题目 9：Jack’s Car Rental Problem&附加题

2018211316班 2018211568号 杜明欣

任务定义

> Jack 有两个租车点，1号租车点和2号租车点，每个租车点最多可以停放20车。Jack 每租出去一辆车可以获利10美金。每天租出去的车与收回的车的数量服从泊松分布。每天夜里，Jack 可以在两个租车点间进行车辆调配，每晚最多调配5辆车，且每调配一辆车花费2美金。
>
> 本题目的任务是：定义上述问题对应的 MDP 四元组，并采用策略迭代方法，求解最优调配策略以使得盈利最优化。

实验环境

> windows系统、spyder软件

方法描述

MDP四元组

定义

S	**1，2号租车点停放车的数量**
A	**1号租车点向2号租车点调配车辆数**
P	**状态转移概率**
R	**状态s下执行a动作能获得期望回报**

计算

> - 状态集S：每个停车点最多停放20辆车，状态（0，0）-（21,21）
> - 动作集A：调配车辆数，动作（-5）- （5）
> - 转移概率P：在状态S下执行a动作使状态变为S'的概率，环境变化由租车、还车、调车引起
> - P(S->S')=P(租车情况) * P(还车情况) 已知租车还车符合泊松分布，且1，2两租车点租车还车行为是相互独立的，使用联合分布
> - 期望回报R：R（S->S'；a）=SUM[P(租车情况) * P(还车情况)*Money(租车收益)]

动态规划

学习过程

> - 初始化策略、状态、动作
> - 策略迭代
>   - 策略评估
>   - 迭代更新值函数矩阵，直到值函数矩阵收敛为止

- 策略改进
  - 逐个状态遍历所有action，取值函数最大的作为新策略

贝尔曼等式

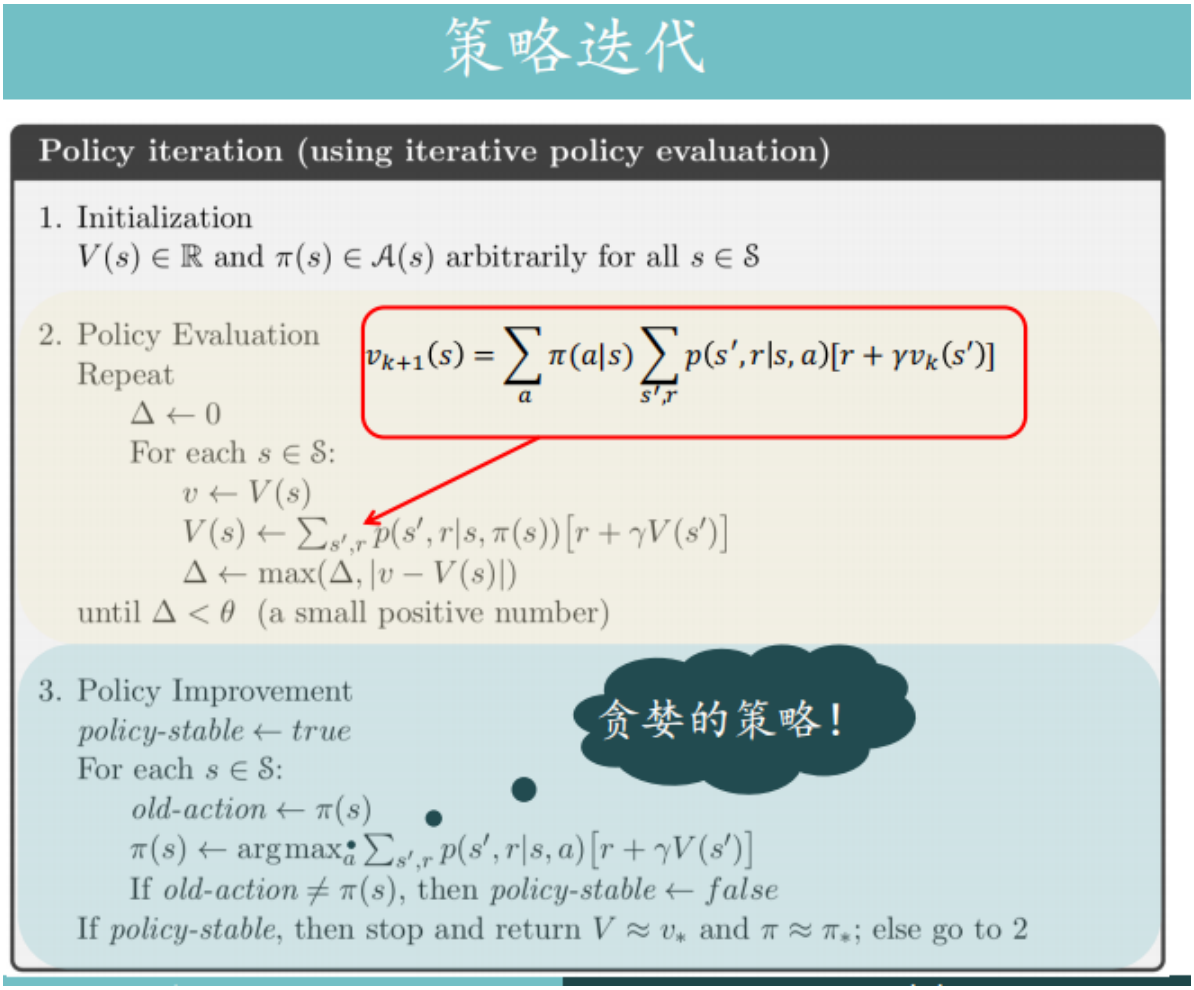
```
R[t] = r[t+1]+r[t+2]+r[t+3]+...
     = r[t+1]+R[t+1]
```

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right]
\end{aligned}
$$

核心算式--值函数计算

```
value+=rent_prob * rec_prob * (rent_earn + GAMMA * value_matrix[car1,car2])
value-=abs(car_now[0]-state[0]) * TRANS_MONEY
```

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



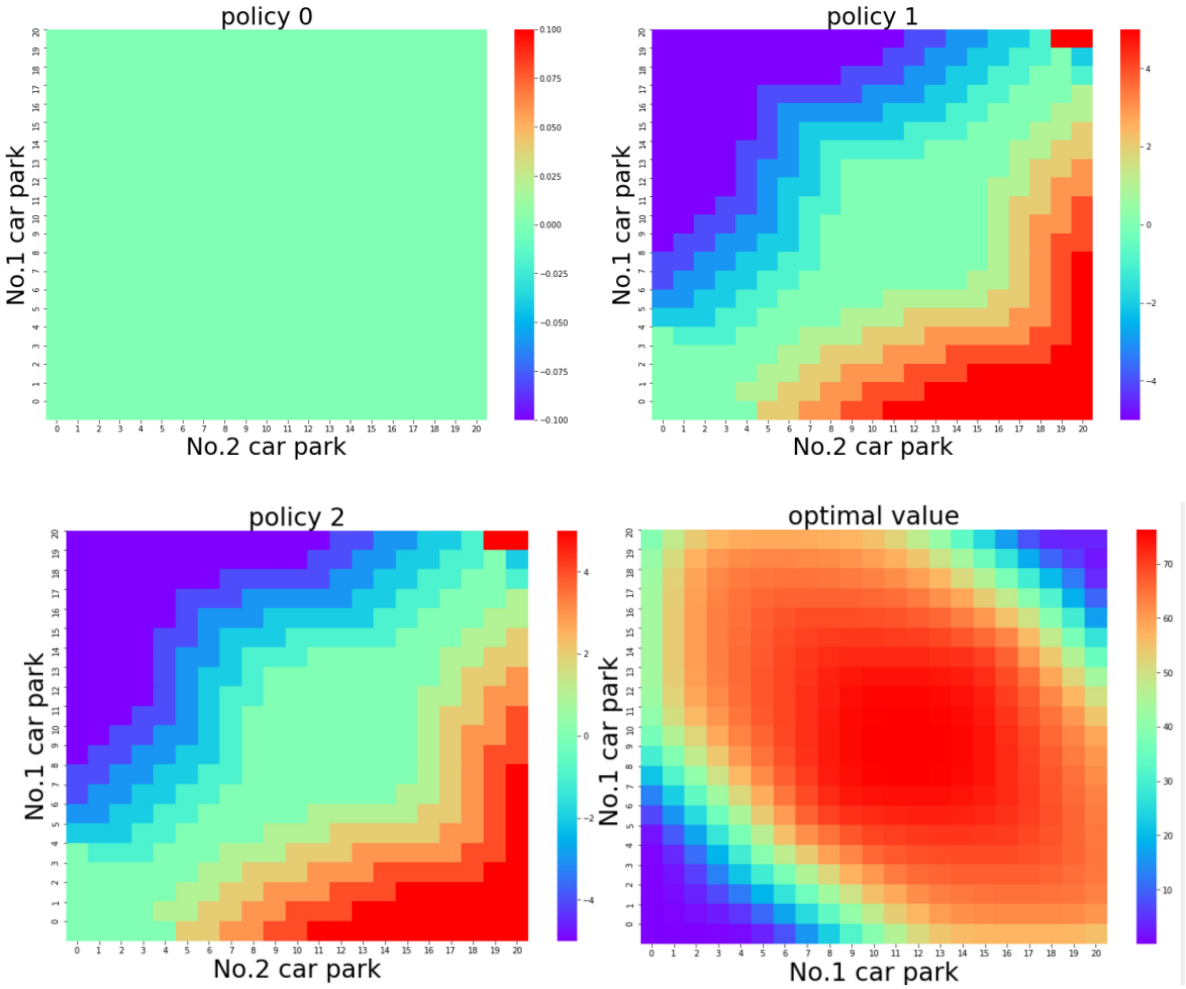
结果分析

最优调配策略

```
[[ 0  0  0  0  0  2  2  3  3  4  4  5  5  5  5  5  5  5  5  5  5]
 [ 0  0  0  0  1  1  2  2  3  3  4  4  4  5  5  5  5  5  5  5  5]
 [ 0  0  0  0  0  1  1  2  2  3  3  3  4  4  4  5  5  5  5  5  5]
 [ 0  0  0  0  0  0  1  1  2  2  2  3  3  3  4  4  4  4  4  5  5]
 [ 0 -1 -1 -1  0  0  0  1  1  1  2  2  2  3  3  3  3  3  4  4  5]
 [-2 -2 -2 -1 -1  0  0  0  0  1  1  1  2  2  2  2  2  3  3  4  5]
 [-3 -3 -2 -2 -1 -1  0  0  0  0  0  1  1  1  1  1  2  2  3  4  5]
 [-4 -3 -3 -2 -2 -1 -1  0  0  0  0  0  0  0  0  1  1  2  3  4  5]
 [-4 -4 -3 -3 -2 -2 -1 -1  0  0  0  0  0  0  0  0  1  2  3  4  5]
 [-5 -4 -4 -3 -3 -2 -2 -1  0  0  0  0  0  0  0  0  1  2  3  4  4]
 [-5 -5 -4 -4 -3 -3 -2 -1  0  0  0  0  0  0  0  0  1  2  3  3  4]
 [-5 -5 -5 -4 -4 -3 -2 -1  0  0  0  0  0  0  0  0  1  2  2  3  4]
 [-5 -5 -5 -5 -4 -3 -2 -1 -1  0  0  0  0  0  0  0  1  1  2  3  3]
 [-5 -5 -5 -5 -4 -3 -2 -2 -1  0  0  0  0  0  0  0  0  1  2  2  3]
 [-5 -5 -5 -5 -4 -3 -3 -2 -1 -1 -1  0  0  0  0  0  0  1  1  2  2]
 [-5 -5 -5 -5 -4 -4 -3 -2 -2 -2 -1 -1 -1 -1 -1  0  0  0  1  1  2]
 [-5 -5 -5 -5 -5 -4 -3 -3 -3 -2 -2 -2 -2 -2 -1 -1  0  0  0  1  1]
 [-5 -5 -5 -5 -5 -4 -4 -4 -3 -3 -3 -3 -3 -2 -2 -1 -1  0  0  0  1]
 [-5 -5 -5 -5 -5 -5 -5 -4 -4 -4 -4 -4 -3 -3 -2 -2 -1 -1  0  0 -1]
 [-5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -4 -4 -3 -3 -2 -2 -1 -1  0 -2]
 [-5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -4 -4 -3 -3 -2 -2 -1  5  5]]
```

迭代过程

```
Epoch 0
经过0轮策略评估，评估震荡为69.87339836834252
经过1轮策略评估，评估震荡为6.170201474950304
经过2轮策略评估，评估震荡为0.5057918384139981
经过3轮策略评估，评估震荡为0.035231285310018734
经过4轮策略评估，评估震荡为0.0018956660899220878
经过5轮策略评估，评估震荡为7.692491475808083e-05
the 0 epoch , policy stable False
```



改进思路

> - 本实验中采用的是确定性策略、然而我们可以采用非确定性策略，具体实现方案如下：
>
>   策略改进时选取值函数最大的前三个动作作为可选策略，并按值函数进行归一化，求出这三个动作的选择概率。
> - 增加策略选择的随机性，使整个强化学习过程充满探索的思想，具体实现方案如下：
>
>   策略改进时设置一个随机数random-index(0-1)，当随机数大于0.8时，随机选取动作作为改进的策略

心得体会

> 本次实验是强化学习的一个小小尝试选用策略迭代方法思路清晰简明，重点在于理解贝尔曼等式以及策略评估、改进的方法。在策略改进过程中使用的是贪婪算法可以增添随机策略，使得强化学习过程存在探索发现的可能。