Reinforcement Learning Task1

姚冠宇

2020211260070

策略迭代收敛过慢，用价值迭代，4x4格子两步收敛，两种方法代码都已一并提交

先展示最终结果：

价值迭代：（两轮收敛）

```
yaoguanyu@MacBook-Air ~ % /usr/local/bin/python3 /User
第1轮迭代结果
状态价值：
 0.000  0.000 -1.000 -1.000
 0.000 -1.000 -1.000 -1.000
-1.000 -1.000 -1.000  0.000
-1.000 -1.000  0.000  0.000
策略：
EEEE <ooo <ooo <v>^
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
<v>^ oo>o oo>o EEEE
第2轮迭代结果
状态价值：
 0.000  0.000 -1.000 -1.900
 0.000 -1.000 -1.900 -1.000
-1.000 -1.900 -1.000  0.000
-1.900 -1.000  0.000  0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
价值迭代一共进行2轮
状态价值：
 0.000  0.000 -1.000 -1.900
 0.000 -1.000 -1.900 -1.000
-1.000 -1.900 -1.000  0.000
-1.900 -1.000  0.000  0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
```

策略代表了处于该位置agent会采取什么动作，可以调整环境大小。

策略迭代：（三轮就策略就收敛了没必要到第10轮，这里的轮次是指策略提升的轮次）

```
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
第7轮策略迭代结果
状态价值：
-0.000 -0.000 -0.250 -0.306
-0.000 -0.250 -0.306 -0.250
-0.250 -0.306 -0.250 -0.000
-0.306 -0.250 -0.000 -0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
第8轮策略迭代结果
状态价值：
-0.000 -0.000 -0.250 -0.306
-0.000 -0.250 -0.306 -0.250
-0.250 -0.306 -0.250 -0.000
-0.306 -0.250 -0.000 -0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
第9轮策略迭代结果
状态价值：
-0.000 -0.000 -0.250 -0.306
-0.000 -0.250 -0.306 -0.250
-0.250 -0.306 -0.250 -0.000
-0.306 -0.250 -0.000 -0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
第10轮策略迭代结果
状态价值：
-0.000 -0.000 -0.250 -0.306
-0.000 -0.250 -0.306 -0.250
-0.250 -0.306 -0.250 -0.000
-0.306 -0.250 -0.000 -0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
最终的状态价值函数：
状态价值：
-0.000 -0.000 -0.250 -0.306
-0.000 -0.250 -0.306 -0.250
-0.250 -0.306 -0.250 -0.000
-0.306 -0.250 -0.000 -0.000
策略：
EEEE <ooo <ooo <voo
ooo^ <oo^ <v>^ ovoo
ooo^ <v>^ ov>o ovoo
oo>^ oo>o oo>o EEEE
```

代码注释已经在程序中给出不再赘述，这里大概讲解一下程序结构方便理解

总共定义了三个类一个方法：
class ValueIteration 和 class PolicyIteration 分别代表做策略迭代和价值迭代的agent
class env主要是环境，需要给出奖励值和转移函数

def print_agent()方法是将agent的状态价值函数和在每个位置的动作可视化打印出来方便查看

先讲解class env
存储了每个位置的奖励值，query函数输入当前位置和动作，给出奖励，下一个时刻的位置和转移概率。

剩下两个智能体的class结构比较类似，主要存储环境，状态价值和在每个状态的动作。剩下的区别主要是策略迭代和状态迭代算法带来的区别。

最后在主程序可以自定义行列数（可以不是4行4列），然后输入0或1来查看策略迭代或价值迭代的结果。