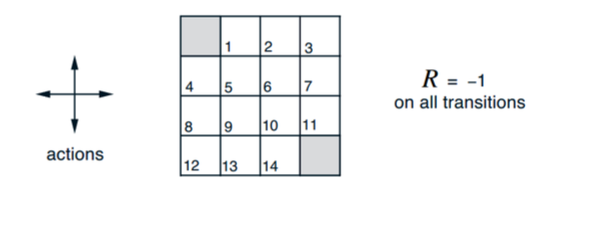
实验1：策略迭代与值迭代

1. 实验目的
2. 熟悉并掌握MDP的内容；
3. 深入理解策略迭代，掌握策略迭代的基本原理；
4. 掌握强化学习设计与编程实现的基本技能
5. 实验环境
6. 接入Internet 的实验主机；
7. Windows 7/8/10操作系统，IOS, linux操作系统等
8. 实验内容

一个4x4的Gridworld，位置(0,0)是一个reward为0的终结态，位置(3,3)是一个reward为10的终结态，位置(2,2)是一个reward为-10的终结态，其余全部为初始为0的非终结态。每个非终结态有4个动作(上，下，左，右)，每移动一步（不管是否撞墙）的代价都是-1，撞墙会回到原来的位置（即原地不动）。求每个位置到达获利最大的终结态的策略。

加分实验：

考虑如下Gridworld



非终止的状态为 S = {1,2，…，14}，阴影的方格表示终止状态。每个状态下的候 选动作集为 A={up，down，right，left}，每个动作对应的状态转移都是固定的: 例如，p(6,-1|5,right)=1, p(7,-1|7,right)=1, p(10,r|5,right)=0.在到达终点之前，每走一步的收益均为-1。

现假设在状态 13 下面引入一个额外状态 15，状态 15 下采取动作 left，up，right， down 相应转移到状态 12,13,14,15。所有原来状态的状态转移函数保持不变。(1) 计算在均等随机策略下vπ(15)的值。(2)假设在状态 13 下的状态转移函数修 改为:采取动作 down 会转移到状态 15，重新计算在均等随机策略下v (15)的 π 值。

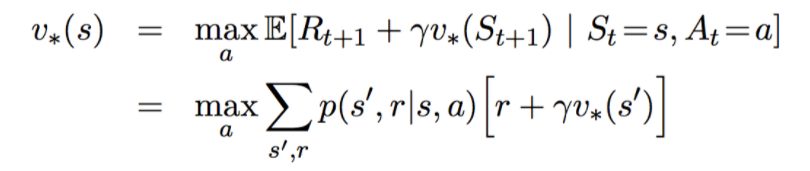
1. 实验步骤

【具体方法参考课程使用书籍】

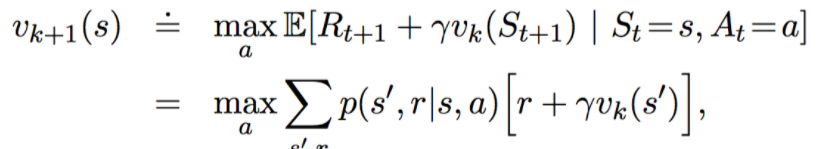
对问题的分析，解决方案的各个环节的功能要求，描述解决方案时，应当注意的条件及限制规则。

问题分析：该问题有一个reward为10的终结态和reward为-10的终结态。各点走向应该尽量避免走-10的终结态，尽量向+10的终结态靠近。但是因为还有一个0的终结态，考虑到路上行走所花的代价，应该尽量权衡两者利弊。

Value iteration：采取bellman最优方程得到



将该式子转化为迭代形式



最终收敛后即可做出最优策略

Policy iteration：初始策略为全部向右走。基于该策略进行策略评估，使用bellman方程更新Value Function。然后基于Value Function对该策略进行策略改进，使用Greedy policy产生出当前的最优policy继续进行策略评估。直到当前策略与Greedy policy产生的策略一致为止。最终的策略即为最优策略。

需要注意的问题：不同于random policy，状态转移时一但有最优的策略选择就一定要选择最优的策略。

加分实验：

第一问与第二问的区别在于状态13向下转移时是否可以到达状态15.问题一不能到达，问题二可以到达。其他与基于ramdom policy下的迭代基本一致

1. 实验方式

每位同学独立上机编程实验，实验指导教师现场指导。

1. 参考内容

强化学习

参考书籍：《Reinforcement Learning:An Introduction》

作者：Richard S. Sutton and Andrew G. Barto

1. 相关软件下载

Code::Blocks <http://www.codeblocks.org/> (C++编辑器)

PyCharm <http://www.jetbrains.com/pycharm/> （Python编辑器）

1. 实验报告要求
2. 原题描述
3. 问题的分析与公式原理
4. 算法实现：代码以及必要的注释
5. 实验结果以及结果分析
6. 实验心得（未来的展望与思考）