 0) \\
 &\approx -2.60
 \end{aligned}$$

$$\begin{aligned}
 V(S_5) &= \frac{1}{3} * 1 * (r_1 + \gamma V(S_{10})) + \frac{1}{3} * 1 * (r_2 + \gamma V(S_0)) + \frac{1}{3} * 1 * (r_3 + \gamma V(S_6)) \\
 &= \frac{1}{3} * 1 * (0 + 0) + \frac{1}{3} * 1 * (1 + 0) + \frac{1}{3} * 1 * (0 + 0) \\
 &\approx 0.33
 \end{aligned}$$

4.1 策略迭代 (16)

➤ 评估过程

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

0.01	-0.13	-0.73	-0.27	1.39
0.02	-0.49	-2.60	-2.29	0.00
0.09	-2.46		-2.60	0.27
0.33	0.13	-2.46	-0.49	-0.13
0.00	0.33	0.09	0.02	0.01

-0.16	-0.58	-1.15	-0.11	1.46
-0.27	-1.27	-3.48	-0.65	0.00
-0.54	-3.36		-3.28	0.04
0.39	-0.83	-3.36	-1.27	-0.31
0.00	0.39	-0.54	-0.27	-0.16

$\tau = 2$:

$\tau = 1$

$\tau = 2$

$$\begin{aligned} V(S_{24}) &= \frac{1}{2} * 1 * (r_1 + \gamma V(S_{19})) + \frac{1}{2} * 1 * (r_2 + \gamma V(S_{23})) \\ &= 0.5 * (3 + 0.8 * 0) + 0.5 * (0 + 0.8 * (-0.11)) \\ &= 1.46 \end{aligned}$$

$$\begin{aligned} V(S_{17}) &= \frac{1}{4} * 1 * (r_1 + \gamma V(S_{22})) + \frac{1}{4} * 1 * (r_2 + \gamma V(S_{17})) \\ &\quad + \frac{1}{4} * 1 * (r_3 + \gamma V(S_{16})) + \frac{1}{4} * 1 * (r_4 + \gamma V(S_{18})) \\ &= 0.25 * (0 + 0.8 * (-0.73)) + 0.25 * (-10 + 0.8 * (-2.60)) \\ &\quad + 0.25 * (0 + 0.8 * (-1.27)) + 0.25 * (0 + 0.8 * (-2.29)) \\ &\approx -3.48 \end{aligned}$$

$$\begin{aligned} V(S_5) &= \frac{1}{3} * 1 * (r_1 + \gamma V(S_{10})) + \frac{1}{3} * 1 * (r_2 + \gamma V(S_0)) + \frac{1}{3} * 1 * (r_3 + \gamma V(S_6)) \\ &= \frac{1}{3} * (0 + 0.8 * 0.09) + \frac{1}{3} * 1 * (0 + 0.8 * 0) + \frac{1}{3} * (0 + 0.8 * 0.13) \\ &\approx 0.39 \end{aligned}$$

4.1 策略迭代 (17)

➤ 评估过程

- ✓ 当 $\tau = 30$ 时, $|V_\tau(s) - V_{\tau-1}(s)|_\infty < \theta$, $V_\tau(s)$ 认为已经**收敛**于 $v_\pi(s)$, 计算得到的 $v_\pi(s)$ 就是在策略 π 下的有效评估。

$$\pi(a|s) = 1/|\mathcal{A}(s)|:$$

-1.11	-1.36	-1.62	-0.33	1.37
-1.42	-2.37	-4.37	-0.99	0.00
-1.83	-4.72		-3.99	-0.30
-0.73	-2.16	-4.65	-2.16	-0.89
0.00	-0.72	-1.77	-1.28	-0.87

$$\tau = 30$$

4.1 策略迭代 (18)

➤ 每轮状态值函数的更新过程

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$\tau = 0$

0.01	-0.13	-0.73	-0.27	1.39
0.02	-0.49	-2.60	-2.29	0.00
0.09	-2.46		-2.60	0.27
0.33	0.13	-2.46	-0.49	-0.13
0.00	0.33	0.09	0.02	0.01

$\tau = 1$

-0.16	-0.58	-1.15	-0.11	1.46
-0.27	-1.27	-3.48	-0.65	0.00
-0.54	-3.36		-3.28	0.04
0.39	-0.83	-3.36	-1.27	-0.31
0.00	0.39	-0.54	-0.27	-0.16

$\tau = 2$

-1.09	-1.34	-1.61	-0.32	1.37
-1.39	-2.35	-4.35	-0.98	0.00
...	-1.80	-4.69		-3.98
-0.71	-2.14	-4.63	-2.15	-0.88
0.00	-0.69	-1.75	-1.26	-0.85

...

-1.11	-1.36	-1.62	-0.33	1.37
-1.42	-2.37	-4.37	-0.99	0.00
-1.83	-4.72		-3.99	-0.30
-0.73	-2.16	-4.65	-2.16	-0.89
0.00	-0.72	-1.77	-1.28	-0.87

$\tau = 29$

-1.11	-1.36	-1.62	-0.33	1.37
-1.42	-2.37	-4.37	-0.99	0.00
-1.83	-4.72		-3.99	-0.30
-0.73	-2.16	-4.65	-2.16	-0.89
0.00	-0.72	-1.77	-1.28	-0.87

$\tau = 30$

4.1 策略迭代 (19)

➤ 异步动态规划法 (Asynchronous Dynamic Programming, ADP)

采用异步计算方式时，每一轮迭代都直接用**新产生的值函数**来替换**旧的值函数**，不需要对上一轮迭代的状态值函数进行备份，既减少了迭代次数，又节省了存储空间。对于上述扫地机器人任务，采用异步计算方式进行评估，**30**轮迭代后既可以收敛到 v_{π} ，而采用同步计算方式，收敛到 v_{π} 则需要**51**轮迭代。另外，使用**异步计算方式**时，每次遍历并不需要对所有的状态值函数都做一次更新，而可以任意顺序更新状态值，这样其中的某些状态值可能会在其他状态值更新一次之前已经更新过多次。

4.1 策略迭代 (20)

➤ ADP的特点归纳

- ✓ ADP可以对更新顺序进行调整，通常重要的状态优先更新；
- ✓ 实际情况中，ADP必须保证完成所有状态的价值更新；
- ✓ ADP并不一定能减少计算量。但该方法的作用在于：
算法在改进策略之前不需要陷入无望的长时间扫描。


4.1 策略迭代 (21)

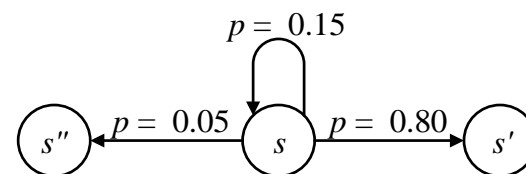
例4.2 对随机环境扫地机器人任务进行策略评估。

重新考虑图中描述的随机环境

MDP问题:

假设由于地面的问题，采取某一动作后，状态转换不再确定。当采取某一动作试图向某一方向移动时，机器人成功移动的概率为**0.80**，保持原地不动的概率为**0.15**，移动到相反方向的概率为**0.05**。

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
	1	2	3	4

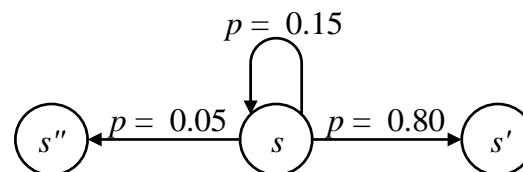


4.1 策略迭代 (22)

在随机环境下，状态空间、动作空间与确定环境是完全相同的，其随机性主要体现在状态转移函数和奖赏函数上。根据任务的随机性，状态转移只能用概率来表示。

➤ 状态转移函数

$$p(s, a, s') = \begin{cases} 0.80, & \text{如果 } s + a = s' \text{ 且 } s \neq s' \\ 0.15, & \text{如果 } s = s' \text{ 且 } s \neq [1,1] \text{ 且 } s \neq [5,4] \\ 0.05, & \text{如果 } s - a = s' \text{ 且 } s \neq s' \end{cases}$$



4.1 策略迭代 (23)

➤ 奖赏函数

在随机环境下，奖赏的获取不单纯受 (s, a) 的影响，还与下一状态 s' 相关。

$$r(s, a, s') = \begin{cases} +3, & \text{如果 } s \neq [5, 4] \text{ 且 } s' = [5, 4] \\ +1, & \text{如果 } s \neq [1, 1] \text{ 且 } s' = [1, 1] \\ -10, & \text{如果 } s + a = [3, 3] \text{ 且 } s = s' \\ 0, & \text{其他} \end{cases}$$

4.1 策略迭代 (24)

➤ 评估过程

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$\tau = 0$

0.00	-0.07	-0.48	-0.13	1.16
0.01	-0.35	-2.19	-0.11	0.00
0.06	-2.10		-2.19	0.37
0.28	0.10	-2.10	-0.35	-0.07
0.00	0.28	0.06	0.01	0.00

$\tau = 1$

$\tau = 1$:

$$V(S_{24}) = \frac{1}{2} * [0.8 * (r_{11} + \gamma V(S_{11})) + 0.15 * (r_{12} + \gamma V(S_{17})) + 0.05 * (r_{13} + \gamma V(S_{17}))]$$

$$+ \frac{1}{2} * [0.8 * (r_{21} + \gamma V(S_{17})) + 0.15 * (r_{12} + \gamma V(S_{17})) + 0.05 * (r_{13} + \gamma V(S_{22}))]$$

$$= \frac{1}{2} * [0.8 * (3 + 0.8 * 0) + 0.15 * (0 + 0.8 * 0) + 0.05 * (1 + 0.8 * 0)]$$

$$+ \frac{1}{2} * [0.8 * (0 + 0.8 * 0) + 0.15 * (0 + 0.8 * 0) + 0.05 * (0 + 0.8 * 0)]$$

$$\approx 1.16$$

$$V(S_{17}) = \frac{1}{4} * [0.8 * (r_{11} + \gamma V(S_{22})) + 0.15 * (r_{12} + \gamma V(S_{17})) + 0.05 * (r_{13} + \gamma V(S_{17}))]$$

$$+ \frac{1}{4} * [0.8 * (r_{21} + \gamma V(S_{17})) + 0.15 * (r_{12} + \gamma V(S_{17})) + 0.05 * (r_{13} + \gamma V(S_{22}))]$$

$$+ \frac{1}{4} * [0.8 * (r_{31} + \gamma V(S_5)) + 0.15 * (r_{32} + \gamma V(S_5)) + 0.05 * (r_{23} + \gamma V(S_5))]$$

$$+ \frac{1}{4} * [0.8 * (r_{11} + \gamma V(S_{10})) + 0.15 * (r_{12} + \gamma V(S_5)) + 0.05 * (r_{13} + \gamma V(S_{10}))]$$

$$+ \frac{1}{4} * [0.8 * (r_{21} + \gamma V(S_0)) + 0.15 * (r_{12} + \gamma V(S_5)) + 0.05 * (r_{13} + \gamma V(S_{10}))]$$

$$+ \frac{1}{4} * [0.8 * (r_{31} + \gamma V(S_6)) + 0.15 * (r_{32} + \gamma V(S_5)) + 0.05 * (r_{23} + \gamma V(S_5))]$$

$$= \frac{1}{3} * [0.8 * (0 + 0.8 * 0) + 0.15 * (0 + 0.8 * 0) + 0.05 * (1 + 0.8 * 0)]$$

$$+ \frac{1}{3} * [0.8 * (1 + 0.8 * 0) + 0.15 * (0 + 0.8 * 0) + 0.05 * (0 + 0.8 * 0)]$$

$$+ \frac{1}{3} * [0.8 * (0 + 0.8 * 0) + 0.15 * (0 + 0.8 * 0) + 0.05 * (0 + 0.8 * 0)]$$

$$\approx 0.28$$

4.1 策略迭代 (25)

➤ 评估过程

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

0.00	-0.07	-0.48	-0.13	1.16
0.01	-0.35	-2.19	-0.11	0.00
0.06	-2.10		-2.19	0.37
0.28	0.10	-2.10	-0.35	-0.07
0.00	0.28	0.06	0.01	0.00

-0.07	-0.34	-0.81	-0.23	1.38
-0.15	-0.95	-3.00	-0.39	0.00
-0.35	-2.93		-2.85	0.26
0.36	-0.58		-0.95	-0.15
0.00	0.36	-0.35	-0.15	-0.07

$\tau = 2$:

$\tau = 1$

$\tau = 2$

$$\begin{aligned}
 V(S_{17}) &= \frac{1}{4} * [0.8 * (r_{11} + \gamma V(S_{22})) + 0.15 * (r_{12} + \gamma V(S_{17})) + 0.05 * (r_{13} + \gamma V(S_{17}))] \\
 &\quad + \frac{1}{4} * [0.8 * (r_{21} + \gamma V(S_{23})) + 0.15 * (r_{22} + \gamma V(S_{24})) + 0.05 * (r_{23} + \gamma V(S_{24}))] \\
 V(S_{24}) &= \frac{1}{2} * [0.8 * (r_{11} + \gamma V(S_{19})) + 0.15 * (r_{12} + \gamma V(S_{24})) + 0.05 * (r_{13} + \gamma V(S_{24}))] \\
 &\quad + \frac{1}{2} * [0.8 * (r_{21} + \gamma V(S_{23})) + 0.15 * (r_{22} + \gamma V(S_{24})) + 0.05 * (r_{23} + \gamma V(S_{24}))] \\
 &= \frac{1}{4} * [0.8 * (3 + 0.8 * 0) + 0.15 * (0 + 0.8 * 1.158) + 0.05 * (0 + 0.8 * 1.158)] \\
 &\quad + \frac{1}{4} * [0.8 * (0 + 0.8 * (-0.023)) + 0.15 * (0 + 0.8 * 1.158) + 0.05 * (0 + 0.8 * 1.158)] \\
 &\quad + \frac{1}{4} * [0.8 * (0 + 0.8 * (-0.106)) + 0.15 * (0 + 0.8 * (-2.185)) + 0.05 * (0 + 0.8 * (-0.951))] \\
 &\approx -3.00
 \end{aligned}$$

4.1 策略迭代 (26)

➤ 评估过程

- ✓ 当 $\tau = 34$ 时, $|V_\tau(s) - V_{\tau-1}(s)|_\infty < \theta$, $V_\tau(s)$ 认为已经**收敛**于 $v_\pi(s)$, 计算得到的 $v_\pi(s)$ 就是在策略 π 下的有效评估。

$$\pi(a|s) = 1/|\mathcal{A}(s)|:$$

-0.80	-1.02	-1.27	0.16	1.37
-1.07	-1.96	-3.89	-0.75	0.00
-1.43	-4.18		-3.56	-0.06
-0.49	-1.79	-4.13	-1.79	-0.61
0.00	-0.48	-1.39	-0.96	-0.60

$$\tau = 34$$

4.1 策略迭代 (27)

➤ 每轮状态值函数更新过程

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$\tau = 0$

0.00	-0.07	-0.48	-0.13	1.16
0.01	-0.35	-2.19	-0.11	0.00
0.06	-2.10		-2.19	0.37
0.28	0.10	-2.10	-0.35	-0.07
0.00	0.28	0.06	0.01	0.00

$\tau = 1$

-0.07	-0.34	-0.81	-0.23	1.38
-0.15	-0.95	-3.00	-0.39	0.00
-0.35	-2.93		-2.85	0.26
0.36	-0.58	-2.93	-0.95	-0.15
0.00	0.36	-0.35	-0.15	-0.07

$\tau = 2$

...	-0.76	-1.00	-1.25	-0.15	1.37
	-1.03	-1.93	-3.87	-0.73	0.00
...	-1.40	-4.14		-3.55	-0.5
	-0.46	-1.76	-4.09	-1.76	-0.59
	0.00	-0.45	-1.36	-0.93	-0.57

$\tau = 10$

...	-0.80	-1.02	-1.27	0.16	1.37
	-1.07	-1.96	-3.89	-0.75	0.00
...	-1.43	-4.18		-3.56	-0.06
	-0.49	-1.79	-4.13	-1.79	-0.61
	0.00	-0.48	-1.39	-0.96	-0.60

$\tau = 33$

	-0.80	-1.02	-1.27	0.16	1.37
	-1.07	-1.96	-3.89	-0.75	0.00
	-1.43	-4.18		-3.56	-0.06
	-0.49	-1.79	-4.13	-1.79	-0.61
	0.00	-0.48	-1.39	-0.96	-0.60

$\tau = 34$

4.1 策略迭代 (28)

➤ 基于动作值函数的策略评估

基于动作值函数的策略评估迭代式为：

$$\begin{aligned} q_{\tau}(s, a) &= \mathbb{E}_{\pi}(G_t \mid S_t = s, A_t = a) \\ &= \sum_{s', r} p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s') q_{\tau-1}(s', a') \right] \end{aligned}$$

4.1 策略迭代 (29)

算法 4.2 基于动作值函数的策略评估算法

输 入	初始策略 $\pi(a s)$, 动态性 p , 奖赏函数 r , 折扣因子 γ
初始化	1. 对任意 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$, 初始化动作值函数, 如 $Q(s,a) = 0$ 2. 阈值 θ 设置为一个较小的实数值, 如 $\theta = 0.01$
策 略 评 估	3. repeat 对每一轮策略评估 $\tau = 1, 2, \dots$ 4. $\delta \leftarrow 0$ 5. for 每个状态-动作对 (s,a) do 7. $q \leftarrow Q(s,a)$ 8. $Q(s,a) \leftarrow \sum_{s',r} p(s',r s,a) \left[r + \gamma \sum_{a'} (\pi(a' s') Q(s',a')) \right]$ 9. $\delta \leftarrow \max(\delta, q - Q(s,a))$ 10. end for 11. until $\delta < \theta$
输 出	$q_* = Q$

4.1 策略迭代 (30)

例4.3 基于Q值函数对确定环境扫地机器人任务进行策略评估

- ✓ 整体状态与动作条件与例4.1保持一致。
- ✓ 动作顺序: *Up, Down, Left, Right*
- ✓ 折扣系数: $\gamma = 0.8$

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
	1	2	3	4

4.1 策略迭代 (31)

Q_τ \ 状态	...	S_2	...	S_6	S_7	S_8	...	S_{12}	...
$\tau = 0$...	0.000 0.000; 0.000 *.***	...	0.000 0.000; 0.000 *.***	0.000 0.000; 0.000 *.***	0.000 0.000; 0.000 *.***
$\tau = 1$...	0.000 0.267; 0.000 *.***	...	0.000 0.267; 0.000 0.267	-10.0 0.107; 0.000 0.071	0.000 -1.964; 0.000 0.0189

$$Q(S_7, Up) = r + \gamma \sum_{a'} (\pi(a' | S_7) Q(S_7, a'))$$

$\tau = 1$:

$$\begin{aligned}
 &= -10 + 0.8 * \left[\frac{1}{4} * Q(S_7, Up) + \frac{1}{4} * Q(S_7, Down) + \frac{1}{4} * Q(S_7, Left) + \frac{1}{4} * Q(S_7, Right) \right] \\
 &= -10 + 0.8 * (0.25 * 0 + 0.25 * 0 + 0.25 * 0 + 0.25 * 0) \\
 &= -10.00
 \end{aligned}$$

$$Q(S_7, Down) = r + \gamma \sum_{a'} (\pi(a' | S_2) Q(S_2, a'))$$

$$\begin{aligned}
 &= 0 + 0.8 * \left[\frac{1}{3} * Q(S_2, up) + \frac{1}{3} * Q(S_2, Left) + \frac{1}{3} * Q(S_2, Right) \right] \\
 &= 0 + 0.8 * \left(\frac{1}{3} * 0 + \frac{1}{3} * 0.267 + \frac{1}{3} * 0 \right) \\
 &\approx 0.071
 \end{aligned}$$

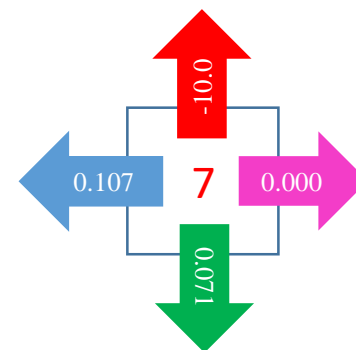
$$Q(S_7, Left) = r + \gamma \sum_{a'} (\pi(a' | S_6) Q(S_6, a'))$$

$$\begin{aligned}
 &= 0 + 0.8 * \left[\frac{1}{4} * Q(S_6, up) + \frac{1}{4} * Q(S_6, Down) + \frac{1}{4} * Q(S_6, Left) + \frac{1}{4} * Q(S_6, Right) \right] \\
 &= 0 + 0.8 * \left(\frac{1}{4} * 0 + \frac{1}{4} * 0.267 + \frac{1}{4} * 0.267 + \frac{1}{4} * 0 \right) \\
 &\approx 0.107
 \end{aligned}$$

$$Q(S_7, Right) = r + \gamma \sum_{a'} (\pi(a' | S_8) Q(S_8, a'))$$

$$\begin{aligned}
 &= 0 + 0.8 * \left[\frac{1}{4} * Q(S_8, Up) + \frac{1}{4} * Q(S_8, Down) + \frac{1}{4} * Q(S_8, Left) + \frac{1}{4} * Q(S_8, Right) \right] \\
 &= 0 + 0.8 * \left[\frac{1}{4} * 0 + \frac{1}{4} * 0 + \frac{1}{4} * 0 + \frac{1}{4} * 0 \right] \\
 &= 0.00
 \end{aligned}$$

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4



20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
	1	2	3	4

4.1 策略迭代 (32)

面向确定环境扫地机器人任务的动作值函数 q_π 评估过程

Q_τ \ 状态	...	S_2	...	S_6	S_7	S_8	...	S_{12}	...
$\tau = 0$...	0.000 0.000; 0.000 *.***	...	0.000 0.000; 0.000 *.***	0.000 0.000; 0.000 *.***	0.000 0.000; 0.000 *.***
$\tau = 1$...	0.000 0.267; 0.000 *.***	...	0.000 0.267; 0.000 0.267	-10.0 0.107; 0.000 0.071	0.000 -1.964; 0.000 0.0189
$\tau = 2$...	-1.964 0.314; 0.189 *.***	...	-1.957 0.314; -1.964 0.314		-2.093 -2.689; -0.102 -0.218
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots		\vdots
$\tau = 28$...	-3.719 -0.573; -0.102 *.***	...	-3.773 -0.585; -3.719 -0.573	-13.719 -1.729; -1.729 -1.417	-3.189 -3.719; -0.709 -1.024
$\tau = 29$...	-3.719 -0.573; -0.102 *.***	...	-3.773 -0.585; -3.719 -0.573	-13.719 -1.729; -1.729 -1.417	-3.189 -3.719; -0.709 -1.024
q_π	...	-3.719 -0.573; -0.102 *.***	...	-3.773 -0.585; -3.719 -0.573	-13.719 -1.729; -1.729 -1.417	-3.189 -3.719; -0.709 -1.024

结论

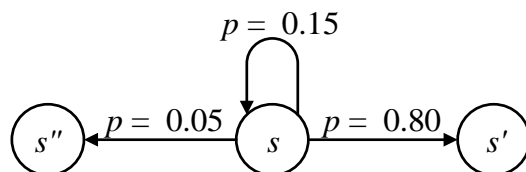
当 $\tau = 30$ 时, $|Q_\tau(s, a) - Q_{\tau-1}(s, a)|_\infty < \theta$, 认为 $Q_\tau(s, a)$ 已经收敛于 $q_\pi(s, a)$, 计算得到的 $q_\pi(s, a)$ 就是在策略 π 下的有效评估。

4.1 策略迭代 (33)

例4.4 基于Q值函数对随机环境扫地机器人任务进行策略评估

- ✓ 整体状态与动作条件与例4.1保持一致。
- ✓ 动作顺序: *Up, Down, Left, Right*
- ✓ 折扣系数: $\gamma = 0.8$
- ✓ 状态转移情况如图所示。

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
	1	2	3	4



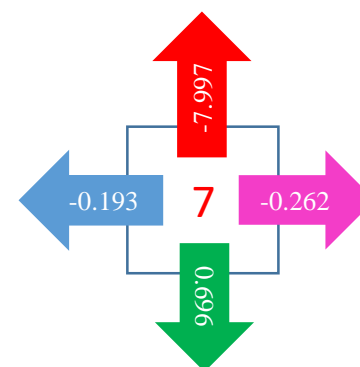
4.1 策略迭代 (34)

Q_τ \ 状态	...	S_2	...	S_6	S_7	S_8	...	S_{12}	...
$\tau = 0$...	0.000 0.000; 0.000 *.***	...	0.000 0.000; 0.000 *.***	0.000 0.000; 0.000 *.***	0.000 0.000; 0.000 *.***
$\tau = 1$...	0.000 0.188; 0.019 *.***	...	0.012 0.197; 0.024 0.189	-7.997 -0.193; -0.262 0.696	0.001 -1.463; 0.135

$\tau = 1$:

$$\begin{aligned}
 Q(S_7, Up) &= p(s', r | s, Up) \left[r + \gamma \sum_{a'} (\pi(a' | s') Q(s', a')) \right] \\
 &= p(S_7, r | S_7, Up) \left[r_1 + \gamma \sum_{a'} (\pi(a' | S_7) Q(S_7, a')) \right] \\
 &+ p(S_7, r_2 | S_7, Up) \left[r_2 + \gamma \sum_{a'} (\pi(a' | S_7) Q(S_7, a')) \right] \\
 &+ p(S_2, r_3 | S_7, Up) \left[r_3 + \gamma \sum_{a'} (\pi(a' | S_2) Q(S_2, a')) \right] \\
 &= 0.8 * \left[-10 + 0.8 * \left(\frac{1}{4} * Q(S_7, Up) + \frac{1}{4} * Q(S_7, Down) + \frac{1}{4} * Q(S_7, Left) + \frac{1}{4} * Q(S_7, Right) \right) \right] \\
 &+ 0.15 * \left[0 + 0.8 * \left(\frac{1}{4} * Q(S_7, Up) + \frac{1}{4} * Q(S_7, Down) + \frac{1}{4} * Q(S_7, Left) + \frac{1}{4} * Q(S_7, Right) \right) \right] \\
 &+ 0.05 * \left[0 + 0.8 * \left(\frac{1}{3} * Q(S_2, Up) + \frac{1}{3} * Q(S_2, Left) + \frac{1}{3} * Q(S_2, Right) \right) \right] \\
 &= 0.8 * \left[-10 + 0.8 * \left(\frac{1}{4} * 0.00 + \frac{1}{4} * 0.00 + \frac{1}{4} * 0.00 + \frac{1}{4} * 0.00 \right) \right] \\
 &+ 0.15 * \left[0 + 0.8 * \left(\frac{1}{4} * 0.00 + \frac{1}{4} * 0.00 + \frac{1}{4} * 0.00 + \frac{1}{4} * 0.00 \right) \right] \\
 &+ 0.05 * \left[0 + 0.8 * \left(\frac{1}{3} * 0.00 + \frac{1}{3} * 0.19 + \frac{1}{3} * 0.02 \right) \right] \\
 &= -7.997
 \end{aligned}$$

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4



4.1 策略迭代 (44)

➤ 评估过程

- ✓ 同理可以计算动作值函数 $Q(S_7, Down) = -0.696$,
 $Q(S_7, Left) = -0.193, Q(S_7, Right) = -0.262$ 。

按顺序计算完一轮后，得到动作值函数 $Q(s, a)$ 。

- ✓ 当 $\tau = 30$ 时, $|Q_\tau(s, a) - Q_{\tau-1}(s, a)|_\infty < \theta$ ，认为 $Q_\tau(s, a)$ 已经收敛于 $q_\pi(s, a)$ ，计算得到的 $q_\pi(s, a)$ 就是在策略 π 下的有效评估。

4.1 策略迭代 (45)

➤ 面向随机环境扫地机器人任务的动作值函数 q_π 评估过程

状态 q_τ	S_1	S_2	S_3	...	S_7	...	S_{24}
q_0	0.00;×;0.00;0.00	0.00;×;0.00;0.00	0.00;×;0.00;0.00	...	0.00;0.00;0.00;0.00	...	×;0.00;0.00;×
q_1	0.00;×;0.80;0.08	0.00;×;0.19;0.02	0.00;×;0.04;0.00	...	-8.00;-0.70;-0.19;-0.26	...	×;2.40;0.08;×
q_2	0.11;×;0.84;0.14	-1.45;×;0.18;-0.02	-0.25;×;-0.28;-0.04	...	-9.76;-1.10;-0.77;-0.64	...	×;2.60;0.18;×
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
q_{33}	-1.22;×;0.69;-0.90	-2.86;×;-0.51;-0.80	-1.30;×;-0.51;-0.80	...	-11.19;-1.89;-1.71;-1.71	...	×;2.62;0.19;×
q_{34}	-1.22;×;0.69;-0.90	-2.86;×;-0.51;-0.80	-1.30;×;-0.51;-0.80	...	-11.19;-1.89;-1.71;-1.71	...	×;2.62;0.19;×
q_π	-1.22;×;0.69;-0.90	-2.86;×;-0.51;-0.80	-1.30;×;-0.51;-0.80	...	-11.19;-1.89;-1.71;-1.71	...	×;2.62;0.19;×

4.1 策略迭代 (46)

➤ 策略改进

策略的优劣性可以由值函数来评价。通过策略评估迭代得到值函数，再利用动作值函数来寻找更好的策略。

假设已知某一策略 π 的值函数 v_π 或 q_π ，目的是寻找一个更优策略 π' 。

4.1 策略迭代 (47)

➤ 特殊情况到一般情况对策略改进方法

✓ 特殊情况

针对单一状态 s 和特定动作 a ，制定如下约定以获得新策略 $\pi' (\pi' \neq \pi)$:

- ◆ 在状态 s 下选择一个新动作 $a (a \neq \pi(s))$ ；
- ◆ 保持后续（其他）状态所执行的动作与原策略 π 给出的动作相同。

4.1 策略迭代 (48)

➤ 特殊情况到一般情况对策略改进方法

- ◆ 根据动作值函数贝尔曼方程，得到的价值为：

$$\begin{aligned} q_{\pi}(s, a) &= \mathbb{E}_{\pi} \left(R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s, A_t = a \right) \\ &= \sum_{s', r} p(s', r \mid s, a) [r + \gamma v_{\pi}(s')] \end{aligned}$$

- ◆ 若 $q_{\pi}(s, a) \geq v_{\pi}(s)$ 成立，则说明满足以上约定的策略 π' 优于或等价于 π 。

4.1 策略迭代 (49)

➤ 特殊情况到一般情况对策略改进方法

✓ 一般情况

将单一状态和特定动作的情况进行拓展。对任意状态 $s \in \mathcal{S}$ ，若存在任意的两个确定策略 π 和 π' 满足策略改进定理，则说明在状态 s 处采取策略 π' 时，能得到更大的值函数，即 π' 优于或等价于 π 。

策略改进定理为：

$$q_{\pi}(s, \pi'(s)) \geq v_{\pi}(s)$$

即：

$$v_{\pi'}(s) \geq v_{\pi}(s)$$

4.1 策略迭代 (50)

➤ 特殊情况到一般情况对策略改进方法

$$\begin{aligned} v_{\pi}(s) &\leq q_{\pi}(s, \pi'(s)) \\ &= \mathbb{E}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s, A_t = \pi'(s)] && \text{由式 (4.2) 推出} \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s] && \text{将条件 } \pi'(s) \text{ 提出} \\ &\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_{\pi}(S_{t+1}, \pi'(S_{t+1})) | S_t = s] && \text{由式 (4.4) 推出, } v_{\pi}(S_{t+1}) \leq q_{\pi}(S_{t+1}, \pi'(S_{t+1})) \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma \pi(S_{t+2})] | S_t = s] && \text{由式 (4.2) 推出} \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_{\pi}(S_{t+2}) | S_t = s] && \text{期望展开} \\ &\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_{\pi}(S_{t+3}) | S_t = s] \\ &\dots \\ &\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s] = \mathbb{E}_{\pi'}[G_t | S_t = s] = v_{\pi'}(s) \end{aligned}$$

4.1 策略迭代 (51)

➤ 确定贪心策略

$$\pi'(a | s) = \begin{cases} 1 & a = \arg \max_a q_{\pi}(s, a) \\ 0 & \text{其他} \end{cases}$$

➤ 随机贪心策略

$$\pi'(a | s) = \begin{cases} \pi'(a' | s) & \pi'(a' | s) \in (0, 1] \text{ 且 } \sum_{a'} \pi'(a' | s) = 1 \\ 0 & \text{其他} \end{cases}$$

4.1 策略迭代 (52)

➤ 4.1.2 策略迭代

策略迭代的关键部分是策略评估，首先评估状态的价值，然后根据状态的动作值进行相应的策略改进，并进行下一轮评估和改进，直到策略稳定。策略改进可以通过求解静态最优问题来实现，通过状态动作值来选择动作，通常比策略评估容易。

4.1 策略迭代 (53)

➤ 4.1.2.1 基于状态值函数的策略迭代

基于状态值函数的策略迭代算法主要包括以下3个阶段：

- (1) 初始化策略函数 $\pi(a | s)$ 和状态值函数 $V_0(s)$;
- (2) **策略评估**：在当前策略 π 下，使用贝尔曼方程更新状态值函数 $V_t(s)$ ，直到收敛于 $v_\pi(s)$ ，再计算出 $q_\pi(s, a)$ 。
- (3) **策略改进**：基于 $q_\pi(s, a)$ ，通过贪心策略得到更优策略。

4.1 策略迭代 (54)

算法 4.3 基于随机 MDP 的状态值函数 v_π 策略迭代算法

输 入		初始策略 $\pi(a s)$ ，动态性 p ，奖赏函数 r ，折扣因子 γ
初始化		<ol style="list-style-type: none"> 对任意 $s \in \mathcal{S}$，初始化状态值函数：如 $V(s) = 0$ 阈值 θ 设置为一个较小的实值
策略迭代		3. repeat 对每一轮策略迭代 $l = 1, 2, \dots$
	策略评估	4. 在当前策略 $\pi(a s)$ 下，通过算法 4.1 策略评估迭代，得到值函数 v_π ： $V(s)$
	策略改进	<ol style="list-style-type: none"> $policy_stable \leftarrow \mathbf{True}$ for 每个状态 s do $old_policy \leftarrow \pi(a s)$ $Q(s, a) = \sum_{s', r} p(s', r s, a) [r + \gamma V(s')]$ $max_action \leftarrow \arg \max_a Q(s, a) ; \quad max_count \leftarrow \text{count}(\max(Q(s, a)))$

4.1 策略迭代 (55)

	<div>10. for 每个状态-动作对 (s, a') do</div> <div>11. if $a' == \text{max_action}$ then</div> <div>12. $\pi(a' s) = 1 / (\text{max_count})$</div> <div>13. else</div> <div>14. $\pi(a' s) = 0$</div> <div>15. end if</div> <div>16. end for</div> <div>17. if $\text{old_policy} \neq \pi(a s)$ then</div> <div>18. $\text{policy_stable} \leftarrow \text{False}$</div> <div>19. end for</div>
	20. until $\text{policy_stable} = \text{True}$
输 出	$v_* = V$, $\pi_* = \pi$

4.1 策略迭代 (56)

➤ 例4.5 将基于状态值函数的策略迭代算法4.3应用于例4.1的确定环境扫地机器人任务

经过多轮迭代后，得到下表所示的**值函数**和**策略迭代**更新过程。

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
	1	2	3	4

4.1 策略迭代 (57)

表4.3 面向确定环境扫地机器人任务的状态值函数策略迭代更新过程

	S_1	S_2	S_3	...	S_7	...	S_{24}
v_0	0.00	0.00	0.00	...	0.00	...	0.00
π_0	0.33;0.00;0.33;0.33	0.33;0.00;0.33;0.33	0.33;0.00;0.33;0.33	...	0.25;0.25;0.25;0.25	...	0.00;0.5;0.5;0.00
v_1	-0.72	-1.77	-1.28	...	-4.65	...	1.37
q_1	-1.73;×;1.00;-1.42	-3.72;×;-0.57;-1.02	-1.73;×;-1.42;-0.69	...	-10.00;-1.42;-1.73;-1.73	...	×;3.00;-0.26;×
π_1	0.00;0.00;1.00;0.00	0.00;0.00;1.00;0.00	0.00;0.00;0.00;1.00	...	0.00;0.00;1.00;1.00	...	0.00;1.00;0.00;0.00
v_2	1.00	0.80	1.54	...	0.64	...	3.00
q_2	0.64;×;1.00;0.64	0.51;×;0.80;1.23	1.54;×;0.64;1.54	...	-10.00;0.64;0.64;1.54	...	×;3.00;1.92;×
π_2	0.00;0.00;1.00;0.00	0.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

4.1 策略迭代 (58)

表4.3 面向确定环境扫地机器人任务的状态值函数策略迭代更新过程 (续)

π_4	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	1.00;0.00;0.00;0.00	...	0.00;1.00;0.00;0.00
v_5	1.00	1.23	1.54	...	1.54	...	3.00
q_5	0.98;×;1.00;0.98	1.23;×;0.80;1.23	1.54;×;0.98;1.54	...	-10.00;0.98;0.98;1.54	...	×;3.00;1.92;×
π_5	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00
v_6	1.00	1.23	1.54	...	1.54	...	3.00
q_6	0.98;×;1.00;0.98	1.23;×;0.80;1.23	1.54;×;0.98;1.54	...	-10.00;0.98;0.98;1.54	...	×;3.00;1.92;×
π_6	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00
π_*	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00

4.1 策略迭代 (59)

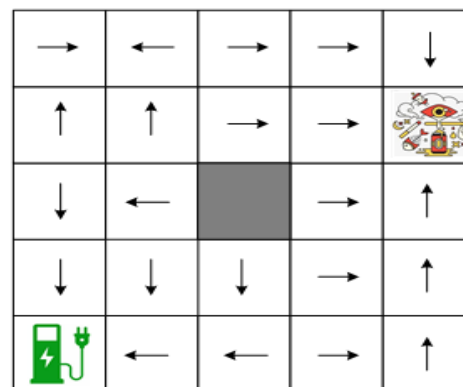
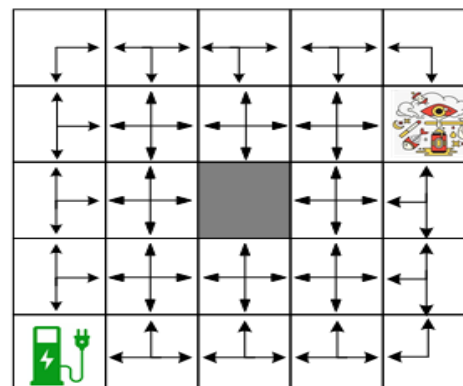
面向确定环境扫地机器人任务的状态值函数及策略迭代更新过程图

$l = 0 :$

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	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0

$l = 1 :$

-0.79	-0.96	-1.1	-0.02	1.49
-1.01	-1.7	-3.15	-0.46	0.0
-1.29	-3.4		-2.88	0.09
0.41	-1.51	-3.36	-1.53	-0.54
0.0	-0.4	-1.24	-0.89	-0.57



4.1 策略迭代 (60)

面向确定环境扫地机器人任务的状态值函数及策略迭代更新过程图（续）

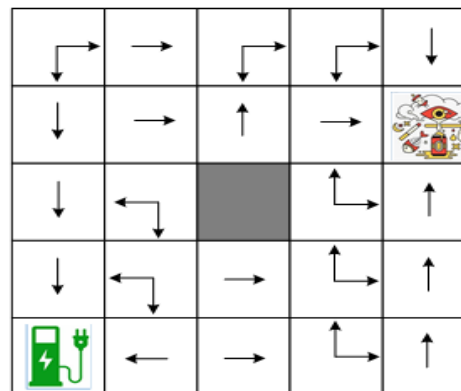
$l = 2 :$

0.0	0.0	1.92	2.4	3.0
0.0	0.0	2.4	3.0	0.0
0.8	0.64		2.4	3.0
1.0	0.8	0.64	1.92	2.4
0.0	1.0	0.8	1.54	1.92

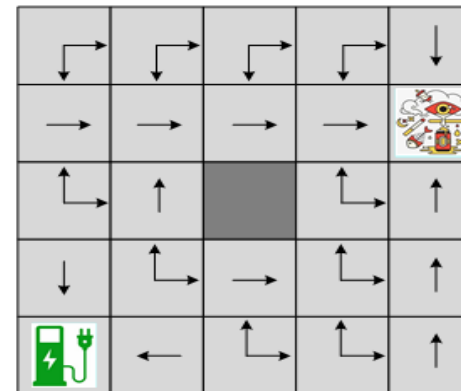
\vdots
 \vdots

$l = 5 :$

1.23	1.54	1.92	2.4	3.0
1.54	1.92	2.4	3.0	0.0
1.23	1.54		2.4	3.0
1.0	1.23	1.54	1.92	2.4
0.0	1.0	1.23	1.54	1.92



\vdots
 \vdots



4.1 策略迭代（61）

➤ 例4.6 汽车租赁问题。

汽车租赁场（A，B租赁场）：

- ✓ 每租出一辆车，获得10美元的租金；
- ✓ 两租赁场移动车辆费用2美元；
- ✓ 假设每个租赁场租车和还车的数量是一个泊松随机量：
即：期望数量 n 的概率为： $\frac{\lambda^n}{n!} e^{-\lambda}$
- ✓ 任何一个租赁场车辆总数不超过20辆车；
- ✓ 当天还回的车辆第2天才能出租。
- ✓ 两个租车场之间每天最多可移车数量为5辆。

4. 策略迭代 (62)

➤ 汽车租赁任务的MDP数学建模如下:

➤ 状态空间:

状态共 $21 \times 21 = 441$ 个, 即:

$$S = \{[0, 0], [0, 1], [0, 2], \dots, [10, 10], [10, 11], \dots, [20, 19], [20, 20]\}$$

➤ 动作空间:

可移动车辆数目不超过5辆。设A租赁场向B租赁场移车为“-”，B租赁场向A租赁场移车为“+”。离散化为11个不同的动作。即:

$$A(s) = \{-5, -4, -3, -2, -1, 0, +1, +2, +3, +4, +5\}$$

4.1 策略迭代 (63)

➤ 状态转移函数

✓ 状态转移函数 $p([2, 5], +1, [1, 2])$ 计算为:

$$\begin{aligned} p([2, 5], +1, [1, 2]) = & \left(\frac{\lambda_{A1}^2}{2!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A1}^0}{0!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{B1}^2}{2!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B1}^0}{0!} e^{-\lambda_{B1}} \right) \\ & + \left(\frac{\lambda_{A1}^2}{2!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A1}^0}{0!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{B1}^3}{3!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B1}^1}{1!} e^{-\lambda_{B1}} \right) \\ & + \left(\frac{\lambda_{A1}^2}{2!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A1}^0}{0!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{B1}^4}{4!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B1}^2}{2!} e^{-\lambda_{B1}} \right) \\ & + \left(\frac{\lambda_{A1}^3}{3!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A1}^1}{1!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{B1}^2}{2!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B1}^0}{0!} e^{-\lambda_{B1}} \right) \\ & + \left(\frac{\lambda_{A1}^3}{3!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A1}^1}{1!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{B1}^3}{3!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B1}^1}{1!} e^{-\lambda_{B1}} \right) \\ & + \left(\frac{\lambda_{A1}^3}{3!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A1}^1}{1!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{B1}^4}{4!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B1}^2}{2!} e^{-\lambda_{B1}} \right) \end{aligned}$$

4.1 策略迭代 (64)

➤ 状态转移函数

✓ 如果 $0 \leq S'_A - S_A - a + n_A \leq 20$ 且 $0 \leq S'_B - S_B + a + n_B \leq 20$,
则有:

$$\begin{aligned} p(s, a, s') &= p([s_A, s_B], a, [s'_A, s'_B]) \\ &= \sum_{n_A=0}^{s_A+a} \sum_{n_B=0}^{S_B-a} \left(\frac{\lambda_{A1}^{n_A}}{n_A!} e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A2}^{(S'_A-S_A-a+n_A)}}{(S'_A-S_A-a+n_A)!} e^{-\lambda_{A2}} \right) * \left(\frac{\lambda_{B1}^{n_B}}{n_B!} e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B2}^{(S'_B-S_B+a+n_B)}}{(S'_B-S_B+a+n_B)!} e^{-\lambda_{B2}} \right) \end{aligned}$$

4.1 策略迭代 (65)

➤ 奖赏函数

该问题的立即奖赏函数为： $r([s_A, s_B], a, [s'_A, s'_B])$ ，即在当前状态 $s = [s_A, s_B]$ 下，采取动作 a ，到达下一状态 $s' = [s'_A, s'_B]$ 得到的立即奖赏。

✓ 两个租赁场的租车收益：

$$r_2 = \sum_{n_A=0}^{S_A+a} \sum_{n_B=0}^{S_B-a} \left[\left(\frac{\lambda_{A1}^{n_A}}{n_A!} * e^{-\lambda_{A1}} \right) * \left(\frac{\lambda_{A2}^{(S'_A - S_A - a + n_A)}}{(S'_A - S_A - a + n_A)!} * e^{-\lambda_{A2}} \right) * \left(\frac{\lambda_{B1}^{n_B}}{n_B!} * e^{-\lambda_{B1}} \right) * \left(\frac{\lambda_{B2}^{(S'_B - S_B + a + n_B)}}{(S'_B - S_B + a + n_B)!} * e^{-\lambda_{B2}} \right) * (n_A + n_B) * 10 \right]$$

4.1 策略迭代 (66)

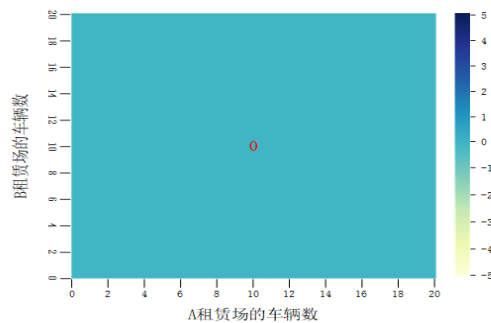
➤ 奖赏函数

✓ 两个租赁场之间的移车费用： $r_1 = -2 * |a|$ ；

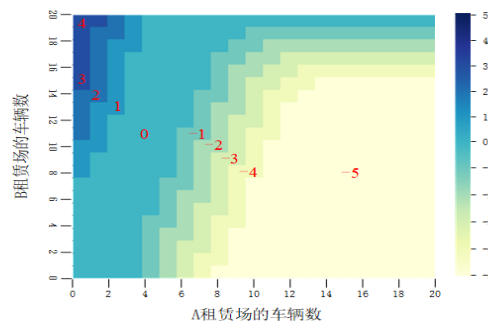
获得的立即奖为： $r([s_A, s_B], a, [s'_A, s'_B]) = r_1 + r_2$

4.1 策略迭代 (67)

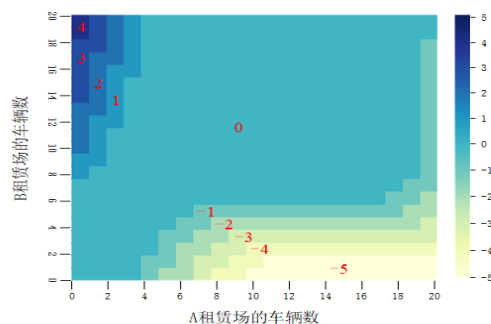
关于汽车租赁问题的策略迭代过程



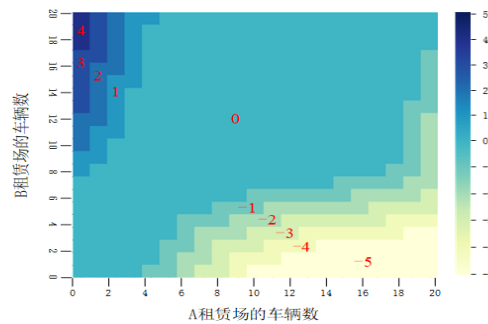
(1) 策略 π_0



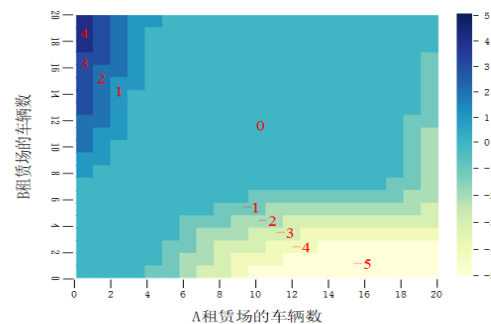
(2) 策略 π_1



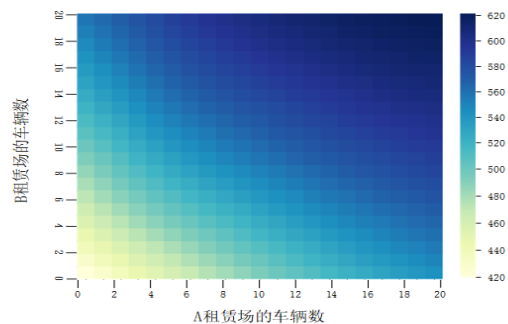
(3) 策略 π_2



(4) 策略 π_3



(5) 策略 π_4



(6) 最优价值

4.1 策略迭代 (68)

➤ 4.1.2.2 基于动作值函数的策略迭代

算法 4.4 基于随机 MDP 的动作值函数 q_π 策略迭代算法

输 入	初始策略 $\pi(a s)$, 动态性 p , 奖赏函数 r , 折扣因子 γ
初始化	1. 对任意 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$, 初始化动作值函数: 如 $Q(s,a) = 0$ 2. 阈值 θ 设置为一个较小的实值
	3. repeat 对每一轮策略迭代 $l = 1, 2, \dots$
策略 评 估	4. 在当前策略 $\pi(a s)$ 下, 通过算法 4.2 策略评估迭代, 得到动作值函数 q_π : $Q(s,a)$

4.1 策略迭代 (69)

策略迭代	策略改进	5. $policy_stable \leftarrow \mathbf{True}$
		6. for 每个状态 s do
		7. $old_policy \leftarrow \pi(a s)$
		8. $max_action \leftarrow \arg \max_a Q(s,a) ; \quad max_count \leftarrow \text{count}(\max_a(Q(s,a)))$
		9. for 每个状态-动作对 (s,a') do
		10. if $a' == max_action$ then
		11. $\pi(a' s) = 1 / (max_count)$
		12. else
		13. $\pi(a' s) = 0$
		14. end if
		15. end for
		16. if $old_policy \neq \pi(a s)$ then
		17. $policy_stable \leftarrow \mathbf{False}$
		18. end for
		19. until $policy_stable = \mathbf{True}$
	输出	$q_* = Q, \quad \pi_* = \pi$

4.1 策略迭代 (70)

例4.7 将基于动作值函数的策略迭代算法4.4应用于例4.1的确定环境扫地机器人任务。

表4.4 面向确定环境扫地机器人任务的基于动作值函数的策略迭代更新过程

	S_1	S_2	S_3	...	S_7	...	S_{24}
q_0	0.00;×;0.00;0.00	0.00;×;0.00;0.00	0.00;×;0.00;0.00	...	0.00;0.00;0.00;0.00	...	×;0.00;0.00;×
π_0	0.33;0.00;0.33;0.33	0.33;0.00;0.33;0.33	0.33;0.00;0.33;0.33	...	0.25;0.25;0.25;0.25	...	0.00;0.5;0.5;0.00
q_1	-1.73;×;1.00;-1.42	-3.72;×;-0.57;-1.02	-1.73;×;-1.42;-0.69	...	-10.00;-1.42;-1.73;-1.73	...	×;3.00;-0.26;×
π_1	0.00;0.00;1.00;0.00	0.00;0.00;1.00;0.00	0.00;0.00;0.00;1.00	...	0.00;0.00;1.00;1.00	...	0.00;1.00;0.00;0.00
q_2	0.64;×;1.00;0.64	0.51;×;0.80;1.23	1.54;×;0.64;1.54	...	-10.00;0.64;0.64;1.54	...	×;3.00;1.92;×
π_2	0.00;0.00;1.00;0.00	0.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00

4.1 策略迭代 (71)

表4.4 面向确定环境扫地机器人任务的基于动作值函数的策略迭代更新过程（续）

\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
π_4	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	1.00;0.00;0.00;0.00	...	0.00;1.00;0.00;0.00
q_5	0.98; \times ;1.00;0.98	1.23; \times ;0.80;1.23	1.54; \times ;0.98;1.54	...	-10.00;0.98;0.98;1.54	...	\times ;3.00;1.92; \times
π_5	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00
q_6	0.98; \times ;1.00;0.98	1.23; \times ;0.80;1.23	1.54; \times ;0.98;1.54	...	-10.00;0.98;0.98;1.54	...	\times ;3.00;1.92; \times
π_6	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00
π_*	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00

目 录

4

前言

4.1

策略迭代

4.2

值迭代

4.3

广义策略迭代

4.4

小结

4.2 值迭代 (1)

- 在策略迭代中，每轮策略改进之前都涉及策略评估，每次策略评估都需要多次遍历才能保证状态值函数在一定程度上得到收敛，这将消耗大量的时间和计算资源。

✓ 值迭代公式

$$\begin{aligned} v_{\ell}(s) &= \max_a \mathbb{E}_{\pi} (R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s, A_t = a) \\ &= \max_a \sum_{s', r} p(s', r | s, a) [r + \gamma v_{\ell-1}(s')] \end{aligned}$$

4.2 值迭代 (2)

- 根据迭代次数与策略稳定的相互关系，考虑在单步评估之后就进入改进过程，即采取截断式策略评估，在一次遍历完所有的状态后立即停止策略评估，进行策略改进，这种方法称为值迭代。

4.1 值迭代 (3)

➤ 在有穷状态空间MDP中，基于状态值函数的值迭代算法。

算法 4.5 基于状态值函数的值迭代算法

输 入	动态性 p ，奖赏函数 r ，折扣因子 γ
初始化	1. 对任意 $s \in \mathcal{S}$ ，初始化状态值函数，如 $V(s) = 0$ 2. 阈值 θ 设置为一个较小的实值
评 估 过 程	3. repeat 对每一轮值迭代 $l = 1, 2, \dots$ 4. $\delta \leftarrow 0$ 5. for 每个状态 s do 6. $v \leftarrow V(s)$ 7. $V(s) \leftarrow \max_a \sum_{s', r} p(s', r s, a) [r + \gamma V(s')]$ 8. $\delta \leftarrow \max(\delta, v - V(s))$ 9. end for 10. until $\delta < \theta$

4.2 值迭代 (4)

➤ 在有穷状态空间MDP中，基于状态值函数的值迭代算法（续）

最 优 策 略	<pre>11. for 每个状态 s do 12. $max_action \leftarrow \arg \max_a \sum_{s',r} p(s',r s,a)[r + \gamma V(s')]$; 13. $max_count \leftarrow \text{count}(max_action)$ 14. for 每个状态-动作对 (s,a') do 15. if $a' == max_action$ then 16. $\pi(a' s) = 1 / (max_count)$ 17. else 18. $\pi(a' s) = 0$ 19. end if 20. end for 21. end for</pre>
输 出	$v_* = V, \pi_* = \pi$

4.2 值迭代 (5)

➤ 在有穷状态空间MDP中，基于动作值函数的值迭代算法。

算法 4.6 基于动作值函数的值迭代算法

输 入	动态性 p ，奖赏函数 r ，折扣因子 γ
初始化	1. 对任意 $s \in \mathcal{S}$ ， $a \in \mathcal{A}(s)$ ，初始化动作值函数，如 $Q(s, a) = 0$ 2. 阈值 θ 设置为一个较小的实值
评 估 过 程	3. repeat 对每一轮值迭代 $l = 1, 2, \dots$ 4. $\delta \leftarrow 0$ 5. for 每个状态-动作对 (s, a) do 6. $q \leftarrow Q(s, a)$ 7. $Q(s, a) = \sum_{s', r} p(s', r s, a) \left[r + \gamma \max_{a'} Q(s', a') \right]$ 8. $\delta \leftarrow \max(\delta, q - Q(s, a))$ 9. until $\Delta < \theta$

4.2 值迭代 (6)

➤ 在有穷状态空间MDP中，基于动作值函数的值迭代算法（续）

最 优 策 略	<pre>10. for 每个状态 s do 11. $max_action \leftarrow \arg \max_a Q(s, a) ; \ max_count \leftarrow \text{count}(\max(Q(s, a)))$ 12. for 每个状态-动作对 (s, a') do 13. if $a' == max_action$ then 14. $\pi(a' s) = 1. / (max_count)$ 15. else 16. $\pi(a' s) = 0$ 17. end if 18. end for 19. end for</pre>
输 出	$q_* = Q, \ \pi_* = \pi$

4.2 值迭代 (7)

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$l = 0$

0.51	0.41	0.33	2.40	3.00
0.64	0.51	0.41	3.00	0.00
0.80	0.64		0.41	3.00
1.00	0.80	0.64	0.51	0.41
0.00	1.00	0.80	0.64	0.51

$l = 1$

➤ 例4.8 将基于状态值函数的值迭代算法4.5应用于确定环境扫地机器人任务。

- ✓ 当 $l = 0$ 时（ l 为值迭代次数），对于所有的 s 初始化为 $V(s) = 0$ ；
- ✓ 当 $l = 1$ 时，以状态 S_{24} 为例。在策略 π 下，只能采取**向下**和**向左**2个动作，概率各为0.5。采取向下的动作时，到达状态 S_{19} ($V(S_{19}) = 0$)，并可以捡到垃圾，获得 $r_1 = +3$ 的奖赏；采取向左的动作时，到达状态 S_{23} ($V(S_{23}) = 2.40$)，获得 $r_2 = 0$ 的奖赏。

4.2 值迭代 (8)

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$l = 0$

0.51	0.41	0.33	2.40	3.00
0.64	0.51	0.41	3.00	0.00
0.80	0.64		0.41	3.00
1.00	0.80	0.64	0.51	0.41
0.00	1.00	0.80	0.64	0.51

$l = 1$

$$\begin{aligned}
 V(S_{24}) &= \max(r_1 + \gamma V(S_{19}), r_2 + \gamma V(S_{23})) \\
 &= \max(3 + 0.8 * 0, 0 + 0.8 * 2.40) \\
 &= \max(3, 1.92) \\
 &= 3.00
 \end{aligned}$$

$$\begin{aligned}
 V(S_5) &= \max(r_1 + \gamma V(S_{10}), r_2 + \gamma V(S_0), r_3 + \gamma V(S_6)) \\
 &= \max(0 + 0.8 * 0, 1 + 0.8 * 0, 0 + 0.8 * 0) \\
 &= 1.00
 \end{aligned}$$

$$\begin{aligned}
 V(S_6) &= \max(r_1 + \gamma V(S_{11}), r_2 + \gamma V(S_1), r_3 + \gamma V(S_5), r_4 + \gamma V(S_7)) \\
 &= \max(0 + 0.8 * 0, 0 + 0.8 * 1.00, 0 + 0.8 * 1.00, 0 + 0.8 * 0) \\
 &= 0.80
 \end{aligned}$$

4.2 值迭代 (9)

- 当 $l = 2$ 时，以状态 S_{22} 、 S_{23} 、 S_{24} 为例，计算状态值函数。异步计算方式，通常与迭代的计算顺序有关，根据例4.1规定，在每一轮次中，这3个状态的计算顺序为 S_{22} 、 S_{23} 、 S_{24} 。

0.51	0.41	0.33	2.40	3.00
0.64	0.51	0.41	3.00	0.00
0.80	0.64		0.41	3.00
1.00	0.80	0.64	0.51	0.41
0.00	1.00	0.80	0.64	0.51

$l = 1$

0.51	0.41	1.92	2.40	3.00
0.64	0.51	2.40	3.00	0.00
0.80	0.64		2.40	3.00
1.00	0.80	0.64	0.51	2.40
0.00	1.00	0.80	0.64	0.51

$l = 2$

4.3. 值迭代 (10)

0.51	0.41	0.33	2.40	3.00
0.64	0.51	0.41	3.00	0.00
0.80	0.64		0.41	3.00
1.00	0.80	0.64	0.51	0.41
0.00	1.00	0.80	0.64	0.51

$l=1$

0.51	0.41	1.92	2.40	3.00
0.64	0.51	2.40	3.00	0.00
0.80	0.64		2.40	3.00
1.00	0.80	0.64	0.51	2.40
0.00	1.00	0.80	0.64	0.51

$l=2$

$$\begin{aligned}
 V(S_{22}) &= \max(r_1 + \gamma V(S_{17}), r_2 + \gamma V(S_{21}), r_3 + \gamma V(S_{23})) \\
 &= \max(0 + 0.8 * 2.40, 0 + 0.8 * 0.41, 0 + 0.8 * 2.40) \\
 &= \max(1.92, 0.33, 1.92) \\
 &= 1.92
 \end{aligned}$$

$$\begin{aligned}
 V(S_{23}) &= \max(r_1 + \gamma V(S_{18}), r_2 + \gamma V(S_{22}), r_3 + \gamma V(S_{24})) \\
 &= \max(0 + 0.8 * 3.00, 0 + 0.8 * 1.92, 0 + 0.8 * 3.00) \\
 &= \max(2.40, 1.54, 2.40) \\
 &= 2.40
 \end{aligned}$$

$$\begin{aligned}
 V(S_{24}) &= \max(r_1 + \gamma V(S_{19}), r_2 + \gamma V(S_{23})) \\
 &= \max(3 + 0.8 * 0.00, 0 + 0.8 * 2.40) \\
 &= 3.00
 \end{aligned}$$

4.2 值迭代 (11)

- 当 $l = 6$ 时, $|v_l(s) - v_{l-1}(s)|_\infty < \theta$, 认为 $v_l(s)$ 已经收敛于 $v_*(s)$, 计算得到的 $v_*(s)$ 就是最优状态值函数。

0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00		0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

$l = 0$

0.51	0.41	0.33	2.40	3.00
0.64	0.51	0.41	3.00	0.00
0.80	0.64		0.41	3.00
1.00	0.80	0.64	0.51	0.41
0.00	1.00	0.80	0.64	0.51

$l = 1$

0.51	0.41	1.92	2.40	3.00
0.64	0.51	2.40	3.00	0.00
0.80	0.64		2.40	3.00
1.00	0.80	0.64	0.51	2.40
0.00	1.00	0.80	0.64	0.51

$l = 2$

0.51	1.54	1.92	2.40	3.00
0.64	1.92	2.40	3.00	0.00
0.80	0.64		2.40	3.00
1.00	0.80	0.64	1.92	2.40
0.00	1.00	0.80	0.64	1.92

...

1.23	1.54	1.92	2.40	3.00
1.54	1.92	2.40	3.00	0.00
1.23	1.54		2.40	3.00
1.00	1.23	1.54	1.92	2.40
0.00	1.00	1.23	1.54	1.92

$l = 5$

1.23	1.54	1.92	2.40	3.00
1.54	1.92	2.40	3.00	0.00
1.23	1.54		2.40	3.00
1.00	1.23	1.54	1.92	2.40
0.00	1.00	1.23	1.54	1.92

$l = 6$

4.2 值迭代 (12)

表4.5 确定环境扫地机器人任务的基于状态值函数的值迭代更新过程

v	S_1	S_2	S_3	...	S_7	...	S_{24}
v_0	0.00	0.00	0.00	...	0.00	...	0.00
v_1	1.00	0.80	0.64	...	0.64		3.00
v_2	1.00	0.80	0.64	...	0.64		3.00
v_3	1.00	0.80	0.64	...	0.64		3.00
v_4	1.00	0.80	1.54		1.54		3.00
v_5	1.00	1.23	1.54		1.54		3.00
v_6	1.00	1.23	1.54		1.54		3.00
v_*	1.00	1.23	1.54	...	1.54	...	3.00
q_*	0.98; ×; 1.00; 0.98	1.23; ×; 0.80; 1.23	1.54; ×; 0.98; 1.54	...	-10.00; 0.98; 0.98; 1.54	...	×; 3.00; 1.92; 0.00
π_*	0.00; 0.00; 1.00; 0.00	1.00; 0.00; 0.00; 1.00	1.00; 0.00; 0.00; 1.00	...	0.00; 0.00; 0.00; 1.00	...	0.00; 1.00; 0.00; 0.00

4.2 值迭代 (13)

表4.6 确定扫地机器人任务的基于动作值函数的值迭代更新过程

q	S_1	S_2	S_3	...	S_7	...	S_{24}
q_0	0.00;×;0.00;0.00	0.00;×;0.00;0.00	0.00;×;0.00;0.00	...	0.00;0.00;0.00;0.00	...	×;0.00;0.00;×
q_1	0.00;×;1.00;0.00	0.00;×;0.80;0.00	0.00;×;0.64;0.00	...	-10.00;0.64;0.64;0.00	...	×;3.00;1.92;×
q_2	0.64;×;1.00;0.64	0.51;×;0.80;0.51	0.41;×;0.64;0.41	...	-10.00;0.64;0.64;0.41	...	×;3.00;1.92;×
q_3	0.64;×;1.00;0.64	0.51;×;0.80;0.51	0.41;×;0.64;0.41	...	-10.00;0.64;0.64;0.00	...	×;3.00;1.92;×
q_4	0.64;×;1.00;0.64	0.51;×;0.80;0.51	1.54;×;0.64;1.54	...	-10.00;0.64;0.64;0.00	...	×;3.00;1.92;×
q_5	0.64;×;1.00;0.64	1.23;×;0.80;1.23	1.54;×;0.98;1.54	...	-10.00;0.64;0.64;0.00	...	×;3.00;1.92;×
q_6	0.98;×;1.00;0.98	1.23;×;0.80;1.23	1.54;×;0.98;1.54	...	-10.00;0.98;0.98;1.54	...	×;3.00;1.92;×
q_*	0.98;×;1.00;0.98	1.23;×;0.80;1.23	1.54;×;0.98;1.54	...	-10.00;0.98;0.98;1.54	...	×;3.00;1.92;×
π_*	0.00;0.00;1.00;0.00	1.00;0.00;0.00;1.00	1.00;0.00;0.00;1.00	...	0.00;0.00;0.00;1.00	...	0.00;1.00;0.00;0.00

4.2 值迭代 (14)

➤ 例4.9 赌徒问题

- ✓ 游戏通过投掷骰子累加骰子点数之和来决定赌徒的输赢。
- ✓ 赌徒可以自己选择重新投掷骰子或者结束整局游戏。
- ✓ 如果选择结束整局游戏，骰子总和数刚好18点，则赌徒赢得10元；骰子点数总和超过了18，则输掉（骰子点数总数-18）的资金；少于18，则输掉 $\left(\frac{18 - \text{骰子点数总数}}{2}\right)$ 的资金。
- ✓ 当点数超过或者等于18时，会自动结束整局游戏。

4.2 值迭代 (15)

赌徒问题的MDP数学建模如下：

➤ 状态空间：

当前赌徒骰子点数的总和，即 $S = \{0, 1, 2, 3, 4, 5, \dots, 10, \dots, 23\}$
共 24 个状态。

➤ 动作空间：

赌徒可以选择重新投掷骰子或是结束整局比赛，即两个动作 $\mathcal{A}(s) = \{0, 1\}$ 0代表结束游戏，1代表掷骰子。

4.3. 值迭代 (17)

➤ 状态转移函数:

在当前状态 s 下, 采取动作 a , 到达下一状态 s' 的概率。

假设 $s = 1$ 时执行了动作1。那么 s' 有可能的状态是 $\{2, 3, 4, 5, 6, 7\}$ 共6种状态, 每个状态的概率为 $\frac{1}{6}$ 。但当 $s \geq 36$, 会自动执行动作0来结束整个回合, 并获得回报。

$$p(s, a, s') = \begin{cases} \frac{1}{6}, & a = 1, \quad s \neq [18, 23] \text{ 且 } s' \in [s + 1, s + 6] \\ 1, & a = 0, \quad s = s' \\ 0, & \text{其他} \end{cases}$$

4.2 值迭代 (17)

➤ 奖赏函数:

立即奖赏：赌徒重新投掷骰子会获得0的立即奖赏。

当整局比赛结束，立即奖赏为：

$$r = \begin{cases} -(s-18), & s > 18 \\ +10, & s = 18 \\ -\frac{(18-s)}{2}, & s < 18 \end{cases}$$

目 录

4

前言

4.1

策略迭代

4.2

值迭代

4.3

广义策略迭代

4.4

小结

4.3 广义策略迭代（1）

➤ 广义策略迭代（Generalized Policy Iteration, GPI）

体现了策略评估与策略改进交替进行的一般性，强调策略评估和策略改进的交互关系，而不关心策略评估到底迭代了多少次，或具体的策略评估和策略改进的细节。

在GPI中，策略评估没结束，就可以进行策略改进，只要这两个过程都能不断地更新，就能收敛到最优值函数和最优策略。从这一角度看，值迭代也属于GPI，而实际上几乎所有的强化学习方法都可以被描述为GPI

4.3 广义策略迭代 (2)

➤ 广义策略迭代 (Generalized Policy Iteration, GPI)

GPI体现了评估和改进之间相互竞争与合作的关系：基于贪心策略，使得值函数与当前策略不匹配，而保持值函数与策略一致就无法更新策略。在长期的博弈后，两个流程会趋于一个目标，即最优值函数和最优策略。

4.3 广义策略迭代 (3)

➤ 广义策略迭代 (Generalized Policy Iteration, GPI)

策略总是基于特定的值函数进行改进的，值函数始终会收敛于对应的特定策略的真实值函数，当评估和改进都稳定时，贝尔曼最优方程便可成立，此时得到最优值函数和最优策略。换句话说，值函数只有与当前策略一致时才稳定，且策略只有是当前值函数的贪心策略时才稳定。

4.3 广义策略迭代 (4)

➤ 动态规划一些缺点

- ✓ 在进行最优策略计算时，必须知道状态转移概率 p ;
- ✓ DP的推演是整个树状展开的，计算量大，存储消耗资源多;
- ✓ 每次回溯，所有可能的下一状态和相应动作都要被考虑在内，存在维度灾难问题;
- ✓ 由于策略初始化的随机性，不合理的策略可能会导致算法无法收敛。

目 录

4

前言

4.1

策略迭代

4.2

值迭代

4.3

广义策略迭代

4.4

小结

4.4 小结（1）

- 环境已知的前提下，基于马尔可夫决策过程，动态规划可以很好的完成强化学习任务。策略评估通常对于给定的策略，不断迭代计算每个状态（或状态-动作对）的价值。其迭代方法主要是利用对后继状态（或状态-动作对）价值的估计，来更新当前状态（或状态-动作对）价值的估计，也就是用自举的方法。策略改进是采用贪心算法，利用动作值函数获得更优的策略，每次都选择最好的动作。策略迭代是重复策略评估和策略改进的迭代，直到策略收敛，找到最优的策略。

4.4 小结 (2)

- 但是策略迭代需要多次使用策略评估才能得到收敛的状态（或状态-动作对）值函数，即策略评估是迭代进行的，只有在收敛时，才能停止迭代。值迭代无需等到其完全收敛，提早的计算出贪心策略，截断策略评估，在一次遍历后即刻停止策略评估，并对每个状态进行更新。

4.4 小结 (3)

- 实践证明，值迭代算法收敛速度优于策略迭代算法。广义策略迭代则体现了策略评估与策略改进交替进行的一般性。在GPI中，策略评估和策略改进同时进行，只要这两个过程都能不断地更新，就能收敛到最优值函数和最优策略。从这一角度看，值迭代也属于GPI，而实际上几乎所有的强化学习方法都可以被描述为GPI。

4.5 习题（1）

- 1. （编程）通过策略迭代算法计算：第3章习题2（图3.12）扫地机器人在折扣率 γ 、初始策略为等概率策略的情况下，分别计算确定环境和随机环境下，每个状态的最优状态-动作值。
- 2. （编程）通过策略迭代算法解决更改过的杰克汽车租赁问题。杰克的一个员工在A租赁场工作，家住在B租赁场附近。该员工每天晚上上下班时，愿意免费将一辆车从A租赁场移到B租赁场。其他车辆的移动，包括方向的车辆移动依然需要2美元。在这种情况下，给出汽车租赁问题的解决方案。

4.5 习题 (2)

- 3. (编程) 通过值迭代算法解决赌徒问题。一个赌徒利用硬币投掷的反正面结果来赌博。假如投掷结果是硬币的正面朝上, 那么他将赢得他所压在这一局的相同钱, 如果是反面朝上, 那么他将输掉他的赌金。当这个赌徒赢满100美元或者他输掉他所有的钱时, 赌博结束。每一轮投掷, 赌徒必须取出他资金的一部分作为赌金, 赌金必须是整数。这个问题可以表述为一个非折扣的、阶段有穷马尔可夫决策过程。状态就是赌徒所拥有的资金, $s \in \{1, 2, \dots, 99\}$, 动作就是下赌注, $a \in \{1, 2, \dots, \min(s, 100 - s)\}$ 。除了赌徒达到100美元的目标, 奖赏为+1以外, 其他情况奖赏均为0。状态值函数给出每个状态能够获胜的概率。策略就是如何决定每轮取出多少钱去赌博。最优策略就是使取得最后胜利的概率最大化。 P 代表的就是硬币正面朝上的概率。当 $P=0.4$ 时, 给出值迭代每一轮后值函数的变化情况, 并给出最优策略。

4.5 习题 (3)

- 假设在本章实例确定情况下扫地机器人任务中，对每个状态的立即奖赏都加上一个常量 C ，这对于最终结果是否有影响？请给出解释。
- 5. 简述策略迭代算法和值迭代算法的优缺点，并给出它们各自的适用情况。

The End