强化学习实验报告

201250070 郁博文

伪代码

策略迭代

```
1. 初始化状态值函数 v 和策略 pi
2. v = [0] * n_states
3. pi = [[0.25, 0.25, 0.25, 0.25] for i in range(n_states)]
定义状态转移概率矩阵 P
P = createP()
循环直到收敛：
while True:
4. 初始化最大差值 max_diff 为 0
max_diff = 0
5. 进行策略评估：
for s in range(n_states):
    qsa_list = []
    for a in range(n_actions):
        qsa = 0
        for res in P[s][a]:
            p, next_state, r, done = res
            qsa += p * (r + gamma * v[next_state] * (1 - done))
        qsa_list.append(pi[s][a] * qsa)
    new_v[s] = sum(qsa_list)
    max_diff = max(max_diff, abs(new_v[s] - v[s]))

6. 更新状态值函数 v
v = new_v

7. 如果最大差值小于阈值 theta，结束循环
if max_diff < theta:
    break

8. 进行策略提升：
for s in range(n_states):
    qsa_list = []
    for a in range(n_actions):
        qsa = 0
        for res in P[s][a]:
            p, next_state, r, done = res
            qsa += p * (r + gamma * v[next_state] * (1 - done))
        qsa_list.append(qsa)
    maxq = max(qsa_list)
    cntq = qsa_list.count(maxq)
    pi[s] = [1 / cntq if q == maxq else 0 for q in qsa_list]
```

价值迭代

```
1. 初始化状态值函数 v 和策略 pi
2. 定义状态转移概率矩阵 P
3. 循环直到收敛：
   4. 初始化最大差值 max_diff 为 0
   5. 对于每个状态 s in 状态空间：
      6. 初始化动作值函数 qsa_list 为空列表
      7. 对于每个动作 a in 动作空间：
         8. 初始化动作值函数 qsa 为 0
         9. 对于每个转移结果 res in 状态转移概率矩阵 P[s][a]：
            10. 获取转移概率 p, 下一状态 next_state, 奖励 r, 终止标志 done
            11. 更新动作值函数 qsa += p * (r + gamma * v[next_state] * (1 - done))
         12. 将 qsa 加入动作值函数列表 qsa_list
      13. 更新状态值函数 v[s] 为 qsa_list 中的最大值
      14. 更新最大差值 max_diff 为 max(max_diff, abs(new_v[s] - v[s]))
   15. 更新状态值函数 v 为 new_v
   16. 如果 max_diff 小于阈值 theta，则跳出循环
7. 根据状态值函数 v 更新策略 pi：
   18. 对于每个状态 s in 状态空间：
      19. 初始化动作值函数 qsa_list 为空列表
      20. 对于每个动作 a in 动作空间：
         21. 初始化动作值函数 qsa 为 0
         22. 对于每个转移结果 res in 状态转移概率矩阵 P[s][a]：
            23. 获取转移概率 p, 下一状态 next_state, 奖励 r, 终止标志 done
            24. 更新动作值函数 qsa += p * (r + gamma * v[next_state] * (1 - done))
         25. 将 qsa 加入动作值函数列表 qsa_list
      26. 更新最大动作值 maxq 为 qsa_list 中的最大值
      27. 统计最大动作值的个数 cntq
      28. 更新策略 pi[s] 为 [1 / cntq if q == maxq else 0 for q in qsa_list]
8. 输出状态值函数 v 和策略 pi
```

悬崖漫步环境设计

在这个环境中，使用一个二维的方格世界，包含了 nrow 行和 ncol 列。其中，每个方格都可以看作是一个状态（state），总共有 nrow * ncol 个状态。

环境的状态转移概率由 createP() 方法定义，该方法返回一个三维的列表 P，其中 P[s][a] 是一个包含多个三元组的列表，表示在状态 s 下采取动作 a 后可能转移到的下一个状态、奖励值和是否终止的信息。

具体而言，环境中的每个状态都有四个可能的动作，分别是向上、向下、向左和向右，分别对应 ^,V<,> 这四个动作。在状态转移过程中，可以看到以下设计：

1. 如果当前状态位于悬崖（即最后一行且不在最右侧的方格），则在采取任何动作后都会转移到终止状态（即最后一行最右侧的方格），并且会获得奖励值 -100。
2. 如果当前状态不位于悬崖，则根据当前状态和采取的动作计算下一个状态。如果下一个状态位于悬崖，那么转移到该状态后会终止，并获得奖励值 -100。否则，转移到下一个状态后会获得奖励值 -1。
3. 当下一个状态为终止状态时，设置 done 标志为 True，表示本回合结束。

通过这种状态转移设计，可以模拟悬崖漫步环境的特点，即在最短路径上有一个悬崖，智能体需要通过学习来避免掉落到悬崖上，从起始状态尽可能快地到达终止状态，并获得最大的累积奖励。

策略迭代过程

1. 策略评估（policy evaluation）： 在这段代码中，策略评估通过循环迭代更新状态价值函数（self.v）来估计当前策略的价值函数。首先，初始化状态价值函数为0。然后，通过对每个状态执行贝尔曼方程的更新操作，计算状态的价值函数。具体地，对于每个状态 s，遍历所有可能的动作 a，计算动作 a 在状态 s 下的动作值函数（Q 值），并根据策略 pi(s) 权重对所有动作的动作值函数加权求和，得到状态 s 的新价值函数（self.v[s]）。重复执行这个过程直到状态价值函数的变化小于阈值 self.theta，即 max_diff < self.theta。这个过程称为策略评估，用于估计当前策略下的状态价值函数。
2. 策略提升（policy improvement）： 在这段代码中，策略提升通过更新策略（self.pi）来改善策略。首先，对于每个状态 s，计算在当前状态价值函数下选择每个动作的动作值函数（Q 值），选择具有最大动作值函数的动作作为新的最优动作，并更新策略 pi(s) 为新的最优动作。如果有多个动作具有相同的最大动作值函数，则将概率平均分配给这些最优动作。这个过程称为策略提升，用于根据估计的状态价值函数更新策略。

在策略迭代算法中，策略评估和策略提升两步反复进行，直到策略不再改变（old_pi == new_pi），即达到最优策略。整个过程中，通过不断迭代更新状态价值函数和策略，智能体逐步优化策略，以在环境中找到最优的行为策略。

价值迭代过程

1. 初始化值函数和策略：将值函数v初始化为0，并将策略pi初始化为一个随机策略。
2. 进行价值迭代：在每次迭代中，遍历所有可能的状态s，并计算在当前策略下从状态s执行所有可能的动作a所得到的累积奖励qsa。根据贝尔曼方程，更新值函数v[s]为qsa的最大值。
3. 更新策略：在每次迭代中，根据更新后的值函数v计算新的策略pi。对于每个状态s，选择使得qsa最大的动作a作为新的策略。
4. 判断收敛：在每次迭代中，计算值函数的更新幅度max_diff，如果其小于预定的阈值theta，则停止迭代。
5. 输出结果：在算法收敛后，输出最终的值函数和策略。值函数表示了在每个状态下的预期累积奖励，策略表示了智能体在每个状态下选择的动作。

在代码中，通过while循环进行迭代，直到max_diff小于设定的阈值theta。在每次迭代中，分别计算每个状态下所有动作的累积奖励qsa，并更新值函数v和策略pi。最终，输出更新后的值函数和策略，以及最终的状态价值和策略的可视化结果。

实验结果

策略迭代

```
C:\Users\dell\anaconda3\python.exe C:/Users/dell/Desktop/机器学习/homework05/policy.py
策略评估进行60轮后完成
策略提升完成:
策略评估进行72轮后完成
策略提升完成:
策略评估进行44轮后完成
策略提升完成:
策略评估进行12轮后完成
策略提升完成:
策略评估进行1轮后完成
策略提升完成:
状态价值:
-7.712 -7.458 -7.176 -6.862 -6.513 -6.126 -5.695 -5.217 -4.686 -4.095 -3.439 -2.710
-7.458 -7.176 -6.862 -6.513 -6.126 -5.695 -5.217 -4.686 -4.095 -3.439 -2.710 -1.900
-7.176 -6.862 -6.513 -6.126 -5.695 -5.217 -4.686 -4.095 -3.439 -2.710 -1.900 -1.000
-7.458  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
策略:
ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovoo
ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovoo
ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ovoo
^ooo **** **** **** **** **** **** **** **** **** **** EEEE

Process finished with exit code 0
```

价值迭代

```
 价值迭代一共进行14轮
状态价值:
-7.712 -7.458 -7.176 -6.862 -6.513 -6.126 -5.695 -5.217 -4.686 -4.095 -3.439 -2.710
-7.458 -7.176 -6.862 -6.513 -6.126 -5.695 -5.217 -4.686 -4.095 -3.439 -2.710 -1.900
-7.176 -6.862 -6.513 -6.126 -5.695 -5.217 -4.686 -4.095 -3.439 -2.710 -1.900 -1.000
-7.458  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
策略:
ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovoo
ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovo> ovoo
ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ovoo
^ooo **** **** **** **** **** **** **** **** **** **** EEEE

Process finished with exit code 0
```

gym策略迭代

```
C:\Users\dell\anaconda3\python.exe C:/Users/dell/Desktop/机器学习/homework05/gympolicy.py
C:\Users\dell\anaconda3\lib\site-packages\gym\envs\toy_text\frozen_lake.py:271: UserWarning: WARN: You are calling render method
  logger.warn(
冰洞的索引: {11, 12, 5, 7}
目标的索引: {15}
[(0.3333333333333333, 10, 0.0, False), (0.3333333333333333, 13, 0.0, False), (0.3333333333333333, 14, 0.0, False)]
[(0.3333333333333333, 13, 0.0, False), (0.3333333333333333, 14, 0.0, False), (0.3333333333333333, 15, 1.0, True)]
[(0.3333333333333333, 14, 0.0, False), (0.3333333333333333, 15, 1.0, True), (0.3333333333333333, 10, 0.0, False)]
[(0.3333333333333333, 15, 1.0, True), (0.3333333333333333, 10, 0.0, False), (0.3333333333333333, 13, 0.0, False)]
策略评估进行25轮后完成
策略提升完成:
策略评估进行58轮后完成
策略提升完成:
状态价值:
 0.069  0.061  0.074  0.056
 0.092  0.000  0.112  0.000
 0.145  0.247  0.300  0.000
 0.000  0.380  0.639  0.000
策略:
<ooo ooo^ <ooo ooo^
<ooo **** <o>o ****
ooo^ ovoo <ooo ****
**** oo>o ovoo EEEE

Process finished with exit code 0
```

gym价值迭代

```
C:\Users\dell\anaconda3\python.exe C:/Users/dell/Desktop/机器学习/homework05/gymvalue.py
C:\Users\dell\anaconda3\lib\site-packages\gym\envs\toy_text\frozen_lake.py:271: UserWarning: WARN: You are calling render method without sp
  logger.warn(
冰洞的索引: {11, 12, 5, 7}
目标的索引: {15}
[(0.3333333333333333, 10, 0.0, False), (0.3333333333333333, 13, 0.0, False), (0.3333333333333333, 14, 0.0, False)]
[(0.3333333333333333, 13, 0.0, False), (0.3333333333333333, 14, 0.0, False), (0.3333333333333333, 15, 1.0, True)]
[(0.3333333333333333, 14, 0.0, False), (0.3333333333333333, 15, 1.0, True), (0.3333333333333333, 10, 0.0, False)]
[(0.3333333333333333, 15, 1.0, True), (0.3333333333333333, 10, 0.0, False), (0.3333333333333333, 13, 0.0, False)]
 价值迭代一共进行60轮
状态价值:
 0.069  0.061  0.074  0.056
 0.092  0.000  0.112  0.000
 0.145  0.247  0.300  0.000
 0.000  0.380  0.639  0.000
策略:
<ooo ooo^ <ooo ooo^
<ooo **** <o>o ****
ooo^ ovoo <ooo ****
**** oo>o ovoo EEEE

Process finished with exit code 0
```