

文字作业

1 计算MRPs

由于markov奖励过程中的值函数:

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

可以写成矩阵的形式:

$$v = R + \gamma P v$$

$$\begin{bmatrix} V(s_1) \\ \vdots \\ V(s_n) \end{bmatrix} = \begin{bmatrix} R(s_1) \\ \vdots \\ R(s_n) \end{bmatrix} + \gamma \begin{bmatrix} P_{s_1 s_1} & \dots & P_{s_1 s_n} \\ \vdots & \ddots & \vdots \\ P_{s_n s_1} & \dots & P_{s_n s_n} \end{bmatrix} \begin{bmatrix} V(s_1) \\ \vdots \\ V(s_n) \end{bmatrix}$$

所以有:

$$v = (I - \gamma P)^{-1} R$$

P为:

$$\begin{bmatrix} 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.6 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0.4 & 0.4 & 0 & 0 & 0 & 0 \end{bmatrix}$$

R为:

$$\begin{bmatrix} -2 \\ -2 \\ -2 \\ 10 \\ -1 \\ 0 \\ -5 \end{bmatrix}$$

v为:

code1.py

$$\begin{bmatrix} V(\text{科目一}) \\ V(\text{科目二}) \\ V(\text{科目三}) \\ V(\text{通过}) \\ V(\text{玩手机}) \\ V(\text{睡觉}) \\ V(\text{挂科}) \end{bmatrix} = \begin{bmatrix} -3.03851471 \\ -2.05964843 \\ -0.14912107 \\ 10. \\ -2.09441043 \\ 0. \\ -5.74560537 \end{bmatrix}$$

2 说明等价，以及使用时的区别

由于 R_{t+1} 是随机变量，他可能的值是根据 $\pi(a|s)$ 的，所以 R_{t+1} 的期望就等于 $\sum_{a \in A} \pi(a|s) R(s, a)$ ，同理 S_{t+1} 也是随机变量，他的取值也是根据 $\pi(a|s)$ 推导出的，所以 $v_{\pi}(S_{t+1})$ 其实是在求根据策略 π 执行了动作 a 后可能的几个状态的值函数，所以 $v_{\pi}(S_{t+1})$ 的期望就等于 $\sum_{a \in A} \pi(a|s) \sum_{s' \in S} P_{ss'}^a v_{\pi}(s')$ 。

由于状态动作的值函数是先执行一个动作然后再使用策略得到的期望回报值，所以 R_{t+1} 是确定的一个动作，他的期望就等于他本身 $R(s, a)$ ，然后还是因为 S_{t+1} 和 A_{t+1} 都是随机变量， S_{t+1} 是根据前面执行的动作推导出的可能的几个状态， A_{t+1} 是在状态 S_{t+1} 上使用策略 $\pi(a'|s')$ 得到的可能的几个动作，所以 $q_{\pi}(S_{t+1}, A_{t+1})$ 的期望就等于 $\sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_{\pi}(s', a')$ 。

使用时的区别：感觉没有什么区别啊，要想知道期望还是得通过策略和状态转移函数来计算，请老师指正。


3 已知最优 V 函数,能否得到最优策略呢?如果能,写出两者之间的关系,如果不能说明为什么?

可以，通过寻找一个回报值最大的一个片段，然后片段对应的动作序列就是最优策略。

再者，Q函数和V函数是可以相互转化的，所以更加说明通过V函数可以找到Q函数，找到Q函数后再找到最优Q函数，进而找到最优策略。

编程作业

1

code2.py

各状态平均的回报值:

```
gamma = 0.5
average v(s1):-1.32826585753
average v(s2):-1.87104265778
average v(s3):-0.427706479311
average v(s4):2.76026159176
```

```
gamma = 1
average v(s1):-5
average v(s2):-5
average v(s3):0
average v(s4):2
```

遍历寻找最优策略

code2.py

```
optimal_policy:[  
( 's1', 'quit'),  
( 's2', 'phone'),  
( 's3', 'study'),  
( 's4', 'review')]
```

```
optimal_value:[  
-6.66666667e-01,  
-1.33333333e+00,  
3.00000000e+00,  
1.00000000e+01,  
-4.56030660e-16]
```