Q-Learning算法在随机迷宫问题中的实现

邱光磊 XXX XXX

北京石油化工学院计算机科学与技术系

E-mail：[qiuguanglei@outlook.com](mailto:qiuguanglei@outlook.com)

摘要：该文介绍了Q-Learning算法及其在随机迷宫中的实现。

关键词：强化学习；Q-Learning；迷宫；

Q-Learning算法本质上是一种经验学习，在迷宫终点设定一个统一的最大经验值，给每一次选择设定一个学习系数，使得每次选择能从之后的结果中获得一定的经验。在足够多次的尝试中，每一步选择能够获得的经验值会收敛，根据每一次选择能够获得的经验值，便可得出最优路径。

对于迷宫而言，一共需要两个矩阵来分别记录规定的奖励和获得的经验。奖励矩阵也称R矩阵（Reward），所得经验矩阵称为Q矩阵（Quality）。Q矩阵用来衡量每一个动作所能得到的收益，它会随着尝试次数的增加而发生变化，最终收敛。

以一个3×3的迷宫为例：



R矩阵的大小应为9×9，列记录了现在的所在位置，即当前的状态（State），行记录了下一个房间，即接下来的动作（Action）。这样每一个R矩阵中的值，都表示了当前状态下执行某个动作能获得的收益，对于直接通向终点z的房间a，人为规定一个统一的最大奖励100，因此在R矩阵中状态a对应的动作z的奖励值便为100，即R矩阵中（a，z）的值为100；对于相通但动作不为终点的(state, action)，将它的值设为0，其余所有不通的(state, action)，将它的值设为-1 ，在这个例子中人为规定0号房间为起点，6号房间为终点：



Q矩阵可以看作是一个探员（Agent）的大脑，在最开始，他脑中没有任何经验，他不知道从当前的状态执行某一个动作能过获得什么样的收益，因此Q矩阵也是一个9×9的(state, action)矩阵，并且所有初值均为0：



该方法所要做的事情就是在足够多的尝试下，用奖励矩阵来更新经验矩阵，待经验矩阵收敛后，便可根据每项动作所能获得的收益寻求出最优解，并且是全局最优解。

Q矩阵的更新公式如下：

Q(state, action) = R(state, action) + Gamma × Max[Q(next state, all actions)]

Q矩阵当前(state, action)的值会受到当前能够获得的经验和接下来的(state, action)值的影响，并通过设定学习系数伽马（γ）来从下一次选择中获得经验，伽马的最大值为1，伽马越大则表示越靠后的累积奖赏越重要。当前的经验获得由R矩阵决定，下一次选择能够获得的经验由接下来的所有选择中奖励最高的选择决定。

以当前所处房间7（state = 7），下一步动作为6（action = 6），Gamma = 0.8为例：

Q(7, 6) = R(7, 6) + 0.8 × Q(6, 7) = 100

这样Q矩阵得到一次更新：



下面给出整个Q-Learning算法的计算步骤：

1. 给定参数Gamma和奖励矩阵R
2. 初始化Q矩阵为0矩阵
3. 对于每一步：
   1. 随机选择一个初始的状态state
   2. 若未达到目标状态，则：
      1. 在当前状态state的所有可能行为中选取一个行为action
      2. 利用选定的行为action，得到下一个状态next state
      3. 按照Q矩阵更新公式计算Q(state, action)
      4. 令当前state变为next state

在足够多次的更新后，Q矩阵会收敛至一个固定数值的矩阵，此时便可根据Q矩阵中每步所得的奖励确定出最优路径。

经过100次训练后已经收敛的Q矩阵：



一旦Q矩阵收敛，每一个State对应的最大经验Action的组合便是最优路径：

【0-1-4-7-6】

在随机迷宫中，由于每次随机生成的迷宫中房间的连通性是随机的，起始点和终点的位置也是随机的，因此需要首先记录迷宫中房间的通路情况，事实上就是确定下R矩阵的情况。

由于每个房间只有上下左右四个房间可以与其相通，且相邻的四个房间与中间的房间均有对应的数学关系，在本例中，5号房间对应的上下左右分别是：（5-3），（5+3），（5-1），（5+1）。

在随机设定始末点的情况下，会出现没有通路连接起点与终点的情况，此时需根据房间数设定最大步数设定，防止出现死循环。