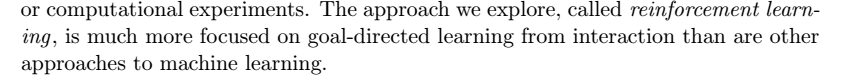
Chapter 1

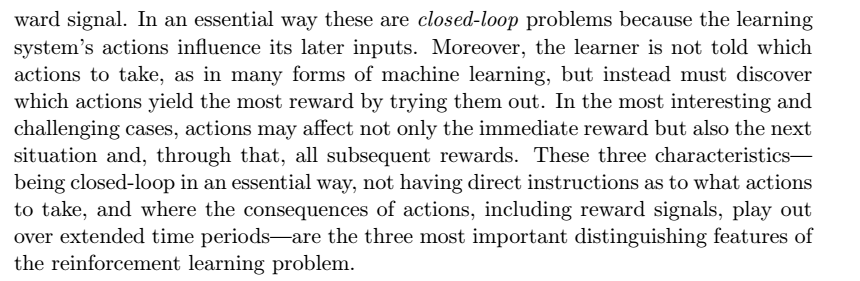
The reinforcement learning problem





1.1Reinforcement Learning

Reinforecement learning problems involve learning what to do-how to map situations to action -so as to maximize a numerical reward signal.



1.当前动作影响接下来的动作

2.没有直接的动作指示，只是告诉系统要产生最大的回报

3.当前动作不单只影响当前的回报，还有接下来的。

supervised learning is learning from a training set of labeled examples provided by a knowledgable external supervisor.(输入X可看做状态，标签y可看做动作。监督学习即认为是根据不同的状态做出不同的动作，这些动作是已知的。)

unsupervised learning is typically about finding structure hidden in collections of unlabeled data.

reinforcement learning is trying to maximize a reward signal instead of trying to find hidden structure.

One of the challenges that arise in reinforcement learning,and not in other kinds of learning ,is the trade off between exploration（探索更好的动作） and exploitation(采用已知好的动作)

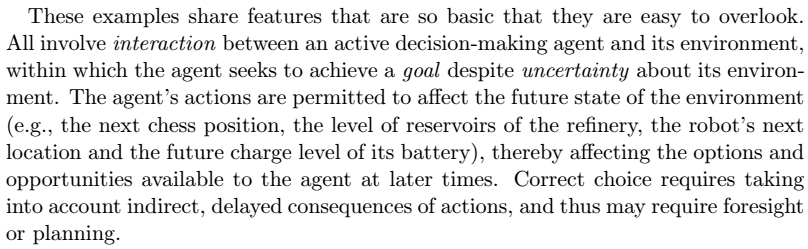
1.2 examples

1.alphago

2.自适应调整参数的控制器

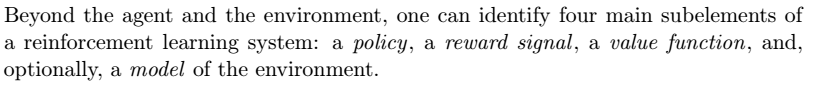
3.移动机器人决定其是否进入新的屋子收集更多的东西或者退回充电地方进行充电。

4.机器人准备早餐



（接上文）At the same time ,in all these example the effects of actions cannot be fully predicted;thus the agent must monitor its environment frequently and react appropriately. All these examples involve goals that are explicit in the sense that the agent can judge progress toward its goal based on what it can sense directly. In all of these examples the agent can use its experience to improve its performance over time

1.3 Element of Reinforcement Learning



A policy defines the learning agent's way of behaving at a given time.Roughly speaking,a policy is a mapping from perceived states of the environment to actions to be taken when in those states.In some cases the policy may be a simple function or lookup table,whereas in other s it may involve extensive computation such as a search process.

In general ,polices may be stochastic

A reward signal defines the goal in a reinforcement learning problem.On each time setp, the environment sends to the reinforcement learning agent a single number ,a reward.The agent's sole objective is to maximize the total reward it receives over the long run.The agent cannot change the function that generates the reward signal.In general,reward signals may be stochastic functions of the state of the environment and the actions taken.

鉴于回报信号告诉我们当前状况的好坏，价值函数则告诉我们在长期运行中的好坏。简单来说，一个状态对应的价值是它未来所有回报的总和。

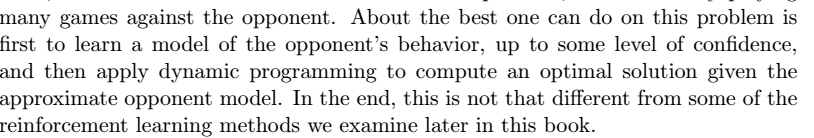
没有回报，就无法得到价值。并且估计价值的唯一目的就是得到更多的回报。然而，当我们做决定时，价值是考虑最多的因素。 事实上，我们所考虑的，几乎是所有强化学习算法中最重要的部分就是找到一个能有效估计价值的方法。

a model of the environment ,模仿坏境的行为，更通俗的，就是能够推断环境将会有何反应。

.1.4 Limitation and scope

在本书中我们所考虑的大多数强化学习方法都是围绕估计价值函数，但仅仅如此是无法解决强化学习问题的。

1.5 An Extended Examoke:Tic-Tac-Toe



先跟对手玩，并得到对手的行为习惯，然后运用动态规划区计算一个最优的方案(常规方法)



井字棋强化学习方法：

1.设置一个数字表，每个数字代表该游戏可能的状态对应的胜利概率，并所有初始化为0.5(表示我们有50%获胜的概率)，该数值认为是状态值，整个表就是一个学习价值函数

2.假设我们已经跟对手玩了很多次了，那么表已经更新过了。通常我们会服从贪心策略，即选择当前状态所能获得最大价值的动作并移动下一状态。然而，偶尔我们会随机选择其他移动。这称为探索移动因为它们可能会使我们到达从未达到的状态。

3.在游戏过程中，游戏状态对应的价值也会不断更改，我们尝试使它们更接近于该状态对应的真实获胜概率。为了实现这一目的，我们会在每一次贪心搜索后更新当前状态的价值值。

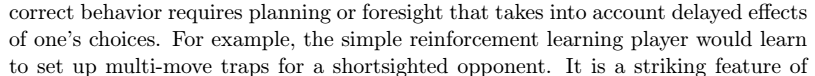


s表示贪心移动前的状态，s'表示贪心移动后的状态。

a是一个正参数，，称为步长参数

该例子阐述了强化学习方法几个关键特征。

1.跟环境进行交互时会有强调学习

2.有一个清晰的目标，

一个强化学习系统在极大的状态集上能有多有效的解决问题与它从过去的经验生成有用的信息是紧密联系的。因此，强化学习需要监督学习方法

self-play（自己与自己玩）

2.评估反馈

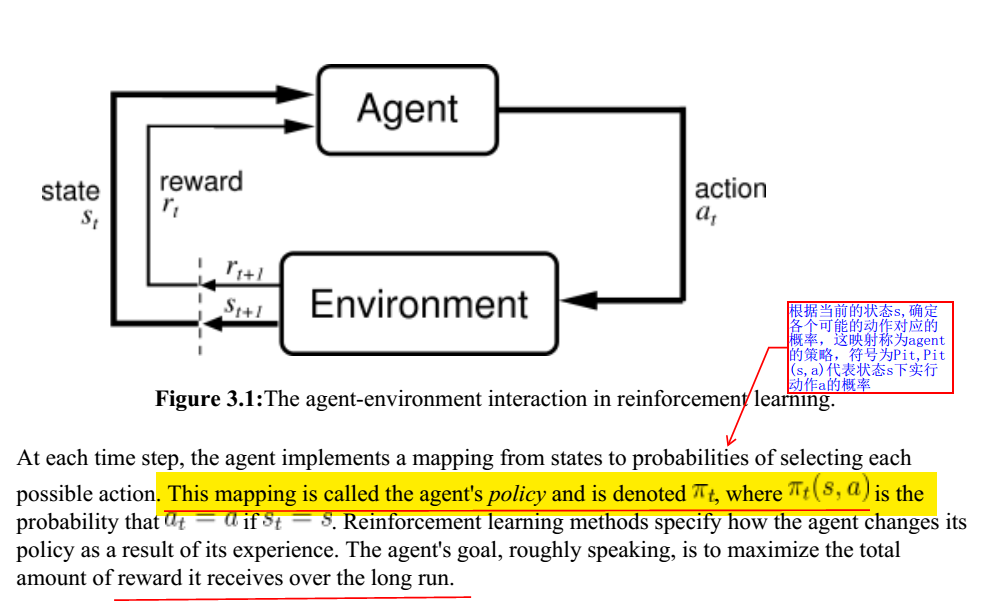
evaluative feedback(reinforce learn)

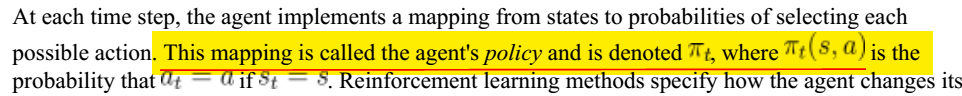
instructive feedback(supervised learn)

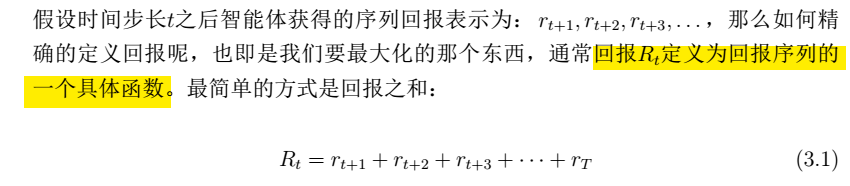
2.2

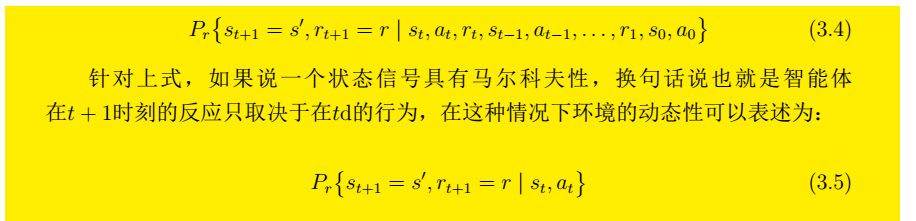
状态s下agent'policy()，即是一个动作

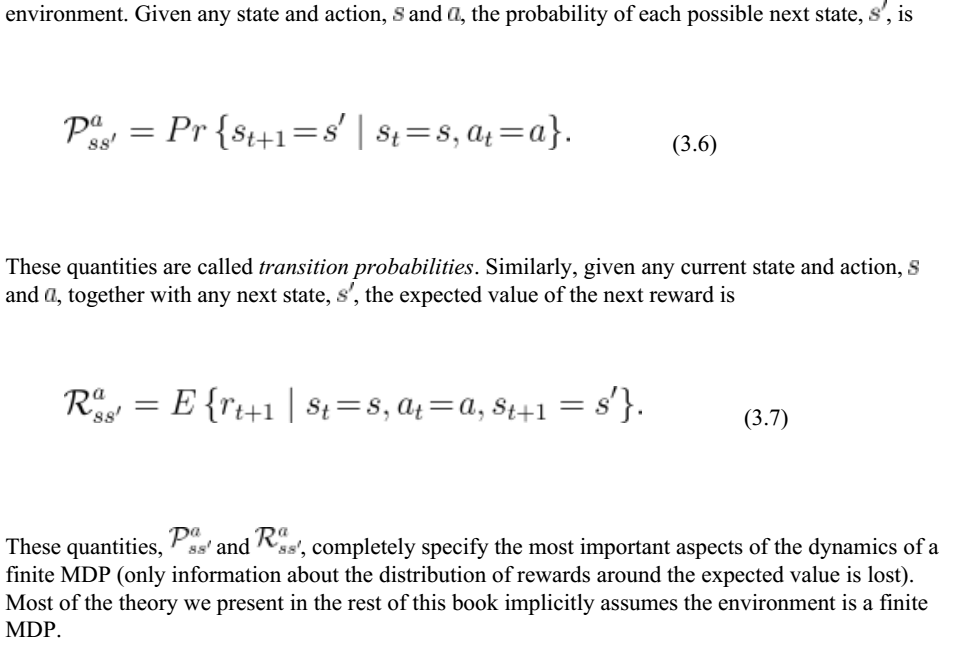
V 价值函数

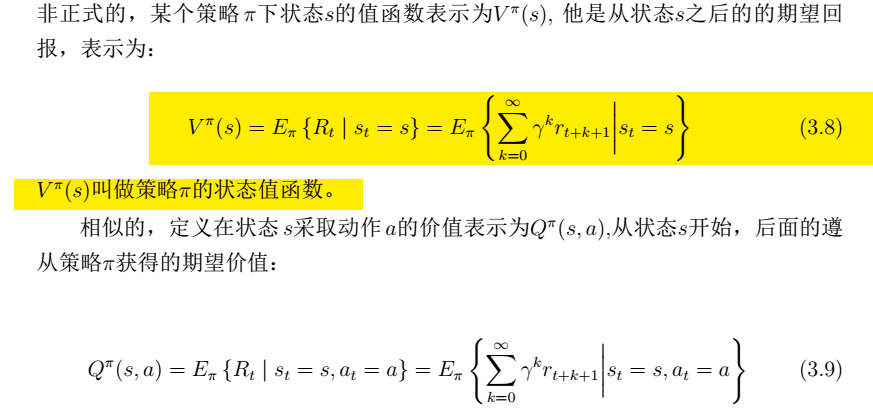


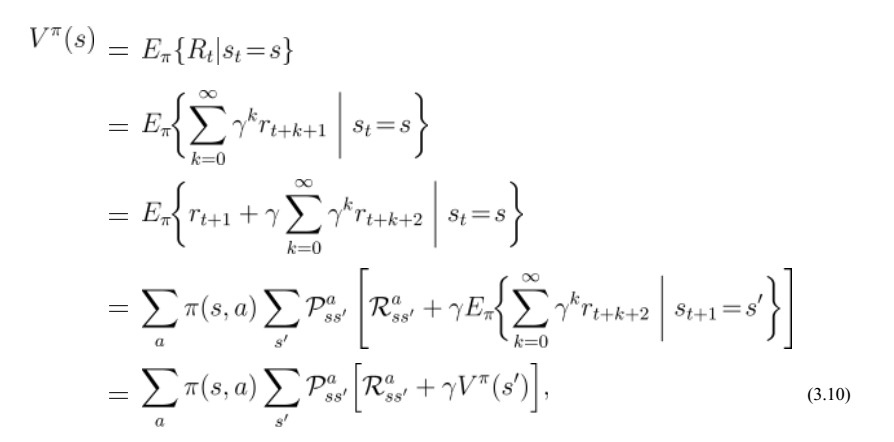


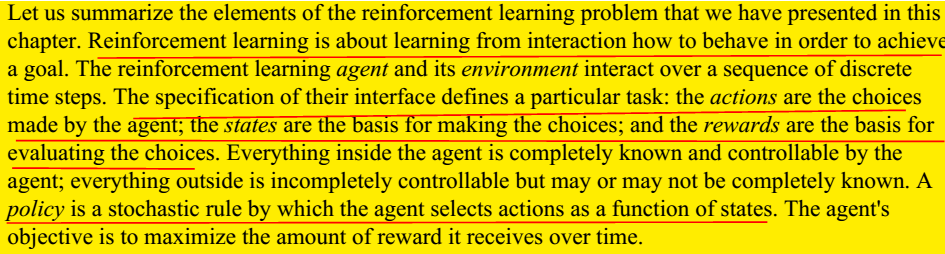








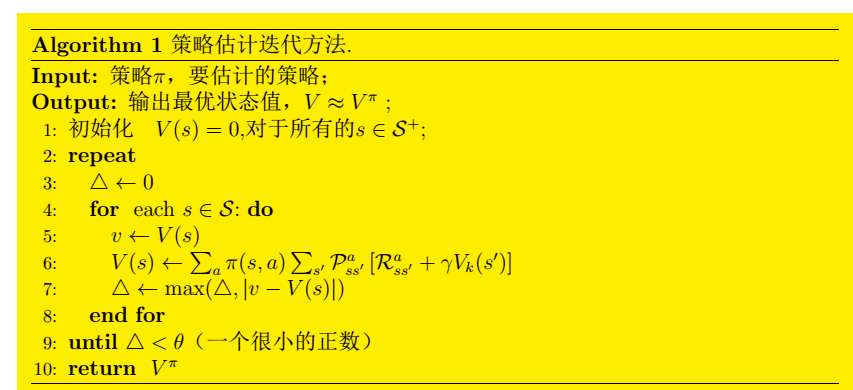


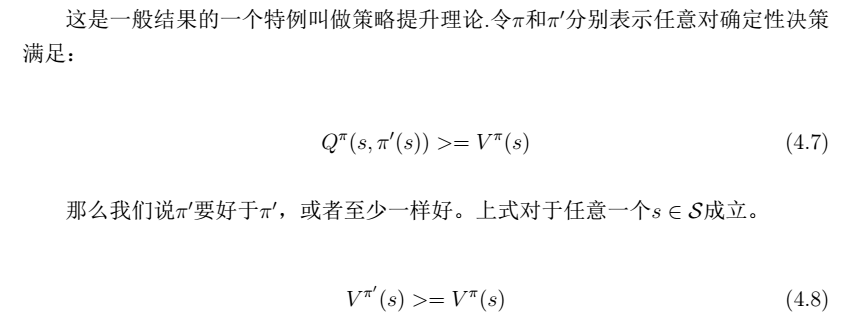


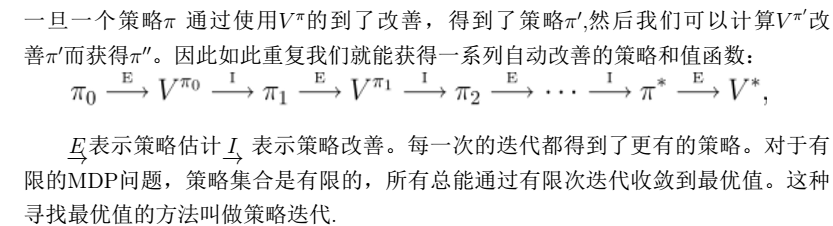
**Chapter 4**

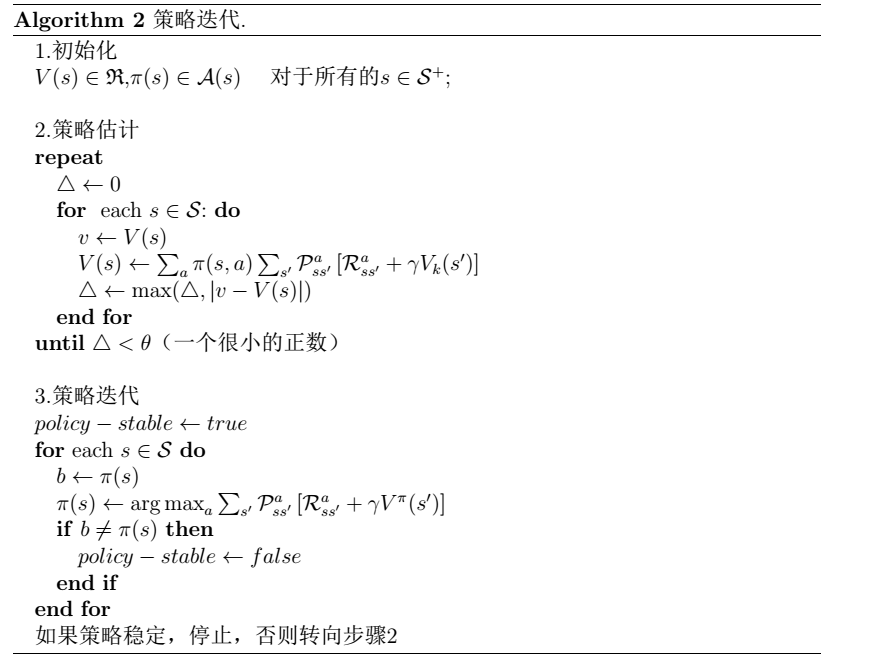
**动态规划**

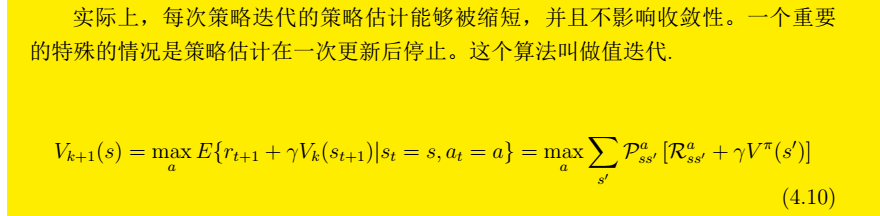
**动态规划这个词指的是用于在马尔科夫决策过程中针对确定模型求解最优策略的一类算法。**

****





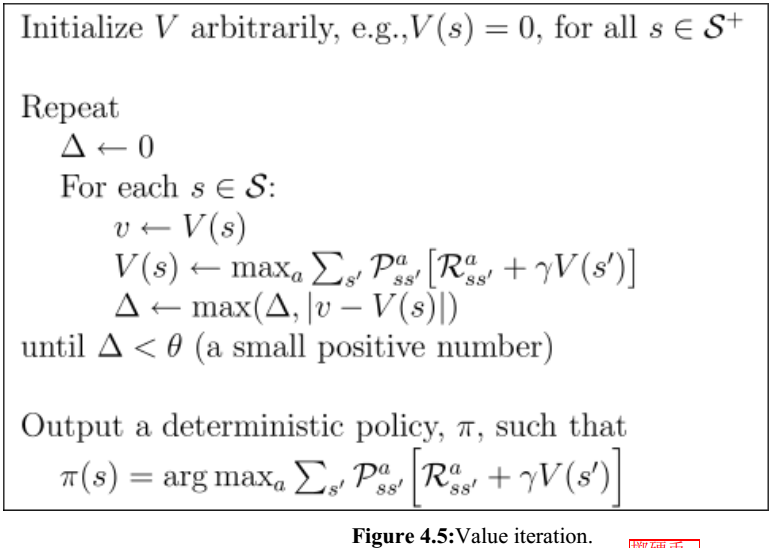




注意策略迭代和值迭代的区别。策略迭代中策略更新计算的是每个状态s下各个行为的期望值，然后用这个值覆盖掉原来的

而值迭代用的是用某个行为的最大值更新原来的值.



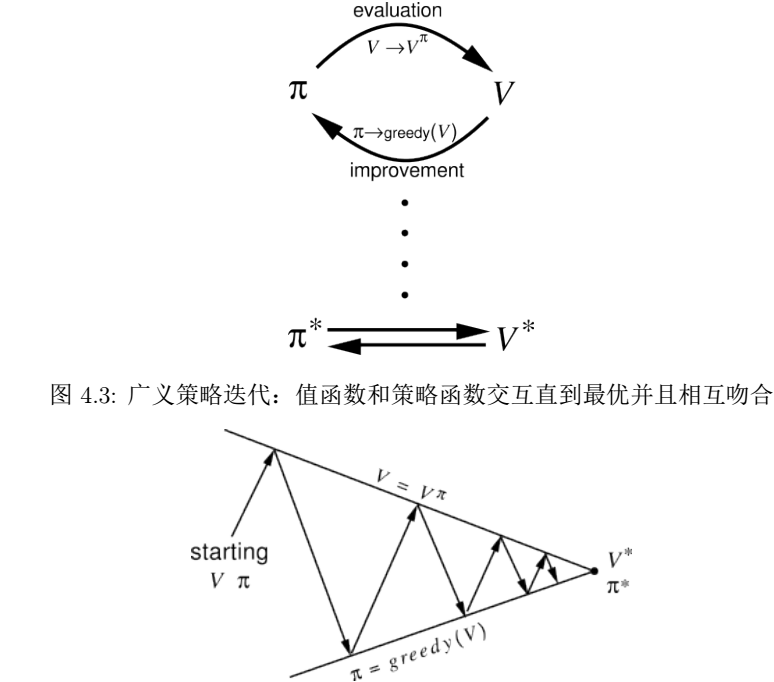


例子：状态1-100赌博决策

仅知道p=0.3(每局获胜概率)

策略：在各环境state下做出动作的选择。如在一个十字路口(状态)，往左是一策略，往右也是一策略

策略迭代过程



策略估计指的是给定策略下 值函数的迭代计算。策略改善指的是在给定值函数下 一个改善的策略。把这二者放在一起，获得了策略迭代和值迭代方法，这是DP算法中最流行的两种算法。在给定MDPs的完全先验知识的情况下，任何一种方法都能可靠的求解出最优策略和值函数。

Chapter 5

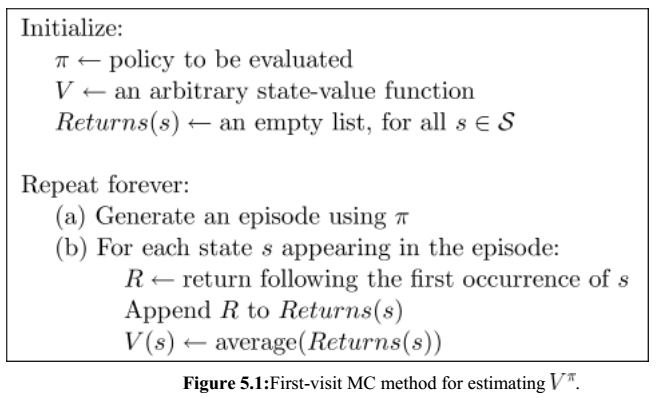
蒙特卡罗方法

和前面章节不同， 这里我们假设对环境知识一无所知。

蒙特卡罗方法只需要在线的或者模拟与环境的交互来抽样状态，动作，回报的序列。

蒙特卡罗方法解决增强学习问题的方法基于抽样回报的均值。当更多的回报被观测到时，平均值应该收敛到期望值，这是所有蒙特卡罗方法暗含的思想

5.1 蒙特卡罗策略评估



every-visit MC 方法是平均那些经过s的片段的回报来估计.first-visit MC 方法只是平均第一次访问s后的回报的平均。

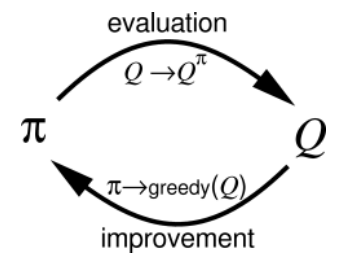


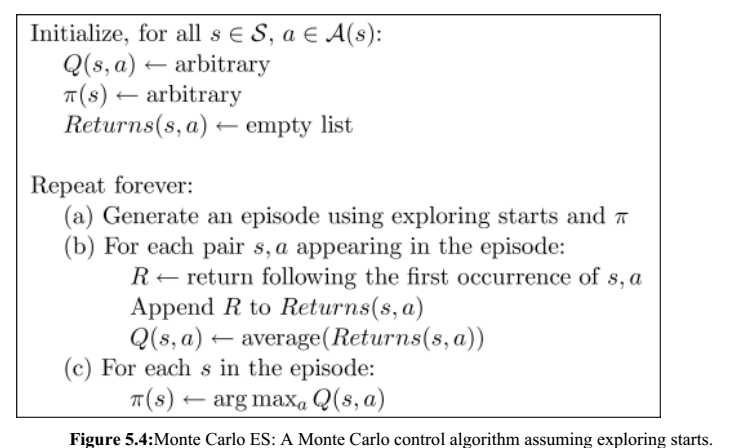
5.2 蒙特卡罗行为值函数的估计

如果模型无法获得，那么估计行为值比状态值更加有用。

5.3 蒙特卡罗控制

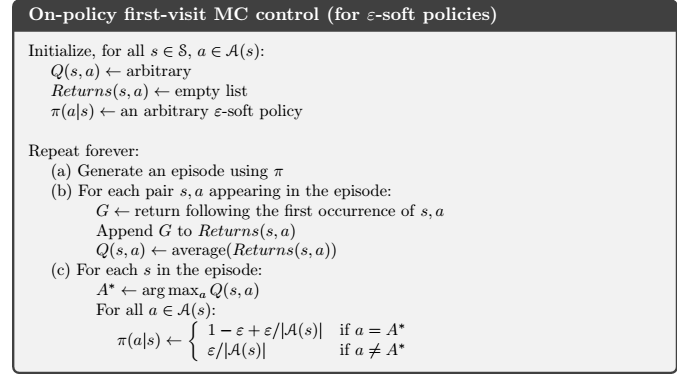
generalized policed iteration(GPI)





5.4 Monte Carlo Control without Exploring Starts

Monto Carlo on-policy 控制方法的大体思想还是GPI。但是没有探索初值假定的条件，

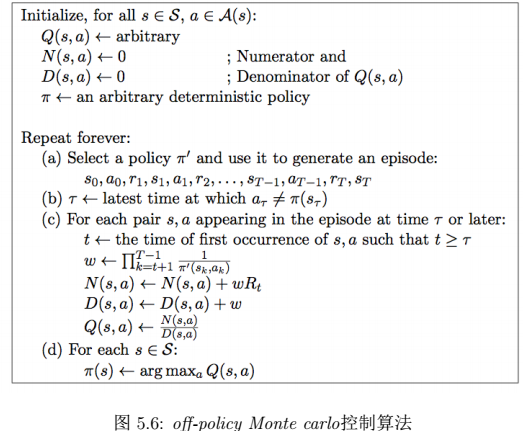


5.5 用一个策略的经验评价另一个策略

目前为止我们使用概率方法的思想是先给你一个策略，用这个策略产生一些片段，对这个片段进行策略评价，得到值函数，然后再更新策略。总的来说策略的更新基于值函数，值函数的获取基于抽样的片段。

当针对一个策略我们没有它产生的片段，却有另一个策略产生的片段，这种情况下我们依然能估计或

5.6 无策略的蒙特卡罗控制



**RL is a general-purpose framework for decision-making**

**RL is for an agent with the capacity to act**

**Each action influences the agent's future state**

**Success is measured by a scalar reward signal**

**Goal：select actions to maximise future reware**

**DL is a general-purpose framework for representation learning**

**Given an objective**

**Learn representation that is required to achieve objective**

**Directly from raw inputs**

**using minimal domain knowledge**

**Deep Reinforcement Learning: AI=RL + DL**

**We seek a single agent which can solve any human-level task**

**RL defines the objective**

**DL gives the mechanism**

**RL+DL = general intelligence**

强化学习系列：

      马尔科夫决策过程是强化学习的理论基础。不管我们是将强化学习应用于五子棋游戏、星际争霸还是机器人行走，我们都假设背后存在了一个马尔科夫决策过程。只不过有的时候我们知道马尔科夫决策过程所有信息(状态集合，动作集合，转移概率和奖励)，有的时候我们只知道部分信息 (状态集合和动作集合)，还有些时候马尔科夫决策过程的信息太大无法全部存储 (比如围棋的状态集合总数为 319×19 )。强化学习算法按照上述不同情况可以分为两种: 基于模型 (Model-based) 和非基于模型 (Model-free)。基于模型的强化学习算法是知道并可以存储所有马尔科夫决策过程信息，非基于模型的强化学习算法则需要自己探索未知的马尔科夫过程。

1. 策略迭代

      模型相关的强化学习算法主要有：策略迭代 (Policy Iteration) 和价值迭代 (Value Iteration)。策略迭代的主要思想是这样的。先随机初始化一个策略 (\pi\_0)，计算这个策略下每个状态的价值 (\pmb{v}\_0)；根据这些状态价值得到新策略 (\pi\_1)；计算新策略下每个状态的价值 (\pmb{v}\_1) ... 直到收敛。计算一个策略下每个状态的价值，被称为策略评估 (Policy Evaluation)；根据状态价值得到新策略，被称为策略改进 (Policy Improvement)。

2.价值迭代

      看完策略迭代，我们发现策略迭代计算起来挺麻烦的，代码实现起来也麻烦。我们能不能把策略评估和策略改进结合得更紧密一些。一个状态下选择动作一旦确定，立刻按照这个动作计算当前状态的价值。这个做法其实就是价值迭代

**模型无关的策略评价**

模型无关的策略评价是，不知道马尔科夫决策过程转移概率和奖励函数的情况下，计算一个策略的每一个状态价值。模型无关的策略评价主要有两种算法，一个是蒙特卡罗算法，另一个叫时差学习算法。

1. 蒙特卡罗算法

      一听到这个名字，我们就知道一个产生样本，通过样本计算状态价值的方法。首先，用当前策略探索产生一个完整的状态-动作-奖励序列。

s1,a1,r1,....,sk,ak,rk∼π (1)

然后，在序列第一次碰到或者每次碰到一个状态s时，计算其衰减奖励之后。

gs=rt+γrt+1+...+γk−trk (2)

最后更新状态价值

S(s)=S(s)+gs

N(s)=N(s)+1

v(s)=S(s)N(s) (3)

  蒙特卡罗算法能够有效地求解模型无关的策略评估，但也存在一些问题。有时我们面临的强化学习问题是持续不断的。比如没有停止指令时，飞行器控制要求不停地根据姿势风向等因素调整，持续保持平稳飞行。这时我们得不到一个完整状态-动作-奖励系列，因此蒙特卡罗算法不适用。为了解决这个问题，人们提出了时差学习算法 (Temperal Difference, TD)。时差学习算法利用马尔科夫性质，只利用了下一步信息。时差学习算法让系统按照策略指引进行探索，在探索每一步都进行状态价值的更新，更新公式如下所示。

v(s)=v(s)+α(r+γv(s′)−v(s))(4)

s 为当前状态， s′ 为下一步状态， r 是系统获得的奖励， α 是学习率， γ 是衰减因子。另外 r+γv(s′) 被称为时差目标 (TD target)， r+γv(s′)−v(s) 被称为时差误差 TDerror。时差学习算法的代码。

**模型无关的策略学习(改善)**

模型无关的策略学习，是在不知道马尔科夫决策过程的情况下学习到最优策略。模型无关的策略学习主要有三种算法: MC Control, SARSA 和 Q learning。

πϵ−greedy(s,a)={1−ϵ+ϵ/|A| a=argmaxaq(s,a)

ϵ|A| a≠argmaxaq(s,a)

蒙特卡罗控制

这个算法生成样本，然后根据样本计算状态-动作价值