[Reinforcement Learning 3](#_Toc16122)

[Chapter 1 3](#_Toc15663)

[Introduction 3](#_Toc18805)

[Reinforcement Learning 3](#_Toc30946)

[Elements of Reinforcement Learning 5](#_Toc29452)

[Limitations and Scope 6](#_Toc7084)

[An Extended Example: Tic-Tac-Toe 7](#_Toc2856)

[Summary 10](#_Toc13453)

[Chapter 2 11](#_Toc29948)

[Part I: Tabular Solution Methods 11](#_Toc20805)

[Multi-armed Bandits 11](#_Toc10587)

[A k-armed Bandit Problem 12](#_Toc18746)

[Action-value Methods 13](#_Toc25700)

[The 10-armed Testbed 14](#_Toc2305)

[Incremental Implementation 16](#_Toc17800)

[Tracking a Nonstationary Problem 17](#_Toc10956)

[Optimistic Initial Values 19](#_Toc7340)

[Upper-Confidence-Bound Action Selection 20](#_Toc19321)

[Gradient Bandit Algorithms 21](#_Toc13690)

[The Bandit Gradient Algorithm as Stochastic Gradient Ascent 22](#_Toc18733)

# Reinforcement Learning

## Chapter 1

### Introduction

当我们思考学习的本质时，我们首先想到的可能是通过与环境的互动来学习。当婴儿玩耍、挥动手臂或四处张望时，他没有明确的老师，但他确实与周围环境有直接的感觉运动联系。运用这种联系可以产生大量的信息，关于因果关系，关于行动的后果，以及为了实现目标应该做些什么。在我们的一生中，这样的互动无疑是了解我们的环境和我们自己的一个主要来源。无论我们是在学习开车，还是在与人交谈，我们都能敏锐地意识到我们所处的环境如何对我们所做的事情做出反应，并试图通过我们的行为来影响所发生的事情。从互动中学习几乎是所有学习和智力理论的基础。

在这本书中，我们探索了一种从互动中学习的计算方法。我们探索理想化的学习情境，评估各种学习方法的有效性，而不是直接理论化人类或动物的学习方式。也就是说，我们采用人工智能研究者或工程师的视角。我们探索能有效解决科学或经济问题的机器设计，通过数学分析或计算实验来评估设计。我们探索的方法，称为强化学习，比机器学习的其他方法更注重从互动中学习目标导向。

### Reinforcement Learning

强化学习是学习做什么，如何将情境映射到行动——从而最大化一个数字奖励信号。学习者不被告知该采取哪些行动，而是必须通过尝试发现哪些行动最有回报。在最有趣和最具挑战性的情况下，行动不仅会影响眼前的回报，还会影响下一种情况，进而影响所有后续的回报。这两个特征——试错搜索和延迟奖励是强化学习的两个最重要的特征。

强化学习，就像许多以ing结尾的话题一样。，例如机器学习和登山运动，同时也是一个问题，一种很好的解决问题的方法，以及研究这个问题及其解决方法的领域。用一个单一的名字来描述这三样东西很方便，但同时又必须在概念上把这三样东西分开，尤其是在强化学习中，区分问题和解决方法是非常重要的;未能作出这种区分是许多混淆的根源。

我们利用动力系统理论的思想，将强化学习问题形式化，具体地说，就是对未知马尔可夫决策过程的最优控制。具体细节第1章。介绍形式化必须等到第3章，但是基本思想是简单地捕捉学习代理所面临的真实问题的最重要的细节，随着时间的推移与环境进行交互，以实现目标。学习主体必须能够在一定程度上感知其环境的状态，并且必须能够采取影响该状态的行动。代理还必须有一个或多个与环境状态相关的目标。马尔可夫决策过程旨在以最简单的形式只包括感觉、行动和目标这三个方面，而不把它们中的任何一个小看。任何适合于解决这类问题的方法，我们都认为是一种强化学习方法。

强化学习不同于监督学习，是目前机器学习领域研究较多的一种学习方式。监督学习是从一个有知识的外部主管提供的一组带标签的例子中学习的。每个示例都是对一个情况的描述，以及系统应该对该情况采取的正确操作的规范(标签)，该规范通常用于标识该情况所属的类别。这类学习的目的是让系统推断或概括它的响应，以便在训练集中没有出现的情况下正确地工作。这是一种重要的学习，但仅凭这一点不足以从交互中学习。在交互问题中，想要获得既正确又能代表代理必须采取行动的所有情况的所需行为的示例通常是不切实际的。在未知的领域——人们期望学习是最有益的——一个代理人必须能够从自己的经验中学习。

强化学习也不同于机器学习研究人员所称的无监督学习，后者通常是寻找隐藏在未标记数据集合中的结构。监督学习和非监督学习这两个术语似乎对机器学习的各个方面进行了详尽的分类，但它们并没有这样做。尽管人们可能倾向于认为强化学习是一种无监督学习，因为它不依赖于正确行为的例子，但强化学习是试图最大化奖励信号，而不是试图寻找隐藏的结构。在一个agent的经验中发现结构对于强化学习当然是有用的，但是它本身并不能解决最大化奖励信号的强化学习问题。因此，我们认为强化学习是第三种机器学习范式，与监督学习和非监督学习以及其他范式并列。

在强化学习中出现的挑战之一，而不是在其他类型的学习中，是探索和开发之间的权衡。为了获得更多的奖励，强化学习代理必须更喜欢它过去尝试过的、在产生奖励方面有效的行为，但要发现这样的行为，它必须尝试它以前没有选择过的行为。为了获得奖励，代理人必须利用他已经经历过的，但是为了在未来做出更好的行动选择，他也必须提高产量。目前的困境是，不能只进行勘探或开发而不失败。代理人必须尝试各种各样的行动，并逐步偏爱那些看起来最好的行动。在随机任务中，每个动作都必须尝试多次，才能获得对预期回报的可靠估计。探索-利用的两难困境已经被数学家们研究了几十年，但一直没有得到解决。现在，我们只是简单地指出，在有监督和无监督的学习中，甚至不存在探索和开发之间平衡的整个问题，至少在它们最纯粹的形式中是这样。

强化学习的另一个关键特征是，它明确地考虑了目标导向的agent与不确定环境交互的整个问题。这与许多只考虑子问题而不考虑子问题如何融入全局的方法形成了对比。例如，我们已经提到，许多机器学习研究都与监督学习有关，而没有明确说明这种能力最终将如何发挥作用。其他研究人员已经发展出了具有一般目标的规划理论，但没有考虑规划在实时决策中的作用，也没有考虑规划所需的预测模型将从何而来。虽然这些方法已经产生了许多有用的结果，但是它们对孤立子问题的关注是一个重要的限制。

强化学习则采取相反的策略，从一个完整的、互动的、目标寻求的代理开始。所有的强化学习主体都有明确的目标，能够感知环境的各个方面，并能够选择与环境相关的行动。此外,它通常是认为从一开始,代理经营尽管重要其面临的环境不确定性时强化学习涉及到规划,它必须解决计划和实时行为选择之间的相互作用,以及如何获得和改善环境模型的问题,强化学习涉及监督学习,这样做的具体原因决定了哪些功能是关键的，哪些不是。为了使学习研究取得进展，必须对重要子问题进行分离和研究，但这些子问题应该是那些在完整的、交互的、目标寻求的agent中发挥明确作用的子问题，即使完整agent的所有细节还不能完全填充。

对于一个完整的、互动的、目标寻求的代理，我们并不总是指像一个完整的有机体或机器人那样的东西。这些都是明显的例子，但是一个完整的、交互式的、目标寻求的代理也可以是一个更大的行为系统的组成部分。在这种情况下，代理直接与较大系统的其余部分交互，并间接与较大系统的环境交互。一个简单的例子是一个代理，它监视机器人电池的充电水平，并向机器人的控制架构发送命令。这个智能体的环境是机器人的其余部分以及机器人的环境，我们必须超越最明显的智能体及其环境的例子来欣赏强化学习框架的普遍性。

现代强化学习最令人兴奋的一个方面是它与其他工程和科学学科的实质性和富有成效的互动。强化学习是人工智能和机器学习领域数十年以来的一个趋势的一部分，这一趋势趋向于与统计优化和其他数学学科更紧密地结合。例如，一些增强学习方法使用参数化逼近器学习的能力解决了运筹学和控制理论中经典的“维数咒”。更明显的是，强化学习也相互作用。与心理学和神经科学紧密合作，对双方都大有裨益。在所有的机器学习形式中，强化学习是最接近人类和其他动物学习的一种，而强化学习的许多核心算法最初都是受到生物学习系统的启发。强化学习也带来了回报，这既通过一种动物学习的心理模型，这种模型与一些经验数据更匹配，也通过一种对大脑奖励系统的部分无用模型。本书的主体发展了强化学习的概念，涉及工程和人工智能，与心理学和神经科学的联系，总结在第14和15章。

最后，强化学习也是人工智能回归简单一般原则这一更大趋势的一部分。自20世纪60年代末以来，许多人工智能研究人员认为，没有普遍的原则可以被发现，而是由于拥有大量特殊目的的技巧、程序和启发。有人说，如果我们能把足够多的相关信息输入机器，比如一百万或十亿，那么它就会变得智能。基于一般原理的方法，如搜索或学习，被描述为“弱方法”，而基于特定知识的方法被称为“强方法”。这种观点在今天仍然很普遍，但并不占主导地位。从我们的观点来看，现在还为时过早:为寻求普遍原则而作出的努力太少，以致不能得出没有普遍原则的结论。现代人工智能包括寻找学习、搜索和决策的一般原则的大量研究，以及试图整合大量领域知识。目前还不清楚钟摆会摆回多远，但强化学习研究肯定是摆回更简单、更少的人工智能一般原理的一部分。

### Elements of Reinforcement Learning

除了agent和环境之外，我们还可以确定增强学习系统的四个主要子元素:策略、奖励信号、价值函数，以及环境模型(可选)。

策略定义了学习代理在给定时间内的行为方式。粗略地说，一个政策就是从环境的感知状态到在这些状态下要采取的行动的映射。它对应于心理学中所谓的一系列刺激反应规则或关联。在某些情况下，策略可能是一个简单的函数或查找表，而在另一些情况下，它可能涉及大量的计算，比如搜索过程。策略是强化学习代理的核心，因为它本身就足以决定行为。一般来说，政策可能是随机的。

奖励信号定义了强化学习问题的目标。在每一个时间步骤中，环境因素都会向强化学习代理发送一个名为奖励的数字。代理人的唯一目标是最大化其长期获得的总回报。因此，奖励信号定义了对代理来说什么是好事件，什么是坏事件。在生物系统中，我们可能认为奖励类似于快乐或痛苦的经历。它们是代理所面临的问题的直接和决定性特征。奖励信号是改变政策的主要依据;如果政策所选择的行为之后奖励较低，那么政策可能会改变，在未来的情况下选择其他一些行为。一般来说，奖励信号可能是环境状态和所采取行动的随机函数。

奖励信号表示的是直接意义上的好，而值函数则指定的是长期意义上的好。粗略地说，一个状态的价值是一个代理从这个状态开始，在未来可以期望累积的报酬总额。虽然奖励决定了环境国家的直接的、内在的可取性，但价值观念表明，在考虑到可能随后出现的国家以及这些国家所提供的奖励之后，国家的长期可取性。例如，一个状态可能总是产生较低的即时回报，但仍然具有较高的价值，因为它经常被其他产生较高回报的状态所遵循。反之亦然。打个人类的比方，回报多少有点像快乐(如果高)和痛苦(如果低)，而价值则对应于对我们的环境处于特定状态时的高兴或不高兴程度的更精确和更有远见的判断。这样表达，我们希望能够清楚地看到，值函数形式化了一个基本的、熟悉的概念。

从某种意义上说，奖励是首要的，而价值观，作为对奖励的预测，是次要的。没有奖励就没有价值，评估价值的唯一目的就是获得更多的奖励。然而，在做决定和评估决定时，我们最关心的是价值观。行动选择是基于价值判断做出的。我们寻求的行动能带来最高价值的状态，而不是最高的回报，因为从长远来看，这些行动能给我们带来最大的回报。不幸的是，确定价值要比确定奖励难得多。奖励基本上是由环境直接给予的，但是价值必须通过一个代理在其整个生命周期中进行的一系列观察来评估和重新评估。事实上，我们所考虑的几乎所有强化学习算法中最重要的组成部分是一种有效估计值的方法。价值评估的核心作用可以说是我们在过去几十年里学到的关于强化学习的最重要的东西。

一些强化学习系统的第四个也是最后一个要素是环境模型。这是一种模仿环境行为的东西，或者更广泛地说，它允许对环境将如何行为进行推断。例如，给定一个状态和行为，模型可能预测下一个状态和下一个奖励的结果。模型用于计划，我们指的是在实际经历之前考虑可能的未来情况来决定行动过程的任何方式。用于解决使用模型和规划的强化学习问题的方法称为基于模型的方法，而不是显式试验和无模型的简单方法。错误学习者——几乎与计划背道而驰。在第8章中，我们探讨了强化第1章。介绍同时通过反复试验学习的学习系统，学习环境模型，并使用模型进行规划。现代强化学习的范围从低级的反复试验学习到高级的深思熟虑的计划。

### Limitations and Scope

从前面的讨论可以看出，强化学习在很大程度上依赖于状态作为策略和价值函数的输入，以及作为模型的输入和输出的概念。非正式地说，我们可以把状态看作是在特定时间向代理传递某种“环境如何”的感觉的信号。我们在这里使用状态的形式化定义是由第三章中介绍的马尔可夫决策过程框架给出的。然而，更广泛地说，我们鼓励读者遵循非正式的含义，并把状态看作任何可用的信息。向代理询问其环境。实际上，我们假设状态信号是由一些预处理系统产生的，这些预处理系统名义上是代理环境的一部分。在本书中，我们不讨论构造、改变或学习状态信号的问题。我们采取这一方法不是因为我们认为国家代表权是不重要的，而是为了充分集中注意决策问题。换句话说，我们主要关心的不是设计状态信号，而是决定作为任何可用状态信号的函数采取什么行动。(我们会在17.3节的最后一章简要介绍状态设计和构造。)

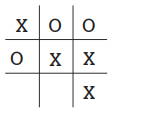
我们在本书中考虑的大多数强化学习方法都是围绕估值函数来构建的，但并不是严格必须这样做才能解决强化学习问题。例如，遗传算法、遗传规划、模拟退火等优化方法被用来处理强化学习问题，而从未使用过值函数。这些方法评估许多非学习代理的“终身”行为，每个代理使用不同的策略与环境交互，并选择那些能够获得最多回报的代理。我们之所以称这些进化方法为进化方法，是因为它们的运作方式类似于生物进化产生具有熟练行为的生物体的方式，即使这些生物体在它们的个体生命周期中没有学会如何去努力。如果策略的空间足够小，或者可以构造好策略，使好的策略很常见或容易找到，或者有大量时间用于搜索，那么演化方法可能是有效的。此外，进化方法在学习代理无法感知其环境的完整状态的问题上具有优势。

我们在本书中考虑的大多数强化学习方法都是围绕估值函数来构建的，但并不是严格必须这样做才能解决强化学习问题。例如，遗传算法、遗传规划、模拟退火等优化方法被用来处理强化学习问题，而从未使用过值函数。这些方法评估许多非学习代理的“终身”行为，每个代理使用不同的策略与环境交互，并选择那些能够获得最多回报的代理。我们之所以称这些进化方法为进化方法，是因为它们的运作方式类似于生物进化产生具有熟练行为的生物体的方式，即使这些生物体在它们的个体生命周期中没有学会如何去努力。如果策略的空间足够小，或者可以构造好策略，使好的策略很常见或容易找到，或者有大量时间用于搜索，那么演化方法可能是有效的。此外，进化方法在学习代理无法感知其环境的完整状态的问题上具有优势。

然而，我们确实包括了一些方法，像进化方法一样，不吸引价值函数。这些方法搜索由一组数值参数定义的策略空间。他们估计应该调整参数的方向，以便最迅速地改进策略的性能。然而，与进化方法不同的是，它们在代理与其环境交互时产生这些估计，因此可以利用个体行为交互的细节。像这样的方法在许多问题中被证明是有用的，一些最简单的强化学习方法就属于这一类(见第13章)。最后。然而，这种类型的最佳方法倾向于以某种形式包含值函数。

### An Extended Example: Tic-Tac-Toe

为了说明强化学习的一般概念，并将其与其他方法进行比较，接下来我们将更详细地考虑一个示例。



想想我们熟悉的孩子的井字游戏。两人轮流在一块3乘3的棋盘上玩。一个玩家玩Xs，另一个玩,操作系统，直到一名玩家在水平方向连续放置三个标记获胜垂直地，或斜地，就像X玩家在游戏中所显示的那样正确的。如果棋盘上没有一个玩家连续得到三个，则x | x比赛不分胜负。因为一个熟练的棋手可以打得不输，让我们一起来吧假设我们面对的是一个不完美的玩家，他的游戏是x有时是错误的，让我们赢了。事实上，就目前而言，让我们考虑到平局和损失对我们同样有害。我们如何构建一个球员会发现对手的缺点，并学会最大化自己获胜的机会?

虽然这是一个简单的问题，但它不能轻易地通过经典技术得到令人满意的解决。例如，经典的博弈论中的“极大极小”解在这里是不正确的，因为它假定对手以一种特定的方式进行博弈。例如，一个极小极大的玩家永远不会到达一个它可能会输的游戏状态，即使事实上它总是因为对手的错误游戏而从那个状态中获胜。序列决策问题的经典优化方法，如动态规划，可以计算任意对手的最优解，但需要输入该对手的完整规范，包括对手在每个棋盘状态下每一步移动的概率。让我们假定，这一资料不是这个问题的先验资料，因为它不是绝大多数实际关心的问题的先验资料。另一方面。这样的信息可以从经验中估计出来，在这种情况下是通过与对手进行许多比赛来获得的。在这个问题上，最好的方法是首先学习对手的行为模型，在一定的置信度范围内，然后运用动态规划计算出给定近似对手模型的最优解。最后，这与我们在本书后面研究的一些强化收益方法并没有太大的不同。

将进化方法应用于这一问题，将直接搜索可能的政策空间，寻找一个具有高胜算的对手。这里的策略是一个规则，它告诉玩家在3×3的棋盘上Xs and Os的每一种可能的配置在游戏的每一种状态下应该采取什么行动。对于所考虑的每一种策略，其获胜概率的估计将通过与对手进行一定数量的博弈来获得。然后，这种评估将指导下一步考虑哪些政策或哪些政策。典型的进化方法是在策略空间上爬坡，依次生成和评估策略，以获得增量改进。或者，也许可以使用一种遗传学风格的算法来维护和评估策略的总体。可以应用数百种不同的优化方法。

这是如何处理井字游戏的问题将使用一个方法利用价值函数。首先，我们设置一个数字表，每个数字代表游戏的每个可能状态。每个数字都是这个状态下我们赢的概率的最新估计。我们把这个估计值看作状态值，整个表就是学习值函数。状态值高于国家B比状态B或被认为是“更好”,如果当前的概率的估计我们赢得来自一个高于B .假设我们总是玩x,然后连续所有州有三个Xs获胜的概率是1,因为我们已经赢了。类似地，对于所有连续三个Os的状态，或者是“被填满”的状态，正确的概率是0，因为我们无法从它们中获胜。我们把所有其他州的初始值设为0.5，表示我们有50%的几率获胜。

我们和对手打了很多比赛。为了选择我们的走法，我们检查每一个可能的走法所产生的状态(黑板上的每个空格对应一个)，并查找它们的当前状态第1章。介绍表中的值。大多数情况下，我们贪得无偿失地移动，选择移动到最有价值的状态，也就是说，具有最高的估计获胜概率。然而，偶尔我们也会从其他动作中随机选择。这些被称为探索性的举动，因为它们使我们体验到一些我们可能从未见过的状态。在游戏过程中所做和考虑的一系列动作可以如图1.1所示。

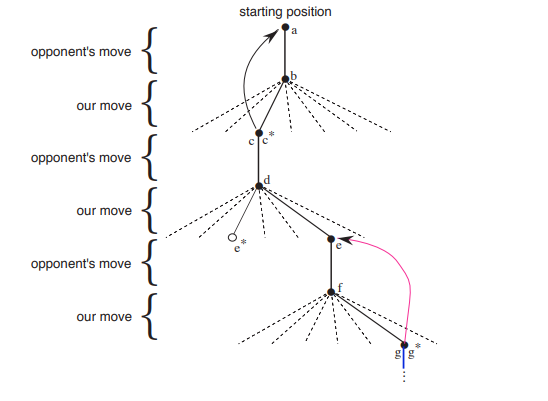


图1.1:井字游戏的一系列动作。实线表示游戏中所采取的动作;虚线表示我们(我们的强化学习玩家)考虑过但没有采取的行动。我们的第二步是探索性的一步，这意味着即使另一个兄弟姐妹的动作，导致e”，排名更高，我们还是采取了这一步。探索性的移动不会导致任何学习，但是我们的其他每个移动都会导致更新，就像文本中详细描述的那样，通过曲线箭头将估估值从树的较晚的节点向上移动到较早的节点一样。

当我们在玩游戏时，我们改变了我们在游戏中所处的状态的价值观。我们试图使他们更准确地估计获胜的可能性。来做到这一点。我们“备份”每个贪婪移动到移动前状态后的状态值，如图1.1中的箭头所示。更精确地说，将更新早期状态的当前值，使其更接近于后期状态的值。这可以通过将较早状态的值移动到较晚状态值的一小部分来实现。如果我们令s表示贪婪移动前的状态，s表示移动后的状态，则对s估计值的更新记为V(s)

搜狗截图20190525113908

其中a是一个很小的正分数，称为步长参数，它影响学习速度。这个更新规则是时间差学习方法的一个例子，之所以称为时间差学习方法，是因为它的更改是基于两个不同时间的估计值之间的差异V(s')-V(s)。

上述方法在这项任务中表现得很好。例如，如果步长参数计随着时间适当减小，那么对于任何固定的对手，该方法收敛于给定玩家最优策略下每个状态的真实获胜概率。此外，所采取的步骤(探索性步骤除外)实际上是针对这个(不完美的)对手的最优步骤。换句话说，该方法收敛于与该对手博弈的最优策略。如果步长参数没有随着时间的推移一直减小到零，那么这个玩家也能很好地对抗那些缓慢改变他们游戏方式的对手。

这个例子说明了演化方法和学习值函数的方法之间的区别。为了评估策略，进化方法将策略固定下来，并与对手进行多次博弈，或者使用对手的模型模拟多次博弈。获胜的频率给出了该策略获胜概率的无偏估计，并可用于指导下一个策略选择。但是，每次政策变更都是在许多游戏之后才会做出的，并且只使用每个游戏的最终结果:忽略游戏期间发生的事情。例如，如果玩家赢了，那么他在游戏中的所有行为都将得到奖励，而不受特定的动作对获胜的影响。甚至可以把功劳归于从未发生过的动作!价值函数的方法。相比之下。允许对各个州进行评估。最后。进化方法和价值函数方法都是在策略空间中进行搜索，而学习价值函数则是利用博弈过程中获得的信息。

这个简单的例子说明了强化学习方法的一些关键特征。首先，在与环境的互动中强调学习，在这种情况下是与对手玩家互动。其次，有一个明确的目标，正确的行为需要计划或远见，考虑到一个人的选择的延迟影响。例如，简单的强化学习玩家会学习为短视的对手设置多步陷阱。强化学习解决方案的一个显著特点是，它可以在不使用对手模型、不对未来状态和动作的可能序列进行显式搜索的情况下，实现规划和展望的效果。

虽然这个例子说明了强化学习的一些关键特征，但它是如此简单，可能会给人一种印象，即强化学习比实际情况更有限。虽然tic tac-toe是一个两个人的游戏，但强化学习也适用于没有外部对手的情况，即“与自然对抗的游戏”。强化学习也不局限于将行为分解为不同的情节的问题，比如井字游戏。只有在每一集的结尾才有奖励。当行为无限期地继续下去，并且在任何时候都可以得到不同数额的奖励时，它同样适用。强化学习也适用于那些甚至不能分解为离散时间步长的问题，比如井字游戏。一般原则也适用于连续时间问题，尽管理论变得更加复杂，我们在这个介绍性的处理中省略了它。

井字游戏有一个相对较小的有限状态集，而强化学习可以在状态集非常大，甚至是无限时使用。例如，Gerry Tesauro(1992,1995)将上述算法与人工神经网络相结合，学习下约1020个状态的双陆棋。有了这么多的状态，就不可能经历超过其中一小部分的状态。Tesauro的程序学会了比以前任何程序都玩得好得多，现在的水平已经达到了世界上最好的人类玩家的水平(见第16章)。神经网络为程序提供了从经验中归纳的能力，因此在新的状态下，它根据从过去面临的类似状态中保存的信息(由其网络决定)选择移动。强化学习系统在处理如此大的状态集问题时的表现如何，与它从过去经验中总结出的恰当程度密切相关。正是在这个角色中，我们最需要带强化学习的监督学习方法。要做到这一点，神经网络和深度学习不是唯一的方法，也不一定是最好的方法。

在这个井字游戏的例子中，学习一开始并没有超出游戏规则的先验知识，但是强化学习绝不需要一个学习和智力的白板视图。相反，先验信息可以以多种方式整合到强化学习中，这对于有效学习是至关重要的。我们也在井字游戏的例子中获得了真实的状态，而强化学习也可以应用于部分状态隐藏的情况。或者当不同的状态在学习者看来是相同的。

最后，井字游戏者能够向前看，并知道每一个可能的移动会导致什么状态。要做到这一点，它必须拥有一个游戏模型，让它能够“考虑”自己的环境将如何变化，以应对自己可能永远不会采取的行动。许多问题都是这样的，但在另一些问题中，甚至缺乏行动效果的短期模型。强化学习可以应用于任何一种情况。不需要模型，但是如果模型可用或可以学习，则可以很容易地使用(第8章)。

另一方面。有一些强化学习方法根本不需要任何环境模型。无模型支持向量机系统甚至不能考虑它们的环境将如何响应单个操作而发生变化。从这个意义上讲，井字游戏是无模式的:它没有任何对手的模型。因为模型必须相当精确才能有用。当解决问题的真正瓶颈是难以建立足够精确的环境模型时，无模型方法比更复杂的方法具有优势。无模型方法是也基于模型方法的重要构建块。在本书中，在讨论如何将它们用作更复杂的基于模型的方法的组件之前，我们将用几章讨论无模型方法。

强化学习可以用于系统的高水平和低水平。尽管井字游戏玩家只学习了游戏的基本动作，但没有什么能阻止强化学习在更高的层次上工作，在更高的层次上，每个“动作”本身都可能是一种可能的复杂解决问题的方法的应用。在分层学习系统中，强化学习可以在多个层次上同时进行。

### Summary

强化学习是一种计算方法，用于理解和自动化目标导向的学习和决策。它与其他计算方法的区别在于，它强调从与环境的直接交互中学到的知识，而不依赖于模范的监督或环境的完整模型。在我们看来，强化学习是第一个认真解决计算问题的领域，当学习从与一个1.7。强化学习的早期历史11环境才能实现长期目标。

强化学习使用马尔可夫决策过程的形式化框架，根据状态、行为和奖励来定义学习代理与其环境之间的相互作用。该框架旨在作为一种简单的方法来表示人工智能问题的基本特征。这些特征包括因果感、不确定性和非决定论以及明确目标的存在感。

价值和价值函数的概念是我们在本书中考虑的大多数强化学习方法的关键特征。我们认为，在策略空间中，值函数对于有效搜索是非常重要的。值函数的使用将强化学习方法与在整个策略的标量评估指导下直接在策略空间中搜索的进化方法区分开来。

## Chapter 2

### Part I: Tabular Solution Methods

在本书的这一部分中，我们几乎用最简单的形式描述了增强学习算法的所有核心思想:状态和操作空间足够小，可以将近似值函数表示为数组或表。在这种情况下，这些方法往往可以找到精确的解，也就是说，它们往往可以找到精确的最优值函数和最优策略。这与本书下一部分描述的近似方法形成了对比。它们只能找到近似解，但反过来又能有效地应用于更大的问题 本书这一部分的第一章描述了在只有一个状态的特殊情况下的强化学习问题的解决方法，称为班迪特问题。第二章描述了我们在本书其余部分中处理的一般问题公式，以及它的主要思想，包括贝尔曼方程和值函数。

接下来的三章描述了求解有限马尔可夫决策问题的三种基本方法:动态规划法、蒙特卡罗方法。而时间差学习的每一种方法都有其优缺点。动态规划方法是很好的数学发展，但需要一个完整和准确的环境模型。蒙特卡罗方法不需要模型，而且概念简单，但不适合步进增量计算。最后，时间差方法不需要模型，是完全增量的，但分析起来比较复杂。这些方法在效率和收敛速度方面也有许多不同之处。

剩下的两章描述了如何将这三类方法组合起来以获得它们各自的最佳特性。在第一章中，我们描述了蒙特卡罗方法的优点如何通过合格性跟踪与时间差方法的优点相结合。在本书的最后一章中，我们展示了如何将时间差学习方法与模型学习和规划方法(如动态规划)相结合，从而完整而统一地解决表格强化学习问题。

### Multi-armed Bandits

强化学习与其他类型学习最重要的区别在于，它使用训练信息来评估所采取的行动，而不是通过给出正确的行动来进行指导。这就产生了对积极探索的需求，对良好行为的明确搜索。纯粹的评价反馈表明所采取的行动有多好。但这不是最好的还是最坏的行动。另一方面，纯粹的指导性反馈表明要采取的正确行动，与实际采取的行动无关。这种反馈是监督学习的基础，包括模式分类、人工神经网络和系统辨识的大部分内容。在纯形式下，这两种反馈是截然不同的:评估性反馈完全依赖于所采取的行动，而指导性反馈独立于所采取的行动。

在本章中，我们研究了强化学习的评估方面，在一个简单的设置，一个不涉及学习在一个以上的情况下行动。这种非联想的设置是大多数先前涉及评估反馈的工作已经完成的一个。它避免了完全强化学习问题的复杂性。研究这个案例使我们能够最清楚地看到评估性反馈与指导性反馈的不同之处，以及如何与之结合。

我们所探讨的这个特殊的非联想、评价反馈问题是k-armed bandit问题的一个简单版本。我们用这个问题来介绍一些基本的学习方法，这些方法我们将在后面的章节中进行扩展，以应用于完全强化学习问题。在本章的最后，我们通过讨论当强盗问题变成联想问题时，也就是当在更多的情况下采取行动时，会发生什么，从而向完全强化学习问题又迈进了一步。

### A k-armed Bandit Problem

考虑下面的学习问题。你反复面临着k种不同的操作或行为的选择。每次选择之后，你都会从一个稳定的概率分布中得到一个数值奖励，这个分布取决于你选择的行为。你的目标是在一段时间内最大化预期的总回报，例如，超过1000个行动选择，或时间步骤。

这是k-armed bandit问题的原始形式，它的名称类似于老虎机“独臂强盗”，除了它有k个杠杆而不是一个。每个动作的选择就像一个剧本和老虎机的杠杆，而奖励是中奖的回报。通过重复行动选择你要把你的行动集中在最好的杠杆上，从而使你的收益最大化。另一个类比是医生在一系列的实验治疗之间做出选择病人。每一个行动都是对一种治疗方法的选择，每一个奖励都是患者的生存或幸福。今天，“强盗问题”一词有时被用来概括这个问题如上所述，但在本书中，我们只是用它来指代这个简单的例子。在我们的k-armed bandit问题中，每一个k动作都有一个预期或平均回报该操作被选中;我们称这个为这个动作的值。我们表示准时选择的动作步骤t为At，对应的奖励为Rt。则任意动作a的值为q∗(a)，是给定a被选中的期望奖励:

搜狗截图20190525115552

如果你知道每个动作的价值，那么解决k-armed bandit问题就变得微不足道了:你总是选择值最高的操作。我们假设您不知道操作值当然，尽管你可能有估计。我们表示一次行动a的估计值步骤t为Qt(a)。我们希望Qt(a)接近q∗(a)。

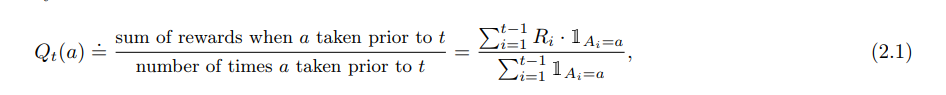
如果您维护操作值的估计值，那么在任何时候，至少有一个操作的估计值最大。我们称之为贪婪行为。当您选择其中一个操作时，我们说您正在利用您当前对操作值的知识。如果您选择一个非贪婪操作，那么我们说您正在进探索，因为这使您能够改进对非贪婪操作值的估计。开发是正确的事情，以最大限度地提高预期回报的一步，但从长远来看，探索可能产生更大的总回报。例如，假设一个贪婪操作的值是确定的，而其他几个操作的估计值几乎一样好，但存在很大的不确定性。不确定性是这样的，至少有一个其他的行为可能比贪婪行为更好，但你不知道哪一个。如果您有很多时间来进行操作选择。然后，最好探索非贪婪行为，并发现其中哪些行为优于贪婪行为。在探索过程中，奖励在短期内较低，但在长期内较高，因为在你发现了更好的行为之后，你可以多次利用它们。由于任何单一的行动选择都不可能同时进行探索和开发，因此常常提到探索和开发之间的“冲突”。

在任何具体情况下，是探索还是利用以复杂的方式更好地取决于估计数、不确定性和剩余步骤数的精确值。对于k-armed bandit的特定数学公式及其相关问题，有许多复杂的勘探开发平衡方法。然而，这些方法中的大多数都对平稳性和先验知识做出了强有力的假设，这些假设在应用程序中或者在我们在后面的章节中考虑的完全强化学习问题中要么被违反要么无法验证。当这些方法的理论假设不适用时，它们的最优性或有界损失的保证就没有多大的安慰作用。

在这本书中，我们不担心以一种复杂的方式平衡勘探和开发;我们只关心如何平衡它们。在这一章中，我们提出了几个简单的平衡问题的方法，并表明他们的工作比总是利用的方法好得多。在强化学习中，平衡探索和开发的需要是一个独特的挑战;我们版本的k-armed bandit问题的简单性使我们能够以一种特别清晰的形式展示这一点。

### Action-value Methods

我们通过更仔细地研究一些简单的方法来评估操作的值，并使用这些方法来做出操作选择决策。记住，一个行为的真正价值是选择该行为时的平均回报。一个自然的估计方法是平均回报。



其中1predicate表示如果预测为真则为1，如果为假则为0的随机变量。如果分母为0，则我们将Qt(a)定义为某个默认值，如0。作为分母趋于无穷，根据大数定律，Qt(a)收敛到q∗(a)。我们称之为样本均值由于每个估计值是相关样本的平均值，所以估计动作值的方法奖励。当然，这只是评估行为价值的一种方法，不一定是最好的方法。不过，现在让我们继续使用这个简单的估算方法，并转向如何进行估算的问题。评估可能用于选择操作。最简单的动作选择规则是选择估计值最高的动作之一，也就是说，前面部分定义的贪婪操作之一。如果不止一个贪婪然后，以某种任意的方式(也许是随机的)在它们之间进行选择。我们写这个

贪心行为选择方法为



其中argmaxa表示操作a，后面的表达式将最大化(再一次,有联系破碎的任意)。贪婪的行为选择总是利用当前的知识来最大化即时的奖励;它根本不花时间抽样那些明显较差的行为，看看它们是否真的可能更好。一个简单的选择是在大多数时候表现得贪婪，但偶尔，说与小概率ε,而不是选择随机从所有的行动以同样的概率,独立的行动价值评估。我们使用这个near-greedy行为选择规则ε-greedy调用方法。这些方法的一个优点是，在极限条件下，随着步骤数的增加，每一步操作将被采样无限次，从而确保所有Qt(a)都收敛到q∗(a)。这当然意味着选择最优行动的概率收敛到大于1−ε,接近确定性。然而，这些只是渐近的保证，并没有说明什么方法的实际效果。

练习2.1在e -贪心行为选择中，对于两个行为且ε = 0.5的情况，贪心行为被选择的概率是多少?

练习2.2:强盗示例考虑一个k-armed Bandit问题，其中k = 4个动作，记为1 2 3 4。考虑申请这个问题一个强盗算法使用ε-greedy行动选择,对于所有的a，样本平均动作值估计值，Q1(a)的初始估计值为0动作和奖励的顺序是A1 = 1， R1 = 1 ，A2 = 2， R2 = 1， A3 = 2， R3 = 2， A4 = 2， R4 = 2，A5 = 3 ，R5 = 0。在这些时间步骤ε的情况可能发生,导致一个动作随机选择。这肯定发生在哪个时间步骤上?时间的脚步就能做到这一点可能发生呢?

### The 10-armed Testbed

粗略评估的相对有效性贪婪和ε-greedy方法,我们在一套测试问题上比较数值。这是一组随机生成的2000个 k-armed bandit问题，k = 10。对于每个bandit问题，如图2.1所示，操作值q∗(a), a = 1, . . . , 10，根据均值为0，方差为1的正态(高斯)分布进行选择。然后，当一种学习方法应用于该问题时在第t步选择的操作时，实际的奖励Rt将从一个平均q∗(At)和方差1的正态分布中选择。这些分布在图2.1中以灰色显示。我们将这组测试任务称为10臂测试床。对于任何一种学习方法，我们都可以测量它的性能和行为，因为当应用于某个强盗问题时，它会随着超过1000个时间步长的经验而改进。这就构成了一次运行。在2000次独立运行中，每次运行都有不同的强盗问题，我们得到了学习算法平均行为的度量。

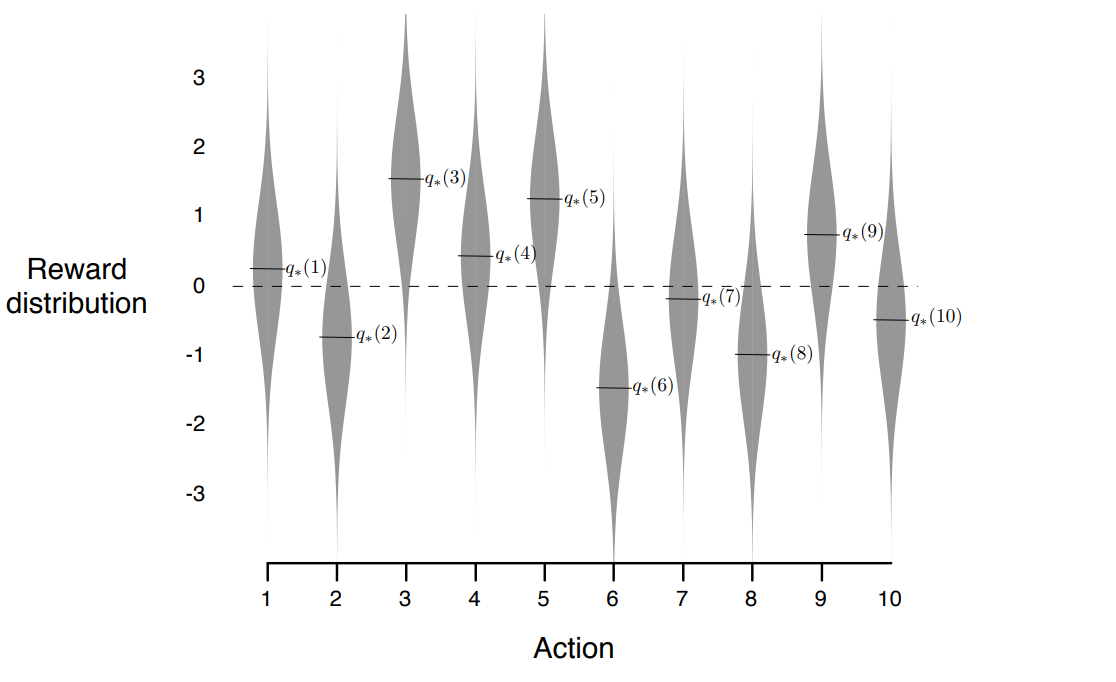
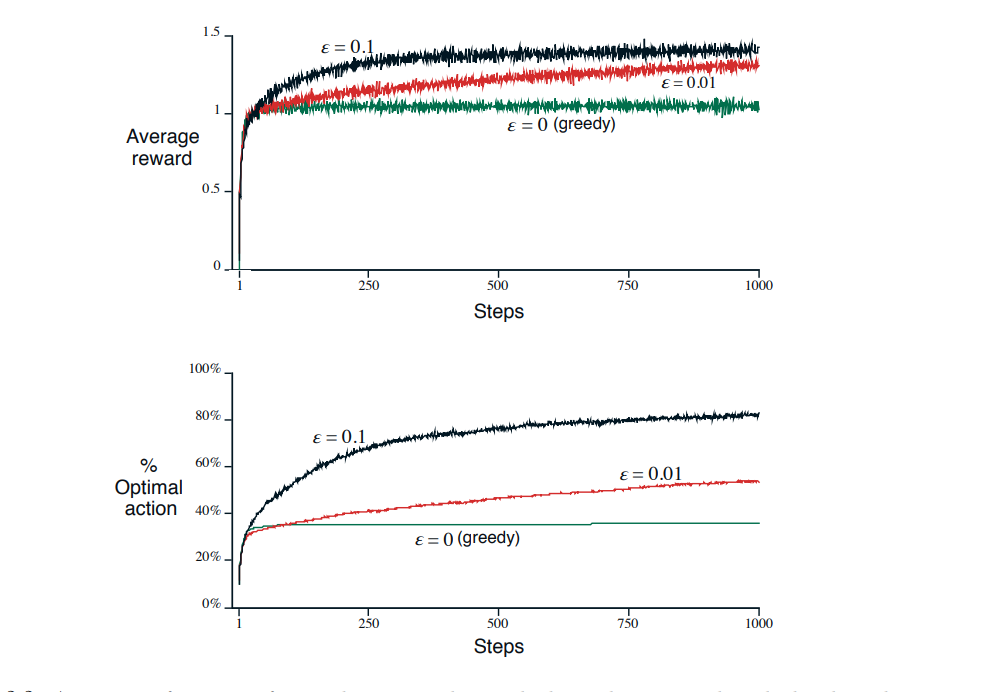


图2.1:一个来自10个武装试验台的强盗问题示例。10个状态中的每一个的真正值q∗(a)根据均值为零的正态分布和单位方差选择动作，然后进行实际操作奖励根据平均q∗(a)单位方差正态分布选择，如这些灰色所示分布。

任何一种学习方法，我们都可以测量它的性能和行为，因为当它应用于强盗问题时，它会随着超过1000个时间步骤的经验而改进。这就构成了一次运行。在2000次独立运行中，每次运行都有不同的强盗问题，我们得到了学习算法平均行为的度量。

图2.2在10臂实验台上，将贪心方法与两种ε-greedy方法(ε = 0.01 and ε = 0.1)进行比较，如上所述。所有方法均采用样本平均法形成其行为价值估计。上面的图表显示了经验带来的预期回报的增加。贪心法在初始阶段的改进速度略快于其他方法，但随后趋于平稳。它每步只获得大约1的奖励，而最好的可能是大约1.55在这个实验台上。从长远来看，贪心方法的性能要差得多，因为它经常陷入执行次优操作的困境。下图显示，贪心方法只在大约三分之一的任务中找到了最优操作。在其他三分之二。其最优行为的初始样本令人失望，而且它再也没有回归。贪婪电子的方法最终表现得更好，因为它们继续探索并提高了识别最优行为的机会。ε = 0.1法探索的次数较多，通常较早发现最优动作，但其选择该动作的次数从未超过91%。ε = 0.01方法改进较慢，但最终在两个性能指标上都优于ε = 0.01方法，如图所示。随着时间的推移，也可以减少ε，以尽可能地得到最佳的高值和低值。



Fioure 2.2: e-贪心动作值方法在10臂实验台上的平均性能。这些数据平均超过2000次运行与不同的强盗问题。所有的方法都使用样本平均值作为其行为价值的估计。

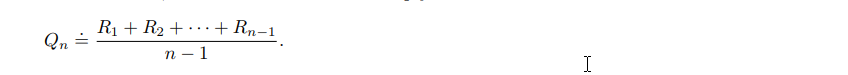
ε-greedy在贪婪的优势取决于任务的方法。例如，假设奖励方差更大，比如10而不是1。与奖励吵着需要更多的探索发现最佳的行动,和ε-greedy方法应该表现更好的相对于贪婪的方法。另一方面，如果奖励方差为零，那么贪婪方法在尝试一次后就会知道每个动作的真实价值。在这种情况下，贪心方法实际上可能执行得最好，因为它很快就会找到最优的操作，然后永远不会探索。但即使是在确定性的情况下，如果我们弱化其他一些假设，探索也是有很大优势的。例如，假设强盗任务是非平稳的，也就是说，动作的真实值随着时间而变化。在这种情况下，甚至在确定性的情况下也需要探索，以确保其中一个非贪婪行为没有变得比贪婪行为更好。我们将在接下来的几章中看到，非平稳性是强化学习中最常见的情况。即使底层任务是平稳和确定的，学习者也会面临一组强盗式的决策任务，随着学习的进行和代理策略的变化，这些任务也会随着时间的推移而变化。强化学习需要探索和开发之间的平衡。

练习2.3在图2.2所示的比较中，哪种方法在长期运行中表现最好就累积奖励和选择最佳行动的概率而言?这样会好多少呢是什么?定量地表达你的答案。

### Incremental Implementation

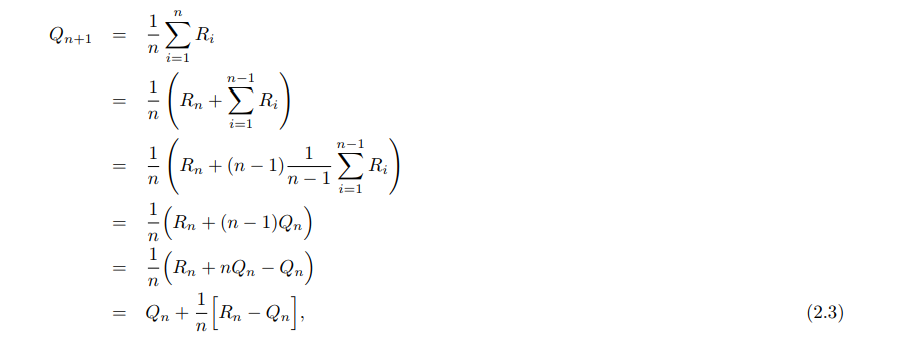
到目前为止，我们所讨论的行动价值方法都是将行动价值估计为观察到的奖励的样本平均值。现在我们要讨论的问题是如何以一种计算效率高的方式计算这些平均值。在特定的。通过固定的内存和固定的步长计算。

为了简化符号，我们只关注一个动作。现在让Ri表示第i次选择这个动作后得到的奖励，让Qn表示它被选择n - 1次后的行为值估计值，现在我们可以简单地写成

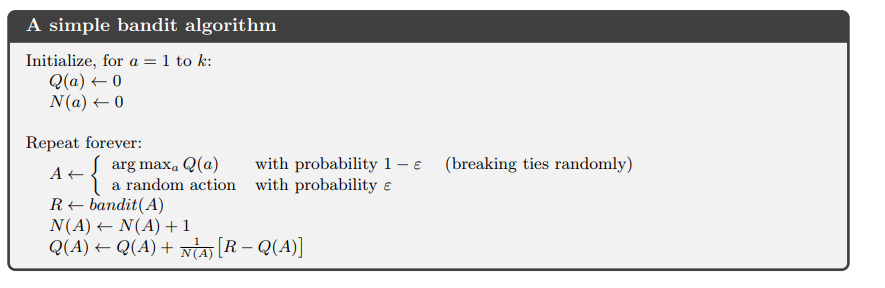


最明显的实现是保存所有奖励的记录，然后执行此操作在需要估计值时进行计算。但是，如果这样做了，那么内存和随着回报的增加，计算需求会随着时间的推移而增长。每个额外的奖励是否需要额外的内存来存储它，并需要额外的计算来计算分子。

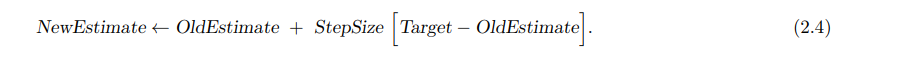
正如您可能怀疑的那样，这并不是真正必要的。设计增量公式是很容易的更新平均与小，不断计算需要处理每一个新的奖励。鉴于Qn第n个奖励，Rn，所有n个奖励的新平均值可以计算出来



这对于n = 1成立，对于任意Q1得到Q2 = R1。这个实现需要内存仅对Qn和n，且仅对每个新奖励进行小计算(2.3)。的伪代码完成强盗算法使用增量计算样本平均值和ε-greedy行动选择显示在下一页的框中。假定函数bandit(a)执行一个动作并返回相应的奖励。



更新规则(2.3)的形式在本书中经常出现。一般形式是



表达式[Target−OldEstimate]是估计中的一个误差。它是通过采取步骤来减少”的目标。“目标被认为是一个理想的行动方向，尽管它可能有噪声。例如，在上面的例子中，目标是第n个奖励。

注意，上述增量方法中使用的步长参数(StepSize)随着时间步长而变化。在处理动作a的第n个奖励时，该方法使用步长参数。在这本书中，我们用表示步长参数，或者更一般地用

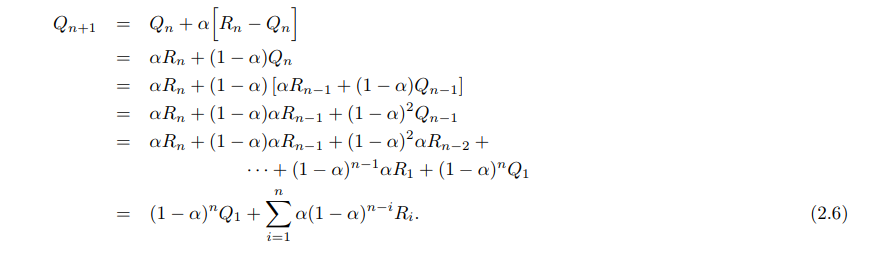
表示步长参数。当 = 时，我们有时会使用非正式的简写形式= ，从而隐式地保留了n对操作的依赖关系，就像我们在本节中所做的那样。

### Tracking a Nonstationary Problem

目前所讨论的平均方法适用于平稳的强盗问题，即报酬概率不随时间变化的强盗问题。如前所述，我们经常遇到强化学习问题，这些问题实际上是非平稳的。在这种情况下，重视近期的奖励比重视长期的奖励更有意义。最常用的方法之一是使用一个常量步长参数。例如，用于更新n-1个过去奖励的平均Qn的增量更新规则(2.3)被修改为



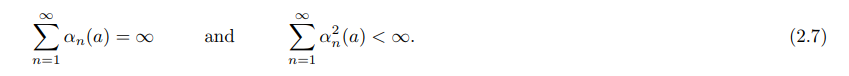
在步长参数α∈(0,1)是恒定的。这导致Qn+1是过去奖励的加权平均值，第一季度的初始估计值为:



我们称之为加权平均因为权重之和是。你可以自己检查一下。注意权重，，奖励R取决于多少奖励之前，n-i，是观察到的。小于1。因此Ri的权重;随着干预奖励的增加而减少。实际上，权重是根据的指数指数衰减的。(如果= 0。然后所有的权重都在最后一个奖励Rn上，因为习惯上 = 1)因此，这有时称为响应加权平均。(**exponential recency-weighted average**)

有时可以方便地一步一步地改变步长参数。让表示步长参数，用于处理第n次选择动作a后收到的奖励指出,选择结果得到了保证收敛的样本均值法真正的行为价值是由大数定律决定的。当然，并不是所有人都能保证收敛选择的序列{}。随机逼近理论中的一个著名结果给出了保证与概率1收敛所需的条件:

(a, b]作为一个集合表示a和b之间的实区间，包括b，但不包括a。表示。



第一个条件是确保步骤足够大，最终能够克服任何初始条件或随机功能。第二个条件保证步骤最终变得足够小，以确保收敛。注意，这两个收敛条件都满足样本平均情况，，但不满足常数步长参数的情况。在后一种情况下，第二个条件没有得到满足，这表明估计数从来没有完全收敛，而是继续随最近收到的奖励而变化。正如我们前面提到的，这在非平稳环境中是很有必要的，而在强化学习中，那些实际上是非平稳的问题是最常见的。此外，满足条件(2.7)的步长参数序列收敛速度非常慢，或者需要进行大量的调整才能获得满意的收敛速度。虽然满足这些收敛条件的步长参数序列在理论工作中经常使用，但在应用和实证研究中却很少使用。

练习2.4如果步长参数、αn不是常数,然后估计Qn加权以前获得奖励的平均值与(2.6)给出的权重不同。是什么在一般情况下，与(2.6)类似的对每个先前奖励的加权，其顺序为步长参数?

练习2.5(编程)设计并进行实验来证明这些困难样本平均法用于非平稳问题。使用10-armed testbed的修改版本其中所有的q∗(a) 一开始是相等的，然后进行独立的随机漫步(比如添加)一个正态分布增量，平均为0，每一个上的所有q∗(a)的标准差为0.01一步)。使用样例平均值增量地为action-value方法准备图2.2所示的图计算,另一个行为价值的方法使用一个常数步长参数,α= 0.1。用ε= 0.1长跑，比如10000步。

### Optimistic Initial Values

到目前为止，我们讨论的所有方法都在一定程度上依赖于最初的行动价值估计值Q1(a)。在统计学的语言中，这些方法由于最初的估计而有偏差。对于样本平均方法，一旦所有操作至少被选择一次，偏差就会消失，但是对于常数的方法，偏差是永久性的，但是随着时间的推移，偏差会减小，如(2.6)所示。在实践中，这种偏见通常不是问题，有时可能非常有用。缺点是，初始估计值实际上变成了用户必须选择的一组参数，即使只是为了将它们全部设置为零。好处是，它们提供了一种简单的方法来提供一些关于预期的奖励水平的先验知识。

初始动作值也可以作为鼓励探索的简单方法。假设,而不是最初的行动值设置为零,当我们在10-armed testbed,我们都设置为+ 5记得,在这个问题上选择从一个正态分布均值为0,方差为1。 + 5的初始估计过于乐观。但是这种乐观鼓励探索行动价值的方法。无论最初选择哪种行动，奖励都低于最初的估计;学习者转向其他动作。对自己得到的回报感到“失望”。结果是，在值估计收敛之前，所有操作都尝试了几次。即使总是选择贪婪行为，系统也会进行大量的探索。

图2.3显示了使用Q1(a) =的贪心方法在**10-armed testbed**上的性能+ 5。相比之下,也显示是一个ε-greedy方法Q1 (a) = 0。起初，乐观方法的性能较差，因为它探索得更多，但最终它的性能更好，因为它的探索随着时间的推移而减少。我们称这种技术为鼓励勘探的乐观开端值。我们认为这是一个简单的技巧，可以非常有效地解决平稳问题，但它是远远的从一个普遍有用的方法到鼓励探索。例如，它不太适合非平稳问题，因为它的探索动力本质上是暂时的。如果任务发生变化，这种方法带来了对探索的新需求，但却无济于事。事实上，任何方法都可以在初始条件上采用任何特殊的方法都不大可能对一般的非平稳情况有所帮助。时间的开始只有一次，所以我们不应该太关注它。这种批评同样适用于样本平均法，它也把时间的开始当作一个特殊的事件，平均所有后续奖励的同等权重。然而，所有这些方法都非常简单，在实践中，它们中的一个——或者它们的简单组合——通常是足够的。在其余的地方在本书中，我们经常使用这些简单的探索技术。

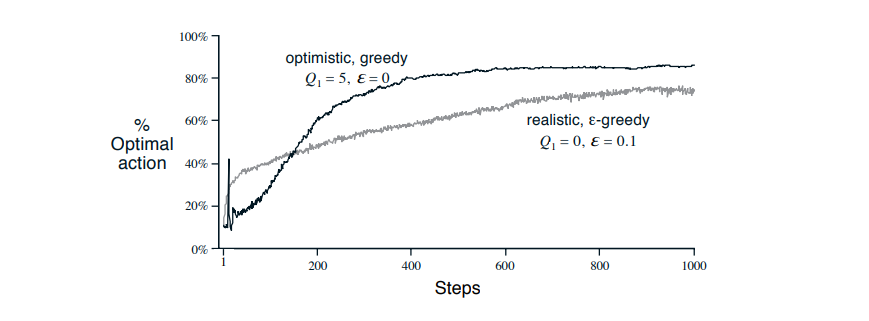


图2.3:乐观的初始行动价值估计对10-armed testbed的影响。这两种方法都使用固定步长参数,。

乐观的方法表现得更差，因为它探索得更多，但最终它表现得更好，因为它的探索随着时间而减少。我们把这种鼓励勘探的技术称为乐观的初始价值。我们认为这是一个简单的技巧，可以非常有效地解决平稳问题，但它远远不是鼓励探索的一种普遍有用的方法。例如，它不太适合非平稳问题，因为它的探索动力本质上是暂时的。如果任务发生变化。这种方法带来了对探索的新需求，但却无济于事。实际上，任何以任何特殊方式关注初始条件的方法都不太可能帮助处理一般的非平稳情况。时间的开始只有一次，所以我们不应该太关注它。这种批评同样适用于样本平均法，该方法也将时间开始视为一个特殊事件，平均所有后续奖励的权重相同。然而，所有这些方法都非常简单。在实践中，一个或几个简单的组合就足够了。在他书的其余部分，我们经常使用这些简单的探索技术。

练习2.6:图2.3中所示的结果应该非常可靠，因为他们是平均超过2000个人，随机选择10个武装土匪任务。那么，为什么会出现这种情况呢振荡和峰值在曲线的早期部分为乐观方法?换句话说，什么平均而言，在特定的早期步骤中，这种方法的性能是更好还是更差?

### Upper-Confidence-Bound Action Selection

探索是必要的，因为行动价值的准确性总是存在不确定性。贪婪的行为是目前看起来最好的。但其他一些行动可能会更好。贪婪操作选择强制尝试非贪婪操作。但要不假思索，不要偏爱那些近乎贪婪或特别不确定的人。根据非贪婪行为的实际最优性，在非贪婪行为中进行选择。考虑到他们的估计与最大值的接近程度以及这些估计中的不确定性。一种有效的方法是根据



其中ln t表示t的自然对数(e≈2.71828)为了等于t)， Nt(a)表示动作a在时间t之前被选中的次数(分母在(2.1)中)，c > 0控制探索的程度。如果Nt(a) = 0，那么a被认为是一个最大化的行为。

这种上置信界(UCB)行为选择的思想是，平方根项是对估计a值的不确定性或方差的度量。因此，最大值是行动a可能的真实值的上界，而c决定了置信度。每次选择a，不确定性就会减少:Nt(a) 增加，当它出现在分母上时，不确定性项就会减少。另一方面。每次选择a以外的操作时。t增加，但是N (a)不变:因为t出现在分子中，所以不确定性估计增加。使用自然对数意味着增长会随着时间变得更小，但是是无界的:最终将选择所有操作，但是操作的估计值更低。或者已经被频繁选择。将随着时间的推移而减少选择的频率。

在10-armed testbed上使用UCB的结果如图2.4所示。UCB通常表现很好，如下所示,但是更困难比ε-greedy超越更一般的强盗强化学习设置考虑在这本书的其余部分。一个困难是处理不稳定问题;将需要比第2.5节所述方法更为复杂的方法。另一个困难是处理大的状态空间，特别是在使用函数逼近as时开发在本书的第二部分。在这些更高级的设置中，UCB操作选择的思想是通常是不实际的。

### Gradient Bandit Algorithms

到目前为止，在本章中，我们已经考虑了估算行动值并使用这些估算值的方法选择操作。这通常是一种好方法，但不是唯一可行的方法。在这一节中我们考虑学习每个动作a的数值偏好，这表示Ht(a)。越大偏好，这个行为被采取的次数越多，但是偏好没有解释奖励。只有一种行为相对于另一种行为的偏好才是重要的;如果我们把1000加到所有的偏好对行动概率没有影响，而行动概率是由软最大值决定的分布(即。，吉布斯分布或玻尔兹曼分布):



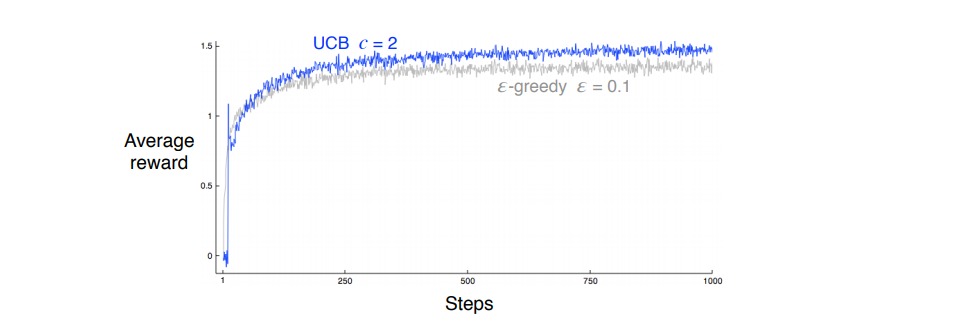
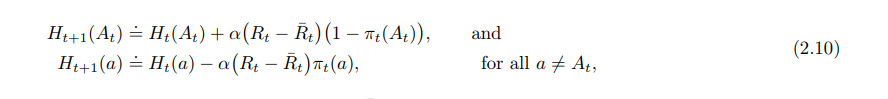


图2.4:10臂实验台上UCB动作选择的平均性能。如图所示，UCB一般执行比ε-greedy选择动作,除了在第一个k步骤,当它选择随机的未被尝试的行动。

这里，我们还引入了一个有用的新符号，πt(a)，表示在t时刻采取行动a的概率。最初，所有首选项都是相同的(例如，H1(a) =0，对于所有a)，以便所有行动都具有相同的被选中的概率。

练习2.7表明，在两个动作的情况下，软最大值分布与给定值相同在统计学和人工神经网络中常用的logistic(逻辑函数)或sigmoid函数。基于随机梯度上升的思想，有一个自然的学习算法。在每一步中，在选择action At并接受reward Rt后，参数更新如下:



其中α> 0是一个步长参数,R¯t∈R是和之前所有奖励的平均值包括时间t，如第2.4节所述，时间t可以增量计算(如有，则可按第2.5节计算)问题是非平稳的)。R¯t项作为比较奖励的基线。如果奖励高于基线，那么未来接受的概率就会增加，如果奖励低于基线，那么概率就会降低。属性中移动非选定的操作相反的决定。

图2.5显示了梯度bandit算法在10-armed testbed 在一个变量上的结果根据均值为+4的正态分布，选择了哪些真实的期望报酬而不是以前的零(和单位方差)。所有回报的上升是绝对的对梯度强盗算法没有影响，因为奖励基线项，这是即时的适应新水平。但是如果忽略基线(也就是说，如果R¯t是常数0在(2.10)中，性能会显著下降，如图所示

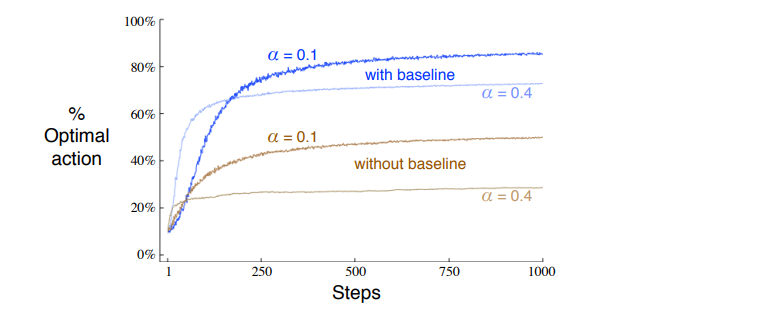
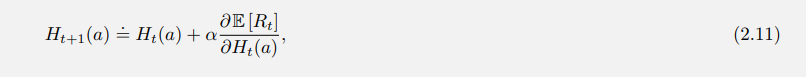


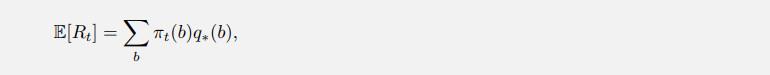
图2.5:带和不带奖励基线的梯度强盗算法的平均性能当 选择接近+4而不是接近0时，将使用10-armed testbed

### The Bandit Gradient Algorithm as Stochastic Gradient Ascent

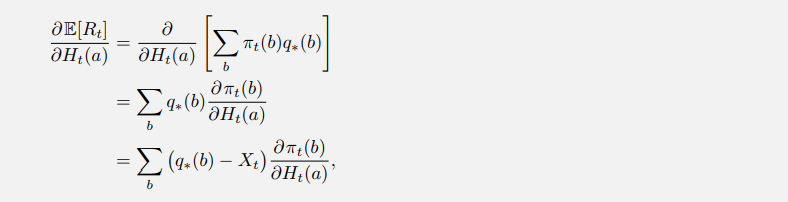
通过将梯度bandit算法理解为a，可以更深入地了解梯度bandit算法梯度上升的随机近似。在精确的梯度上升中，每个偏好Ht(a)将会随着性能的增加而成正比地增加



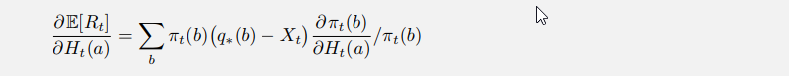
这里的绩效衡量是预期的奖励:



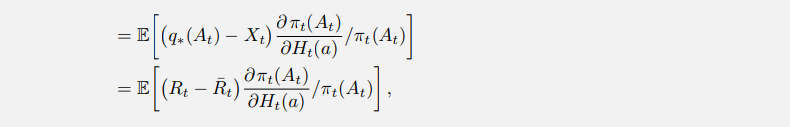
增量效应的度量是这个性能度量对偏好的偏导数。当然，在我们的例子中不可能完全实现梯度上升，因为假设我们不知道q∗(b)，但实际上我们的算法(2.10)的更新在期望值上等于(2.11)，使得算法成为随机梯度上升的一个实例。计算表明，这只需要开始微积分，但采取几个步骤。首先，我们来仔细看看确切的性能梯度:



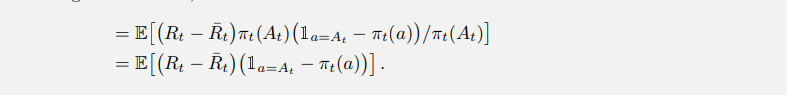
其中Xt可以是任何不依赖于b的标量，我们可以把它包括在这里，因为梯度对所有的作用求和为零。当Ht(a)发生变化时，概率有上升也有下降，但变化的和必须为零，因为总和概率必须保持为1。



方程现在是期望的形式，对所有可能的值b求和随机变量At，然后乘以取这些值的概率。因此



,这里我们选择了Xt = R并且将Rt替换为，这是允许的，因为。因为Rt(给定At)与任何东西都不相关。我们将建立，其中如果a=b,1a=b定义为1，否则为0。假设现在

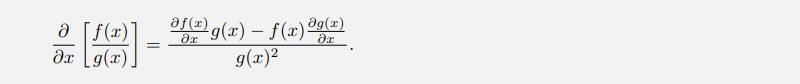


回想一下，我们的计划是将性能梯度作为我们可以在每个步骤中采样的期望。就像我们刚才做的那样，每一步的更新都与样本成比例。用上述期望的一个样本来代替(2.11)的收益率

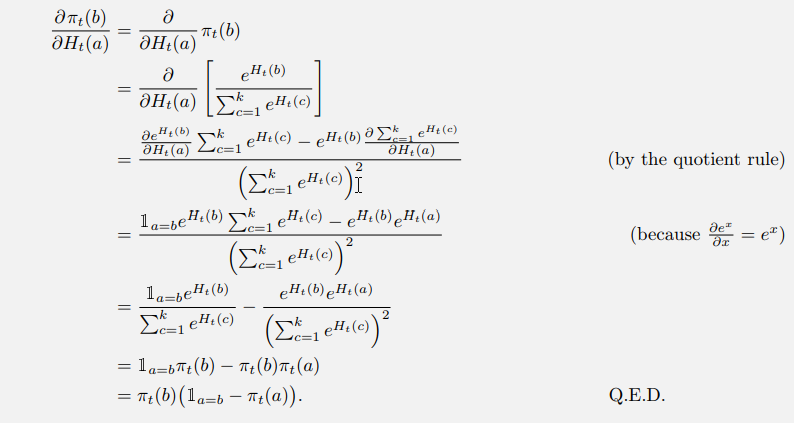
搜狗截图20190525160713

您可能认为它等价于我们的原始算法(2.10)。

因此，只需要证明，正如我们假设的那样。回忆起导数的标准除法定则：



用这个，我们可以写



我们刚刚证明了梯度强盗算法的期望更新等于期望报酬的梯度，因此该算法是随机梯度上升的一个实例。这保证了算法具有鲁棒收敛性。

注意，我们不需要奖励基线的任何属性，除了它不依赖于所选择的操作。例如，我们可以把它设为0或1000，算法仍然是随机梯度上升的一个例子。基线的选择不影响算法的预期更新，但会影响更新的方差，从而影响收敛速度(如图2.5所示)。选择它作为奖励的平均值可能不是最好的，但它很简单，在实践中效果很好。

### Associative Search (Contextual Bandit

到目前为止，在本章中，我们只考虑了非联想任务，即不需要将不同的操作与不同的情况相关联的任务。在这些任务中，学习者要么试图在任务静止时找到一个最佳动作，要么试图在任务非静止时跟踪最佳动作随时间的变化。然而，在一般的强化学习任务中存在不止一种情境，目标是学习策略:从情境到实际行动的映射。最好在这种情况下。为了为整个问题做好准备，我们简要讨论了将非关联任务扩展到关联设置的最简单方法。

例如，假设有几个不同的k-armed bandit任务，并且在每个步骤中都随机选择其中一个任务。因此，强盗任务每一步都在随机变化。在您看来，这是一个单一的、非平稳的k-armed bandit任务，其真实动作值随步骤随机变化。您可以尝试使用本章描述的方法之一来处理非平稳性。但除非真正的行动价值观慢慢改变。这些方法不会很有效。然而，现在假设，当为您选择了一个强盗任务时，您提供了一些关于它的身份(但不是它的动作值)的独特线索。也许你面对的是现实老虎机，当它改变其动作值时，改变其显示的颜色。现在，您可以学习一个策略，将您看到的颜色表示的每个任务与面对该任务时应该采取的最佳行动相关联——例如，如果是红色，则选择arm 1;如果是绿色，选择arm 2。有了正确的策略，您通常可以做得比在没有任何信息来区分强盗任务和其他任务时好得多。

这是一个关联搜索任务的例子。之所以这么叫，是因为它既包括尝试和错误的学习，以寻找最佳的行动，并把这些行动与最佳的情况联系起来。在文献中，联想搜索任务现在常被称为虚拟盗匪。联想搜索任务介于K-武装强盗问题和完全强化学习问题之间。它们就像完全强化学习问题，因为它们涉及到学习策略。但是就像我们版本的k-armed bandit问题一样，每个动作只影响即时的回报。如果行为被允许影响下一个情境和奖励，那么我们就有了完全强化学习问题。我们将在下一章介绍这个问题，并在本书的其余部分考虑它的影响。

练习2.8假设vou面临一个两臂强盗任务，其真实动作值随时间步长随机变化。具体地说，假设对于任意时间步长，动作1和动作2的真实值分别为0.1和0.2(概率为0.5)(案例A)， 0.9和0.8(概率为0.5)(案例A)B).如果你在任何时候都不能说出你所面临的情况，你对成功的最大期望是什么?现在，假设在每一步中都被告知您面临的是情形A还是情形B(尽管您仍然不知道真正的操作值)。这是一个关联搜索任务。在这项任务中，你对成功的最大期望是什么？

### Summary

在这一章中，我们提出了几种平衡勘探和开发的简单方法。e -贪心方法随机选择一小部分时间，而UCB方法则在很小的范围内选择威慑，但在每一步都微妙地偏向于迄今为止收到较少样本的行动，从而实现探索。梯度强盗算法估计的不是动作值。但行动偏好。并且使用软最大值分布，以分级的、概率的方式支持更多的首选操作。初始化估计值的简单权宜之计会导致甚至贪婪的方法也要进行大量的探索。

很自然地，我们会问这些方法中哪一种是最好的。虽然这是一个很难回答的问题，但我们可以在本章中使用的10个测试台上运行它们，并比较它们的性能。复杂的是它们都有一个参数;为了得到有意义的比较，我们必须把它们的性能看作参数的函数。到目前为止，我们的图表已经显示了每个算法和参数设置的学习过程，以生成该算法和参数设置的学习曲线。如果我们为所有算法和所有参数设置绘制学习曲线，那么图就会太复杂、太拥挤，无法进行清晰的比较。相反，我们总结一个完整的学习曲线的平均值超过1000步;这个值与学习曲线下的面积成正比。图2.6显示了本章中各种bandit算法的度量，每个算法都是其自身参数的函数，在x轴上以单一尺度显示。这种图称为参数研究。注意，参数值受两个因子的影响而变化，并以对数尺度表示。还注意了各算法性能的特征倒u型;所有算法他们的参数在中间值为时性能最佳。既不太大也不太小。在评估一个方法时，我们不应该只关注它在最佳参数设置时的表现。以及它对参数值有多敏感。所有这些算法都是相当不敏感的，在大约一个数量级变化的参数值范围内执行得很好。总的来说，在这个问题上，UCB的表现最好。

尽管它们很简单，但我们认为本章介绍的方法可以被认为是最先进的。还有更复杂的方法，但是它们的复杂性和假设使得它们对于我们真正关注的完全强化学习问题不切实际。从第5章开始，我们提出了解决全强化学习问题的学习方法，其中部分使用了本章探索的简单方法。

虽然本章所探讨的简单方法可能是我们目前所能做的最好的办法，但它们远不能完全令人满意地解决勘探和开发之间的平衡问题。

在k-武装土匪问题中，平衡勘探与开发的一种研究比较深入的方法是计算称为Gittins指数的特殊函数。这些方法为比本文考虑的更一般的一类强盗问题提供了一个最优解，但是这种方法假定可能的问题的先验分布是已知的。不幸的是，这种方法的理论和计算的可操作性似乎都不能推广到我们在书的其余部分考虑的完全强化学习问题。

贝叶斯方法假设动作值的初始分布是已知的，然后在每一步之后精确地更新分布(假设真正的动作值是平稳的)。一般来说，更新计算可能非常复杂，但是对于某些特殊的分布(称为共轭先验)，它们很简单。一种可能性是在每一步中根据最佳操作的后验概率选择操作。这种方法，有时称为后验抽样或汤普森抽样，通常执行类似于我们在本章中介绍的最好的无分布方法。

在巴伐利亚背景下，甚至可以计算出勘探和开发之间的最佳平衡。对于任何可能的动作，都可以计算出每一个可能的立即向外的概率以及由此产生的后验分布。这种不断发展的分布成为问题的信息状态。给定一个范围，比如说1000步，一个人可以考虑所有可能的行动，所有可能的结果奖励，所有可能的下一步行动，所有的下一步奖励，等等。根据这些假设，可以确定每个可能事件链的回报和概率。人们只需要挑选最好的。但是可能性之树长得非常快;甚至只有两个行为和两个奖励，这棵树就会有22000片叶子。准确地执行这个庞大的计算通常是不可行的，但也许它可以被有效地近似。该方法将有效地把强盗问题转化为完全强化收益问题的一个实例。最后，我们可能能够使用近似强化学习方法，如本书第二部分中介绍的方法来接近这个最优解。但这是一个研究的主题，超出了这本介绍性书籍的范围。

练习2.9(编程)为练习2.5中概述的非平稳情况制作一个类似于图2.6的图。包含常数步长ε-greedy，a=0.1。Use运行200,000步，作为每个算法和参数设置的性能度量，使用最后100,000步的平均奖励。