值迭代与策略迭代

16337341 朱志儒

一、 值迭代算法

值迭代算法的贝尔曼方程如下：

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

值迭代算法步骤：

1. 初始化所有 V(s)；
2. 对于所有的状态，根据贝尔曼方程对状态值更新。对于状态 s 的可执行的每个动作 a，计算执行该动作后到达下一状态的期望值，取期望值最大的价值作为状态 s 的状态值 V(s)，循环执行本步骤直到所有状态值 V(s)收敛；
3. 在第二步中每个状态 s 计算得到的最大的期望值所对应的动作 a 就是在该状态 s 下最应该执行的动作，这就是最优策略。

在本次实验中，状态是指机器人在 Grid World 中所处的位置，动作是指机器人向上、左、右的移动。在贝尔曼方程中，方程右边是指在状态 s 下，执行动作 a，将有不同的概率到达不同的下一状态，然后计算它们的期望。而机器人在 Grid World 中，它执行一个动作后将到达唯一的下一状态，例如，机器人在初始位置（1，1），执行向上走的动作，到达唯一的位置（2，1）而不可能到达其他位置，所以在本次实验中，贝尔曼方程将变成：

$$V_{k+1} \leftarrow \max_a [R(s,a,s') + \gamma V_k(s')]$$

实现后，设$\gamma = 0.9$，结果如下（每次迭代时各个状态所对应的值）：

```
值迭代：
-0.03 -0.03 1.0 0.0
-0.03 0.0 0.87 0.0
-0.03 -0.03 0.753 0.6476999999999999

-0.05699999999999995 0.87 1.0 0.0
-0.05699999999999995 0.0 0.87 0.0
-0.05699999999999995 0.6476999999999999 0.753 0.6476999999999999

0.753 0.87 1.0 0.0
0.6476999999999999 0.0 0.87 0.0
0.5529299999999999 0.6476999999999999 0.753 0.6476999999999999

右 右 右
上    上
上 右 上 左
```

二、 策略迭代算法

如上所述，贝尔曼方程为：

$$V_{k+1} \leftarrow \max_{a}[R(s,a,s') + \gamma V_k(s')]$$

策略迭代算法步骤：

1. 初始化所有状态值 V(s)和策略π(s)；
2. 在当前策略π下，计算根据贝尔曼方程迭代更新每一个状态的 V(s)，直到所有 V(s)均收敛；
3. 对于每个状态，计算在该状态下所有动作的 T = $[R(s,a,s') + \gamma V_k(s')]$值，最大 T 值所对应的动作就是该状态下的新策略。更新完所有策略后，如果所有状态的策略均没有改变，说明策略已经稳定，算法结束；如果存在一个策略发生改变，则回到第 2 步，在新的策略下更新 V(s)，重复 2、3 直到 V(s)和π(s)都收敛，算法结束；
4. 最后得到的π(s)就是最优策略。

实现后，设$\gamma = 0.9$，结果如下（每次更新策略前各个状态所对应的值）：

```
策略迭代：
-0.2999999999999996 -0.2999999999999996 -0.2999999999999996 0.0
-0.2999999999999996 0.0 -0.2999999999999996 0.0
-0.2999999999999996 -0.2999999999999996 -0.2999999999999996 -1.0

-0.2999999999999996 -0.2999999999999996 1.0 0.0
-0.2999999999999996 0.0 0.87 0.0
-0.2999999999999996 -0.2999999999999996 0.753 0.6476999999999999

-0.2999999999999996 0.87 1.0 0.0
-0.2999999999999996 0.0 0.87 0.0
-0.2999999999999996 0.6476999999999999 0.753 0.6476999999999999

0.753 0.87 1.0 0.0
0.6476999999999999 0.0 0.87 0.0
0.5529299999999999 0.6476999999999999 0.753 0.6476999999999999

右 右 右
上    上
上 右 上 左
```