策略迭代：收敛速度跟v(s)迭代轮数有关

loop

1.先设立一个固定的策略pi(s)，即每个状态对应的action固定。

2.利用状态-价值v(s)所对应的Bellman方程迭代使得v(s)收敛。

3.利用q(s,a)和v(s)的关系求得q(s,a)，需要知道状态转移方程。

4.新的pi(s)由argmax(q(s,a))得到。

5.pi(s)=新的pi(s)。then: goto loop。

特殊：1轮策略迭代

价值迭代：采用动态规划来解决，需要有episode的终点，有一个初始的v(s)函数。（收敛相对较快但没有1轮策略迭代快）

loop

1.根据q(s,a)和v(s)的关系求得q(s,a),需要知道状态方程。

2.新的v(s)=max(q(s,a))。then: goto loop

得到收敛的v(s)后得到pi(s)。

广义策略迭代:交替使用1轮策略迭代和价值迭代（提高收敛速度）

Model-Free:状态转移概率矩阵未知，则Bellman公式有一个转变，由状态转移概率的加权平均变成期望

算法：

1.确定一个初始策略。

2.用策略进行游戏得到一些序列（episode）。

3.一定次数的迭代后得到值函数。

4.得到值函数相当于进行策略的评估，之后进行策略提升，直到收敛。

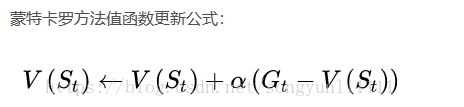
问题：

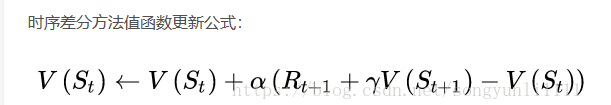
(1)如何得到这些游戏序列。

(2)如何使用序列进行评估。

1.蒙特卡罗方法：

Bellman公式由转移概率加权得到的部分由期望代替。





基于策略梯度的算法：

将值函数表示为策略参数的某个函数