人工智能研究杂志 4 (1996) 237-285 9/95 提交;发表于 5/96

强化学习:调查

Leslie Pack Kaelbling lpk@cs.brown.edu
Michael L. Littman mlittman@cs.brown.edu
Computer Science Department, Box 1910, Brown University
Providence, RI 02912-1910 USA

安德鲁·W·摩尔 awm@cs.cmu.edu
Smith Hall l 221, Carnegie Mel lon University, 5000 Forbes Avenue
美国宾夕法尼亚州匹兹堡 15213

抽象的

本文从计算机科学的角度调查了强化学习领域。它是为熟悉机器学习的研究人员而编写的。总结了该领域的历史基础和广泛选择的当前工作。

强化学习是代理通过与动态环境的试错交互来学习行为所面临的问题。这里描述的工作与心理学工作有相似之处,但在细节和 \reinforcement 这个词的使用上有很大不同。”该论文讨论了强化学习的核心问题,包括交易、探索和利用,建立马尔可夫决策理论、延迟强化学习、构建经验模型以加速学习、利用泛化和层次结构、处理隐藏状态。强化学习的方法。

一、简介

强化学习可以追溯到控制论的早期,并在统计学、心理学、神经科学和计算机科学领域工作。在过去的 5 到 10 年里,它引起了机器学习和人工智能社区的兴趣迅速增加。

它的承诺是一种通过奖励和惩罚来编程代理的方式,而无需指定如何完成任务。但是,要实现这一承诺存在巨大的计算障碍。

本文从计算机科学的角度调查了强化学习的历史基础和当前的一些工作。我们对该领域进行了高层次的概述,并尝试了一些特定的方法。当然,不可能提及该领域的所有重要工作。这不应被视为详尽无遗的说明。

强化学习是代理必须通过与动态环境的试错交互来学习行为所面临的问题。这里描述的工作与心理学中的同名工作有很强的家族相似性,但在细节和\强化这个词的使用上存在很大差异。”它被恰当地认为是一类问题,而不是一组问题技巧。

解决强化学习问题有两种主要策略。第一个是在行为空间中搜索,以找到在环境中表现良好的行为。

这种方法已被遗传算法和遗传编程的工作所采用,

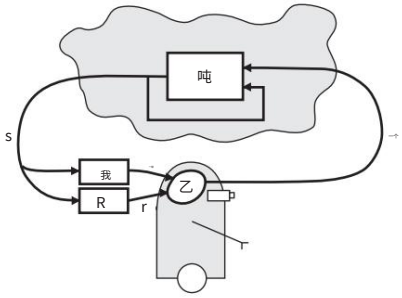

图 1:标准强化学习模型。

以及一些更新颖的搜索技术（Schmidhuber,1996）。第二个是使用统计技术和动态规划方法来估计在世界各国采取行动的效用。本文几乎完全致力于第二组技术,因为它们利用了强化学习问题的特殊结构,而这在一般优化问题中是不可用的。目前尚不清楚哪种方法在何种情况下是最好的。

本节的其余部分致力于建立符号并描述基本的强化学习模型。第 2 节解释了探索和利用之间的权衡,并为强化学习问题的最基本案例提供了一些解决方案,其中我们希望最大化即时奖励。第 3 节考虑了更普遍的问题,其中奖励可以从对获得奖励至关重要的行动中及时延迟。第 4 节考虑了一些经典的无模型算法,用于从延迟奖励中进行强化学习:自适应启发式批评家、TD(·) 和 Q-learning。第 5 节演示了一系列算法,这些算法对代理在环境中的实际操作步骤之间可以执行的计算量很敏感。泛化|主流机器学习研究的基石|具有极大地帮助强化学习的潜力,如第 6 节所述。第 7 节考虑了当代理无法完全感知环境状态时出现的问题。第 8 节列出了强化学习的一些成功应用。

最后,第 9 节以一些关于重要的开放问题和强化学习的未来的推测作为结尾。

1.1 强化学习模型

在标准的强化学习模型中,代理通过感知和动作连接到其环境,如图 1 所示。在交互的每个步骤中,代理接收到作为输入 i 的当前状态 s 的一些指示环境;然后代理选择一个动作 a 来生成作为输出。该动作改变了环境的状态,并且这种状态转换的值通过标量强化信号 r 传达给代理。代理的行为 B 应该选择倾向于增加强化信号值的长期总和的动作。随着时间的推移,它可以通过系统的试验和错误来学习做到这一点,并在本文后面部分的主题的各种算法的指导下进行。

形式上,该模型包括

一组离散的环境状态,S;

一组离散的代理动作,A;和

一组标量强化信号;通常为 f0; 1g,或实数。

该图还包括一个输入函数 I,它确定代理如何查看环境状态;我们将假设它是恒等函数（即代理感知环境的确切状态）,直到我们在第 7 节中考虑部分可观察性。

理解代理与其环境之间关系的一种直观方法是
与以下示例对话。

环境:您处于状态 65。您有 4 个可能的操作。	
特工:	我会采取行动2。
Environment:你获得了 7 个单位的增援。你现在处于状态	
	15. 你有 2 种可能的动作。
代理人:	我会采取行动1。
环境:你获得了-4个单位的增援。你现在处于状态	
	65. 你有 4 种可能的动作。
特工:	我会采取行动2。
Environment:你获得了 5 个单位的增援。你现在处于状态	
	44. 你有 5 种可能的动作。
⋮	⋮

代理的工作是找到一个策略,将状态映射到动作,以最大化一些长期的强化措施。一般来说,我们预计环境将是不确定的;也就是说,在相同的状态下在两个不同的场合采取相同的动作可能会导致不同的下一个状态和/或不同的强化值。这发生在我们上面的示例中:从状态 65 开始,应用动作 2 会在两次产生不同的强化和不同的状态。但是,我们假设环境是静止的;也就是说,进行状态转换或接收特定强化信号的概率不会随着时间而改变。1 强化学习与更广泛研究的监督学习问题在几个方面有所不同。最重要的区别是没有输入/输出对的表示。相反,在选择了一个动作之后,代理会被告知立即奖励和随后的状态,但不会被告知哪个动作最符合其长期利益。智能体有必要积极收集有关可能的系统状态、动作、转换和奖励的有用经验,以优化行动。与监督学习的另一个区别是在线性能很重要:系统的评估通常与学习同时进行。

1. 这个假设可能会令人失望;毕竟,在非固定环境中运行是构建学习系统的动机之一。事实上,后面章节中描述的许多算法在缓慢变化的非平稳环境中是有效的,但是这方面的理论分析很少。

强化学习的某些方面与人工智能中的搜索和规划问题密切相关。 AI 搜索算法通过状态图生成令人满意的轨迹。规划以类似的方式运行,但通常在比图形更复杂的构造中,其中状态由逻辑表达式的组合而不是原子符号表示。这些 AI 算法不如强化学习方法通用,因为它们需要预先定义的状态转换模型,并且除了少数例外假设是确定性的。另一方面,强化学习,至少在理论已经发展的那种离散情况下,假设整个状态空间可以枚举并存储在内存中,这是传统搜索算法不依赖的假设。

1.2 最优行为模型

在我们开始考虑学习最优行为的算法之前,我们必须决定我们的最优模型是什么。特别是,我们必须指定智能体在决定现在如何表现时应如何考虑未来。三个模型一直是本研究的主要工作主题

区域。

nite-horizon 模型是最容易想到的;在给定的时间点,代理应该优化接下来 h 个步骤的预期奖励:

$$E\left(X^{H}_{t=0} r_t\right);$$

它不必担心之后会发生什么。在此表达式和后续表达式中,rt 表示在未来 t 步中收到的标量奖励。该模型可以以两种方式使用。首先,代理将有一个非平稳策略;也就是说,随着时间的推移而变化。在它的第一步,它将采取所谓的 h 步最优动作。这被认为是最好的行动,因为它还有 h 个步骤可以采取行动并获得强化。在下一步,它将采取 (h 1) 步最优动作,依此类推,直到它最终采取 1 步最优动作并终止。在第二个中,代理执行后退水平控制,其中它总是采取 h 步最优动作。代理总是按照相同的策略行动,但 h 的值限制了它在选择行动时看起来有多远。 nite-horizon 模型并不总是合适的。在许多情况下,我们可能无法提前知道代理人生命的确切长度。

无限期折扣模型将代理的长期奖励计入计算,但未来收到的奖励根据折扣因子 进行几何折扣,(其中 0

< 1):

$$E\left(X^{1}_{t=0} {}^{t} r_t\right):$$

我们可以用几种方式来解释。它可以被看作是一个利率,一个再迈出一步的概率,或者一个限制无穷和的数学技巧。该模型在概念上类似于后退水平控制,但贴现模型在数学上比有限水平模型更易于处理。这是引起广泛关注的主要原因

这个模型已经收到了。

另一个最优标准是平均奖励模型,其中假定代理
采取行动优化其长期平均奖励:

$$\lim_{h\to 1} E\left(\frac{1}{h}\sum_{t=0}^{H} r_t\right):$$

这种策略称为增益最优策略;当贴现因子接近 1 时,它可以被视为无限水平贴现模型的极限情况（Bertsekas,1995）。

该标准的一个问题是无法区分两种策略,其中一种在初始阶段获得大量奖励,而另一种则没有。长期平均表现掩盖了代理人生命的任何初始前所获得的奖励。可以推广该模型,使其同时考虑长期平均值和可获得的初始奖励量。

在广义的偏差最优模型中,如果某个策略最大化长期平均值并且初始额外奖励打破了平局,则该策略是首选的。

图 2 通过提供改变最优模型会改变最优策略的环境来对比这些最优模型。在这个例子中,圆圈代表环境的状态,箭头是状态转换。每个状态只有一个动作选择,除了起始状态,它位于左上角并用传入箭头标记。除标记外,所有奖励均为零。在 h = 5 的 nite-horizon 模型下,三个动作的奖励分别为 +6:0、+0:0 和 +0:0,因此应该选择第一个动作;在 = 0:9 的无限水平折扣模型下,三个选项产生 +16:2、+59:0 和 +58:5,因此应选择第二个动作;在平均奖励模型下,应该选择第三个动作,因为它导致平均奖励为 +11。如果我们将 h 更改为 1000 和 0.2,那么第二个动作对于无限水平模型是最优的,而第一个动作对于无限水平折扣模型是最优的;然而,平均奖励模型总是更喜欢最好的长期平均值。由于最优模型和参数的选择非常重要,因此在任何应用中仔细选择它是很重要的。

当智能体的生命周期已知时,nite-horizon 模型是合适的;该模型的一个重要方面是,随着剩余生命周期的缩短,代理的策略可能会发生变化。具有硬期限的系统可以通过这种方式适当建模。无限期贴现模型和偏差最优模型的相对有用性仍在争论中。偏差最优的优点是不需要折扣参数;然而,寻找偏差最优策略的算法还没有像寻找最优无限水平贴现策略的算法那样被理解。

1.3 衡量学习表现

上一节中给出的标准可用于评估给定算法学习的策略。我们还希望能够评估学习本身的质量。

有几种不兼容的措施正在使用中。

最终收敛到最优。许多算法都带有渐近收敛到最优行为的可证明保证（Watkins & Dayan, 1992）。这是令人放心的,但在实践中毫无用处。快速达到平台期的代理

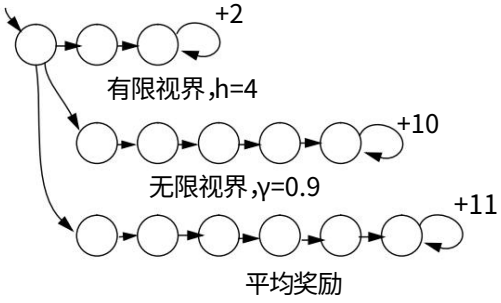

图 2:比较最优模型。所有未标记的箭头产生的奖励为零。

在许多应用中,在 99% 的最优性可能比一个能保证最终最优性但早期学习速度缓慢的代理更可取。

收敛到最优的速度。最优性通常是渐近的结果,因此收敛速度是一个错误的度量。更实际的是收敛到接近最优的速度。该度量要求定义接近最优性的程度是否足够。一个相关的衡量标准是给定时间后的性能水平,这同样需要有人定义给定时间。

应该注意的是,这里我们在强化学习和传统的监督学习之间有另一个区别。在后者中,预期的未来预测准确性或统计效率是主要关注点。例如,在著名的 PAC 框架 (Valiant, 1984) 中,有一个学习期,在此期间错误不计算在内,然后是一个绩效期,在此期间错误计算在内。该框架提供了学习期必要长度的界限,以便对后续性能有概率保证。对于在复杂环境中长期存在的代理来说,这通常是一种不恰当的看法。

尽管嵌入式强化学习和训练/测试视角之间存在不匹配,Fiechter (1994) 为 Q-learning 提供了 PAC 分析（在第 4.2 节中描述）,阐明了这两种观点之间的联系。

与学习速度相关的措施还有一个弱点。仅仅试图尽可能快地达到最优的算法可能会在学习期间招致不必要的大惩罚。一个不太激进的策略需要更长的时间才能达到最优,但在学习过程中获得更大的总强化可能是可取的。

后悔。因此,更合适的衡量标准是,由于执行学习算法而不是从一开始就表现最佳,而获得的奖励预期减少。这个度量被称为遗憾（Berry & Fristedt,1985）。它会惩罚运行期间发生的任何错误。不幸的是,关于算法遗憾的结果很难获得。

1.4 强化学习和自适应控制

自适应控制 (Burghes & Graham, 1980; Stengel, 1986) 也关注从经验中改进一系列决策的算法。自适应控制是一门更加成熟的学科,它关注动态系统,其中状态和动作是矢量,系统动力学是平滑的:围绕所需轨迹线性或局部线性化。自适应控制中一个非常常见的成本函数公式是对偏离期望状态和动作向量的二次惩罚。最重要的是,虽然事先不知道系统的动态模型,必须从数据中进行估计,但动态模型的结构是固定的,模型估计成为参数估计问题。这些假设允许进行深入、优雅和强大的数学分析,从而产生稳健、实用和广泛部署的自适应控制算法。

2. 开发与探索:单一国家案例

强化学习和监督学习之间的一个主要区别是强化学习者必须明确地探索其环境。为了突出探索的问题,我们在本节中处理一个非常简单的案例。在许多情况下,这里描述的基本问题和方法将转移到本文后面讨论的更复杂的强化学习实例。

最简单的强化学习问题被称为 k 臂老虎机问题,它一直是统计学和应用数学文献中大量研究的主题（Berry & Fristedt,1985）。代理人在一个房间里,里面有 k 台赌博机（每个都称为“单臂强盗”）。代理人被允许拉动固定的次数,h。每轮都可以拉动任何手臂。玩机器不需要押金;唯一的成本是浪费拉力玩次优机器。当拉动手臂 i 时,机器 i 支付 o 1 或 0,根据一些潜在的概率参数 pi,其中 payo s 是独立事件并且圆周率是未知的。

代理的策略应该是什么?

这个问题说明了开发和探索之间的基本权衡。
代理人可能认为特定的手臂具有相当高的支付概率;它应该一直选择那个手臂,还是应该选择另一个信息较少但似乎更糟的手臂?这些问题的答案取决于代理人预计玩游戏的时间;游戏持续的时间越长,过早收敛到次优手臂的后果就越严重,代理应该探索的越多。

这个问题有各种各样的解决方案。我们将考虑它们的代表性选择,但要进行更深入的讨论和一些重要的理论结果,请参见 Berry 和 Fristedt (1985) 的书。我们使用术语 \action 来表示智能体选择拉动的手臂。这有助于在第 3 节中过渡到延迟强化模型。非常重要的是要注意赌博机问题在我们对具有单一强化学习环境的定义中只有自我转换的状态。

2.1 节讨论了具有形式正确性结果的基本一态老虎机问题的三种解决方案。尽管它们可以扩展到具有实值奖励的问题,但它们并不直接适用于一般的多状态延迟强化情况。

第 2.2 节介绍了三种技术,它们在形式上没有得到证明,但在实践中得到了广泛的应用,并且可以应用于一般情况（同样缺乏保证）。

2.1 形式上合理的技术

对于非常简单的问题,有一个相当完善的形式探索理论。
尽管它具有指导意义,但它提供的方法不能很好地扩展到更复杂的问题。

2.1.1 动态规划方法

如果代理要执行总共 h 步,它可以使用基本的贝叶斯推理来求解最优策略（Berry & Fristedt,1985）。这需要对参数 fpig 进行假设的先验联合分布,其中最自然的是每个 pi 在 0 和 1 之间独立均匀分布。我们计算从信念状态（代理在此运行期间的经验总结)到动作的映射。在这里,信念状态可以表示为行动选择和支付的列表：fn1; w1; n2; w2; : : : ; NK; wkg 表示一种游戏状态,其中每条手臂 i 已经用 wi payo s 拉动了 ni 次。我们将 V (n1; w1; : : : ; nk; wk) 写为剩余的预期收益,假设总共有 h 个拉动可用,并且我们优化使用剩余的拉动。

如果 P i ni = h,则没有剩余的拉动,并且 V (n1; w1; : : : ; nk; wk) = 0。这是递归定义的基础。如果我们知道剩余 t 个拉动的所有信念状态的 V 值,我们可以计算剩余 t + 1 个拉动的任何信念状态的 V 值：

$$V(n_1, w_1, \ldots, n_k, w_k) = \max_i E\left[\begin{array}{l}\text{如果代理人采取行动 } i \text{,则未来收益,} \\ \text{然后代理人继续最佳行动,} \\ iV(n_1, w_i, \ldots, n_i + 1, w_i + 1, \ldots, n_k) + \\ (1 - i)V(n_1, w_i, \ldots, n_i + 1, w_i, \ldots, n_k)\end{array}\right]$$

$$= \max_i \left[{}_i(w_i + 1)=(n_i + 2)\left[1 + V(n_1, w_i, \ldots, n_i + 1, w_i + 1, \ldots, n_k)\right] + \right.$$
$$\left. (1 - {}_i(w_i + 1)=(n_i + 2))V(n_1, w_i, \ldots, n_i + 1, w_i, \ldots, n_k)\right]$$

我们的 i 是在给定 ni、wi 和
先验概率在哪里。对于导致 beta 分布的均匀先验,i (wi + 1)=(ni + 2)。

对于所有可达到的信念状态,以这种方式填充 V 值表的成本与信念状态的数量乘以行动呈线性关系,因此在视野中呈指数增长。

2.1.2 Gittins 分配指数

Gittins 给出了一种“分配指数”方法,用于在 k 臂老虎机问题的每一步中找到最佳的行动选择（Gittins，1989）。该技术仅适用于折扣预期奖励标准。对于每个行动,考虑次数它已被选择,n,与它支付的次数 o w。对于某些折扣因子,对于每对 n 和 w,都有发布的 \index 值表，“I (n; w)。查找每个动作 i, I (ni; wi) 的索引值。它代表了对行动 i 的预期收益（给定其收益 s 的历史)和我们通过选择它而获得的信息价值之间的平衡。选择具有最大索引值的动作可以保证探索和利用之间的最佳平衡。

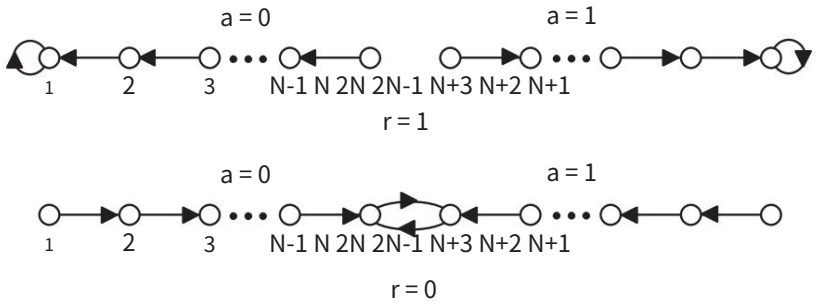

图 3:具有 2N 个状态的 Tsetlin 自动机。第一行显示了前一个动作导致奖励 1 时进行的状态转换;底行显示奖励为 0 后的转换。在图左半部分的状态中,采取行动 0;在右边的那些中,采取了行动 1。

由于保证了最佳探索和技术的简单性（给定索引值表）,这种方法在更复杂的应用程序中具有很大的应用前景。这种方法在立即奖励的机器人操作应用中被证明是有用的 (Salganico & Ungar, 1995)。不幸的是,还没有人能够找到延迟强化问题的指数值的类似物。

2.1.3 学习自动机

自适应控制理论的一个分支致力于学习自动机,由 Narendra 和 Thathachar (1989) 进行了调查,最初被明确描述为有限状态自动机。图 3 中所示的 Tsetlin 自动机提供了一个示例,该示例在 N 接近无穷大时任意接近最优地解决了 2 臂老虎机。

将算法描述为有限状态自动机是不方便的,因此采取了行动,将智能体的内部状态描述为一个概率分布,根据该概率分布选择动作。采取不同行动的概率将根据他们之前的成功和失败进行调整。

在数学心理学文献 (Hilgard & Bower, 1975) 中独立开发的一组算法中的一个例子是线性奖励-不作为算法。设 pi 是代理人采取行动 i 的概率。

当 ai 动作成功时,

$$p_i := p_i + (1 - p_i)\, p_j \quad \text{for } j \neq i$$

$$p_j := p_j$$

当动作 ai 失败时,pj 保持不变（对于所有 j）。

该算法以概率 1 收敛到包含单个 1 和其余 0 的向量（选择概率为 1 的特定动作）。不幸的是,它并不总是收敛到正确的动作;但是它收敛到错误概率的概率可以通过减小来任意减小（Narendra & Thathachar, 1974）。没有关于该算法遗憾的文献。

2.2 Ad-Hoc 技术

在强化学习实践中,一些简单的临时策略很受欢迎。它们很少（如果有的话)是我们使用的最优模型的最佳选择,但它们可能被视为合理的、计算上易于处理的启发式方法。 Thrun (1992) 调查了各种这些技术。

2.2.1 贪心策略

想到的第一个策略是始终选择具有最高估计收益的行动。 aw 是早期的不幸采样可能表明最佳动作的奖励小于从次优动作获得的奖励。次优的动作总是会被挑选出来,而真正的最优动作却缺乏数据,而且它的优势永远不会被发现。代理人必须探索以改善这一结果。

一个有用的启发式方法是面对不确定性时的乐观主义,其中行动被贪婪地选择,但强烈乐观的先验信念被放在他们的报酬上,因此需要强有力的负面证据来消除考虑中的行动。这仍然存在使最佳但不幸的行动挨饿的可衡量的危险,但这种风险可以任意小。类似的技术已被用于多种强化学习算法,包括区间探索方法 (Kaelbling, 1993b)（稍后描述）、Dyna 中的探索奖励 (Sutton, 1990)、好奇心驱动的探索 (Schmidhuber, 1991a) 和优先扫描中的探索机制 (Moore & Atkeson, 1993)。

2.2.2 随机策略

另一个简单的探索策略是默认采取具有最佳估计期望奖励的动作,但概率为 p,随机选择一个动作。该策略的某些版本以较大的 p 值开始以鼓励初始探索,该值逐渐减小。

对简单策略的反对意见是,当它尝试一个非贪婪的动作时,它不太可能尝试一个有希望的替代方案,而不是一个明显没有希望的替代方案。一个稍微复杂一点的策略是玻尔兹曼探索。在这种情况下,采取行动 a 的预期奖励,ER(a) 用于根据分布概率性地选择一个行动

$$P(a) = \frac{e^{ER(a)/T}}{\sum_{a'} e^{ER(a')/T}}$$

温度参数 T 可以随时间减小以减少探索。如果最佳动作与其他动作很好地分开,则此方法效果很好,但当动作的值接近时会有些麻烦。除非非常小心地手动调整温度计划,否则它也可能会不必要地缓慢收敛。

2.2.3 基于区间的技术

当它基于关于动作估计值的确定性或方差的二阶信息时,探索通常更有效。 Kaelbling 的区间估计算法 (1993b) 存储每个动作 ai 的统计数据:wi 是成功次数,ni 是试验次数。通过计算 100 (1)% 的上限来选择一个动作

每个动作的成功概率的置信区间,并选择具有最高上限的动作。较小的参数值鼓励更多的探索。

当 payo 是布尔值时,可以使用二项式分布的正态近似来构造置信区间(尽管二项式应该用于较小的 n)。可以使用相关的统计数据或非参数方法来处理其他收益分布。该方法在经验试验中效果很好。它还与被称为实验设计方法(Box & Draper, 1987)的某一类统计技术有关,该方法用于比较多种处理(例如,肥料或药物)以确定哪种处理(如果有)在尽可能少的一组实验。

2.3 更一般的问题

当有多个状态,但强化仍然是即时的,那么上述任何解决方案都可以复制,每个状态一次。但是,当需要泛化时,这些解决方案必须与泛化方法相结合(见第 6 节);这对于简单的 ad-hoc 方法来说是直截了当的,但不知道如何保持理论上的保证。

这些技术中的许多都专注于收敛到一些很少或从不采取探索性行动的制度;这适用于环境静止的情况。

但是,当环境不稳定时,必须继续进行探索,才能注意到世界的变化。同样,可以修改更多的临时技术以以合理的方式处理此问题(保持温度参数不为 0;衰减区间估计中的统计数据),但不能应用理论上保证的方法。

3.延迟奖励

在强化学习问题的一般情况下,代理的行为不仅决定了它的直接奖励,而且(至少在概率上)决定了环境的下一个状态。这样的环境可以被认为是强盗问题的网络,但代理在决定采取何种行动时必须考虑下一个状态以及即时奖励。代理正在使用的长期最优性模型准确地确定了它应该如何考虑未来的价值。代理必须能够从延迟强化中学习:它可能需要很长的动作序列,接受微不足道的强化,然后最终到达高度强化的状态。代理必须能够根据可以在未来任意远的地方发生的奖励来了解它的哪些行为是可取的。

3.1 马尔可夫决策过程

延迟强化的问题被很好地建模为马尔可夫决策过程(MDP)。
一个 MDP 包括

一组状态 S,

一组动作 A,

奖励函数 R:SA！ <, 和

状态转换函数 T : SA！ (S),其中 (S) 的成员是集合 S 上的概率分布（即,它将状态映射到概率）。我们写 T (s; a; s0) 表示从状态 s 到状态 s 的转换概率

^{0} 使用动作 a。

状态转换函数根据环境的当前状态和代理的动作概率性地指定环境的下一个状态。奖励函数种类期望瞬时奖励作为当前状态和动作的函数。如果状态转换独立于任何先前的环境状态或代理动作,则该模型是马尔科夫模型。

对 MDP 模型有很多很好的参考（Bellman, 1957; Bertsekas, 1987; Howard, 1960; Puterman, 1994）。

尽管一般 MDP 可能具有无限（甚至不可数)状态和动作空间,但我们将仅讨论解决有限状态和有限动作问题的方法。在第 6 节中,我们讨论解决具有连续输入和输出空间的问题的方法。

3.2 给定模型寻找策略

在我们考虑学习在 MDP 环境中表现的算法之前,我们将探索在给定正确模型的情况下确定最优策略的技术。这些动态编程技术将作为学习算法遵循的基础和灵感。我们将注意力主要集中在为无限水平贴现模型寻找最优策略,但这些算法中的大多数也有无限水平和平均案例模型的类似物。我们依赖的结果是,对于无限水平贴现模型,存在一个最优确定性平稳策略（Bellman,1957）。

我们将谈论一个状态的最优值|它是代理在该状态下开始并执行最优策略时将获得的预期无限折扣奖励总和。

作为一个完整的决策策略,它被写成

$$V(s) = \text{最大值}\ E\ X1_{t=0}\ rt!\ :$$

这个最优值函数是唯一的,可以定义为联立方程的解

$$V(s) = \text{最大值}\ 0\ @R(s;a) + X\ s02S\ T(s;a;s0)V(s^{0})1\ \text{8秒2秒}\ ;\ \text{一个};\qquad (1)$$

它断言状态 s 的值是预期的瞬时奖励加上下一个状态的预期折扣值,使用最佳可用动作。给定最优价值函数,我们可以将最优策略指定为

$$(s) = \text{参数最大值}\ 0\ @R(s;a) + X\ s02S\ T(s;a;s0)V(s^{0})1\ A:$$

3.2.1 价值迭代

那么,找到最优策略的一种方法是找到最优价值函数。它可以通过称为值迭代的简单迭代算法确定,该算法可以显示收敛到正确的 V 值（Bellman,1957;Bertsekas,1987）。

初始化 V (s) 任意循环直到策略足够好 loop for
s 2 S loop for a 2 A

$$Q(s; a) := R(s; a) + P_{s02S} T(s; a; s0)V(s0)$$

V (s) := maxa Q(s; a) end
loop end loop

何时停止值迭代算法并不明显。一个重要的结果将当前贪婪策略的性能限制为当前价值函数的贝尔曼残差的函数（Williams & Baird，1993b）。它说如果两个连续价值函数之间的最大差异小于贪心策略的值，（通过在每个状态中选择最大化估计的折扣奖励的动作获得的策略,使用当前估计的价值函数)在任何状态下与最优策略的价值函数的差异不超过 2 =(1)。这为算法提供了一个有效的停止标准。Puterman (1994) 基于跨度半范数讨论了另一个停止标准,它可能导致提前终止。另一个重要的结果是,即使价值函数可能没有收敛,贪婪策略也可以保证在有限的步骤中是最优的（Bertsekas，1987）。在实践中,贪心策略通常在价值函数收敛之前很久就已经是最优的了。

值迭代非常灵活。对 V 的分配不需要按照上面所示的严格顺序完成,而是可以并行异步发生,前提是每个状态的值在无限运行时经常无限更新。 Bertsekas (1989) 广泛处理了这些问题,他也证明了收敛结果。

基于公式 1 的更新称为完整备份,因为它们使用了 infor
来自所有可能的后继状态。可以证明表格的更新

$$Q(s; a) := Q(s; a) + (r + \max_{a0} Q(秒^{0}; a0) Q(s; a))$$

也可以使用,只要 a 和 s 的每一对经常无限更新,s 是从分布 T (s; a; s0) 中采样的,r 是用均值 R(s;a) 和有界方差采样的,并且学习率缓慢下降。这种类型的样本备份（Singh,1993)对于下一节讨论的无模型方法的操作至关重要。

具有完全备份的值迭代算法的计算复杂度,每次迭代,在状态数量上是二次的,在动作数量上是线性的。通常,转移概率 T (s; a; s0) 是稀疏的。如果平均有恒定数量的非零概率下一个状态,则每次迭代的成本与状态数量呈线性关系,与动作数量呈线性关系。如果折扣因子保持不变,达到最优价值函数所需的迭代次数是状态数和最大奖励幅度的多项式。然而,在最坏的情况下,迭代次数在 1=(1) 中呈多项式增长,因此当贴现因子接近 1 时,收敛速度会显着减慢（Littman、Dean 和 Kaelbling,1995b）。

3.2.2 策略迭代

策略迭代算法直接操作策略,而不是通过最优值函数间接找到它。它的运作方式如下:

选择任意策略循环 ^{0}

$:=$ ^{0}

计算策略的价值函数:求解线性方程组

$$V(s) = R(s; (s)) + P\ s02S\ T(s; (s); s0)V(s\ \text{改进每}\ ^{0})$$

个状态的策略:

$$^{0}\ \arg\max a\ (R(s; a) + P\ s02S\ T(s; a; s0)V(s\ (s) :=\ ^{0}))$$

直到 $=$

策略的价值函数只是在每个状态下通过执行该策略将获得的预期无限折扣奖励。它可以通过求解一组线性方程来确定。一旦我们知道了当前策略下每个状态的价值,我们就会考虑是否可以通过改变第一个采取的行动来提高价值。如果可以,我们会更改策略以在这种情况下采取新的行动。这一步保证严格提高策略的性能。当无法改进时,则保证该策略是最优的。

由于最多有 jAjJS j 个不同的策略,并且策略序列在每一步都得到改进,因此该算法最多以指数次数的迭代终止（Puter man,1994）。然而,在最坏的情况下,策略迭代需要多少次迭代是一个重要的悬而未决的问题。众所周知,运行时间是伪多项式的,对于任何固定的折扣因子,MDP 的总大小都有一个多项式界限（Littman 等人,1995b）。

3.2.3 增强价值迭代和策略迭代

在实践中,每次迭代的值迭代要快得多,但策略迭代需要更少的迭代。已经提出了这样的论点,即每种方法都更适合大型问题。 Puterman 的改进策略迭代算法 (Puterman & Shin, 1978) 提供了一种交易迭代时间的方法,以更平滑的方式改进迭代。

基本思想是策略迭代中代价高昂的部分是求解 V 的确切值。我们可以执行修改后的值迭代步骤的几个步骤,而不是找到 V 的精确值,其中策略在连续迭代中保持固定。这可以被证明产生一个近似 V ,它在 中线性收敛。在实践中,这可能会导致显着的加速。

几种加速动态规划收敛的标准数值分析技术可用于加速价值和策略迭代。多重网格方法可用于通过初始以较粗的分辨率执行值迭代来快速播种高分辨率值函数的良好初始近似值 (R ude, 1993)。状态聚合通过将状态组折叠为单个元状态来解决抽象问题（Bertsekas & Casta~non, 1989）。

3.2.4 计算复杂度

值迭代通过产生最优值函数的连续逼近来工作。
每次迭代都可以在 O(jAjjSj 转换函数中执行。但是,所需的迭代^{2}) 步,或者在稀疏的情况下更快
次数可以在折扣因子中呈指数增长 (Condon, 1992);当折扣因子接近 1 时,决策必须基于更远的结果和更远的未来。在实践中,策略迭代收敛于比值迭代更少的迭代,尽管 O(jAjjSj) 的每次迭代成本可能令人望而却步。没有已知的严格的最坏情况限制可用于策略迭代 (Littman et al ., 1995b). 修改后的策略迭代 (Puterman & Shin, 1978) 寻求廉价和有效迭代之间的权衡,并受到一些实践者的青睐 (Rust, 1996)。

$$^{2} + jSj^{3}$$

线性规划 (Schrijver, 1986) 是一个非常普遍的问题,MDP 可以通过通用线性规划包来解决 (Derman, 1970;
D Epenoux, 1963; Ho man & Karp, 1966)。这种方法的一个优点是可以获得商业质量的线性编程包,尽管时间和空间要求仍然很高。从理论上讲,线性规划是唯一可以在多项式时间内求解 MDP 的已知算法,尽管理论上有效的算法在实践中并未被证明是有效的。

4. 学习最优策略:无模型方法

在上一节中,我们回顾了为 MDP 获得最优策略的方法,假设我们已经有一个模型。该模型由状态转移概率函数 T (s; a; s0) 和强化函数 R(s; a) 的知识组成。强化学习主要关注在事先不知道这样的模型时如何获得最优策略。代理必须直接与其环境交互以获得信息,通过适当的算法,可以处理这些信息以产生最佳策略。

在这一点上,有两种方法可以继续。

无模型:无需学习模型即可学习控制器。

基于模型:学习模型,并使用它来派生控制器。

哪种方法更好?这是强化学习社区中一些争论的问题。双方都提出了许多算法。这个问题也出现在其他领域,例如自适应控制,其中直接和间接自适应控制之间的二分法。

本节检查无模型学习,第 5 节检查基于模型的方法。

强化学习代理面临的最大问题是时间信用分配。
当它可能产生深远的影响时,我们如何知道刚刚采取的行动是否是好的?一种策略是等到“结束”,如果结果好就奖励所采取的行动,如果结果不好就惩罚他们。在正在进行的任务中,很难知道“结束”是什么,这可能需要大量的记忆。相反,我们将使用来自值迭代的见解来调整基于状态的估计值

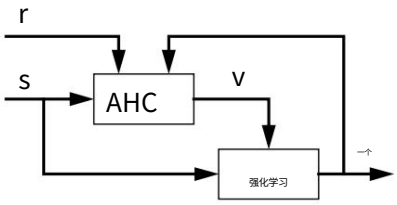

图 4：自适应启发式批评者的架构。

立即奖励和下一个状态的估计值。这类算法被称为时间差分方法（Sutton，1988）。我们将考虑两种不同的时间差分学习策略，用于折扣无限视野模型。

4.1 自适应启发式批评家和 TD( )

自适应启发式批评算法是策略迭代的自适应版本 (Barto, Sutton, & Anderson, 1983)，其中价值函数计算不再通过求解一组线性方程来实现，而是通过称为时差（0）。图 4 给出了这种方法的框图。它由两个组件组成：一个批评家（标记为 AHC）和一个强化学习组件（标记为 RL）。强化学习组件可以是任何 k 臂老虎机算法的实例，经过修改以处理多个状态和非固定奖励。但是，它不会采取行动来最大化瞬时奖励，而是采取行动来最大化由批评者计算的启发式值 v。鉴于正在执行的策略是当前在 RL 组件中实例化的策略，critic 使用真实的外部强化信号来学习将状态映射到它们的预期折扣值。

如果我们想象这些组件交替工作，我们可以看到修改后的策略迭代的类比。 RL 实施的策略是固定的，并且评论家学习该策略的价值函数 V。现在我们 xcritic 并让 RL 组件学习一个最大化新价值函数的新策略，依此类推。然而，在大多数实现中，两个组件同时操作。只有交替执行才能保证在适当的条件下收敛到最优策略。Williams 和 Baird (1993a) 探索了与 AHC 相关的算法的收敛特性，他们称之为“策略迭代的增量变体”（Williams

仍然需要解释批评者如何了解政策的价值。我们定义 hs;一个; r; s0 i 是一个经验元组，总结了环境中的单个转换。这里 s 是代理在转换之前的状态，a 是它的动作选择，r 是它收到的瞬时奖励，s 是它的结果状态。使用更新规则的 Sutton 的 TD(0) 算法 (Sutton, 1988) 学习策略的值

$$V(s) := V(s) + (r + V(s') - V(s)) :$$

每当访问状态 s 时，它的估计值就会更新为更接近 r + V (s 因为 r 是收到的瞬时奖励，V (s 0) 是实际发生的），下一个状态的估计值。这类似于值迭代的样本备份规则|唯一的区别是样本是从现实世界中抽取的，而不是通过模拟已知模型。关键思想是 r + V (s

) 是 V (s) 值的样本，它是

更可能是正确的,因为它包含了真正的 r。如果正确调整学习率（必须缓慢降低)并且策略保持固定,则可以保证 TD(0) 收敛到最优值函数。

上面介绍的 TD(0) 规则实际上是称为 TD(·) 的更通用类算法的一个实例,其中 = 0。TD(0) 在调整价值估计时看起来只领先一步;尽管它最终会得出正确的答案,但这样做可能需要相当长的时间。一般的 TD(·) 规则类似于上面给出的 TD(0) 规则,

$$V(u) := V(u) + (r + V(s^{0}) - V(s))e(u);$$

但它根据其资格 e(u) 应用于每个状态,而不仅仅是前一个状态 s。资格跟踪的一个版本被定义为

$$e(s) = \sum_{k=1}^{\text{传统知识}} {}^{s;sk}, \quad \text{在哪里} \quad {}^{s;sk} = \begin{cases} 1 & \text{如果 } s = sk \\ 0 & \text{否则} \end{cases}.$$

一个状态 s 的资格是它最近被访问的程度;当收到强化信息时,它会根据其资格更新最近访问过的所有状态。当 = 0 时,这相当于 TD(0)。当 = 1 时,大致相当于根据运行结束时访问的次数更新所有状态。请注意,我们可以在线更新资格如下:

$$e(s) := \begin{cases} e(s) + 1 & \text{如果 } s = \text{当前状态} \\ e(s) & \text{否则} \end{cases}.$$

执行一般 TD(·) 的计算成本更高,尽管它通常收敛得更快（Dayan,1992;Dayan & Sejnowski,1994）。最近有一些关于使更新更有效的工作 (Cichosz & Mulawka, 1995) 和改变定义以使 TD(·) 更符合确定性等价方法 (Singh & Sutton, 1996),这在第 5.1 节。

4.2 Q-学习

AHC 的两个组件的工作可以通过 Watkins 的 Q 学习算法以统一的方式完成（Watkins,1989;Watkins & Dayan，1992）。 Q-learning 通常更容易实现。为了理解 Q-learning,我们必须开发一些额外的符号。令 Q (s; a) 为在状态 s 中采取行动 a 的预期折扣强化,然后通过最优选择行动继续。请注意,V (s) 是假设最初采取最佳行动的 s 值,因此 V (s) = maxa Q (s; a)。 Q (s; a) 因此可以递归地写为

$$Q(s;a) = R(s;a) + X \sum_{s02S} T(s;a;s0) \text{ 最大值 } \text{问}（秒^{0}；一个0）：$$

还要注意,由于 V (s) = maxa Q (s; a),我们有 (s) = arg maxa Q (s; a) 作为最优策略。

因为 Q 函数使动作明确,我们可以使用与 TD(0) 基本相同的方法在线估计 Q 值,但也可以使用它们来定义策略,

因为可以通过选择当前状态 Q 值最大的一个动作来选择一个动作。

Q 学习规则是

$$Q(s; a) := Q(s; a) + (r + \max_{a0} Q(秒^{0}; a0) Q(s; a)) ;$$

哪里 hs;一个; r; s0 i 是如前所述的经验元组。如果每个动作在每个状态下在无限次运行中执行无限次并适当衰减，则 Q 值将以 1 到 Q 的概率收敛（Watkins,1989;Tsitsiklis,1994;Jaakkola,Jordan 和 Singh,1994）。Q-learning 也可以扩展到更新之前不止一步发生的状态,如 TD(·) (Peng & Williams, 1994)。

当 Q 值几乎收敛到它们的最优值时,agent 采取贪婪的行动是合适的,在每种情况下,采取具有最高 Q 值的动作。
然而,在学习过程中,需要在开发与探索之间进行艰难的权衡。在一般情况下,对于这个问题没有好的、形式上合理的方法;标准做法是采用第 2.2 节中讨论的一种特殊方法。

在实践层面上,AHC 架构似乎比 Q-learning 更难使用。在 AHC 中很难获得正确的相对学习率,以便两个组件收敛在一起。此外,Q-learning 对探索不敏感:也就是说,Q 值将收敛到最优值,而与收集数据时代理的行为方式无关（只要所有状态-动作对都被足够频繁地尝试）。这意味着,虽然探索-利用问题必须在 Q-learning 中解决,但探索策略的细节不会影响学习算法的收敛性。由于这些原因,Q-learning 是最流行的,并且似乎是从延迟强化中学习的最有效的无模型算法。然而,它并没有解决在大型状态和/或动作空间上进行泛化所涉及的任何问题。此外,它可能会非常缓慢地收敛到一个好的策略。

4.3 平均奖励的无模型学习

如前所述,Q-learning 可以应用于折扣无限水平 MDP。只要保证最优策略达到无奖励吸收状态并定期重置状态,它也可以应用于未折扣问题。

Schwartz (1993) 研究了将 Q-learning 应用于平均奖励框架的问题。尽管他的 R 学习算法似乎对某些 MDP 存在收敛问题,但一些研究人员发现平均奖励标准比折扣标准更接近他们希望解决的真实问题,因此更喜欢 R 学习而不是 Q 学习（Mahadevan， 1994）。

考虑到这一点,研究人员研究了学习最佳平均奖励政策的问题。 Mahadevan (1996) 从强化学习的角度调查了基于模型的平均奖励算法,发现现有算法存在一些困难。

特别是,他表明现有的平均奖励强化学习算法（和一些动态规划算法)并不总是产生偏差最优策略。 Jaakkola、Jordan 和 Singh (1995) 描述了一种具有保证收敛特性的平均奖励学习算法。它使用蒙特卡罗组件来估计代理在环境中移动时每个状态的预期未来奖励。在

此外,Bertsekas 在他的新教科书（1995)中提出了一种类似 Q-learning 的平均案例奖励算法。尽管这项最近的工作为强化学习领域提供了急需的理论基础,但许多重要问题仍未解决。

5. 通过学习模型计算最优策略

上一节展示了如何在不知道模型 T (s; a; s0) 或 R(s; a) 的情况下学习最优策略,甚至无需在途中学习这些模型。尽管这些方法中的许多方法可以保证最终找到最优策略并且每次经验使用的计算时间非常少,但它们对收集的数据的使用效率极低,因此通常需要大量经验才能获得良好的性能。

在本节中,我们仍然假设我们事先不知道模型,但我们通过学习这些模型来检查确实可以运行的算法。这些算法在计算被认为便宜且实际体验成本高的应用中尤其重要。

5.1 确定性等价方法

我们从概念上最直接的方法开始:首先,通过探索环境并保持每个动作结果的统计数据来学习 T 和 R 函数;接下来,使用第 3 节中的一种方法计算最优策略。这种方法称为确定性等价（Kumar & Varaiya,1986）。

这种方法有一些严重的反对意见:

它在学习阶段和行动阶段之间进行了任意划分。

它最初应该如何收集有关环境的数据?随机探索可能是危险的,并且在某些环境中是一种非常低效的数据收集方法,与将经验收集与政策制定更紧密地交织在一起的系统 (Koenig & Simmons, 1993) 相比,它需要的数据成倍增加 (Whitehead, 1991)。有关示例,请参见图 5。

环境变化的可能性也是有问题的。将智能体的生命分解为纯学习和纯行动阶段具有相当大的风险,即如果环境发生变化,基于早期生命的最优控制器会在没有检测到的情况下成为次优控制器。

这个想法的一个变体是确定性等价,其中模型在代理的生命周期中不断学习,并且在每一步,当前模型都用于计算最优策略和价值函数。这种方法非常有效地利用了可用数据,但仍然忽略了探索问题,并且对计算的要求非常高,即使对于相当小的状态空间也是如此。幸运的是,还有许多其他更实用的基于模型的算法。

5.2 动态

Sutton 的 Dyna 架构 (1990, 1991) 采用了一种中间立场,产生了比无模型学习更有效且计算效率高于无模型学习的策略。

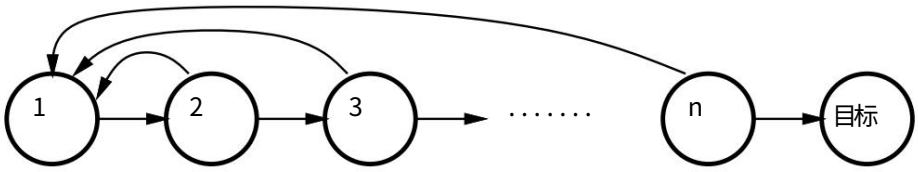

图 5:在这种环境下,由于 Whitehead (1991),随机探索甚至需要采取 O(2n) 步才能达到目标,而更智能的探索策略（例如\假设任何未尝试的行动直接导致目标）) 只需要 O(n2) 步骤。

确定性等价方法。它同时利用经验建立模型（T^ 和 R^）,利用经验调整策略,利用模型调整策略。

Dyna 在与环境交互的循环中运行。给定一个经验元组 hs;一个; s0 ; ri,它的行为如下：

更新模型,增加从 s 到 s 的转换的统计数据,以及在状态 s 中采取行动 a 获得奖励 r。更新后的模型是 T^ 0 在行动 a 和 R^。

使用规则根据新更新的模型更新状态 s 的策略

$$Q(s; a) := \hat{R}(s; a) + \gamma \sum_{s0} \hat{T}(s; a; s0) \max_{a0} Q(秒^{0}; a0) ;$$

这是 Q 值的值迭代更新的一个版本。

执行 k 个附加更新:随机选择 k 个状态-动作对,按照与之前相同的规则进行更新：

$$Q(sk; ak) := \hat{R}(sk; ak) + \gamma \sum_{s0} \hat{T}(sk ; ak; s0) \max_{a0} Q(秒^{0}; a0)$$

通过探索策略选择一个动作 a0 在状态 s 中执行。 0, 基于 Q 值,但可能经过修改

Dyna 算法需要大约 k 倍于每个实例的 Q-learning 计算,但这通常比基于模型的朴素方法要少得多。可以根据计算和采取行动的相对速度来确定合理的 k 值。

图 6 显示了一个网格世界,其中每个单元格中的代理有四个动作（N、S、E、W）,并且确定性地进行到相邻单元格的转换,除非存在块,在这种情况下不会发生移动。正如我们将在表 1 中看到的,Dyna 需要比 Q-learning 少一个数量级的经验步骤来达到最优策略。

然而,Dyna 需要大约六倍的计算量。

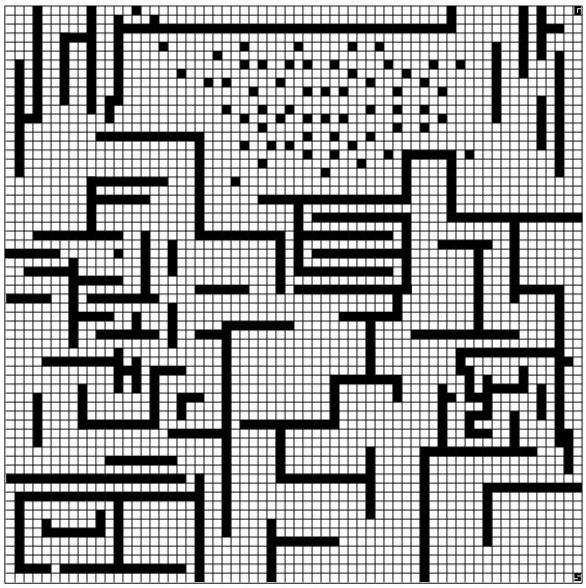

图 6:一个 3277 个状态的网格世界。这被表述为一个最短路径强化学习问题,它产生的结果与在

目标,其他地方的奖励为零,并使用折扣因子。

	备份之前的步骤		
	收敛 531,000	收敛	
Q-学习	62,000		531,000
Dyna优	28,000		3,055,000
先扫地			1,010,000

表 1:文中描述的三种算法的性能。使用的所有方法

“面对不确定性的乐观”的探索启发式:任何状态都不是

默认情况下,以前访问过的被假定为目标状态。使用 Q 学习

确定性迷宫的最佳学习率参数:= 1. Dyna 和

允许优先扫描每次转换进行 k = 200 次备份。为了

优先级扫描,优先级队列通常在所有备份完成之前清空

用过的。

5.3 优先清扫/队列动态

尽管 Dyna 对之前的方法有很大的改进,但它存在相对无向的问题。当目标刚刚达到或代理陷入死胡同时,它特别无助;它继续更新随机的状态-动作对,而不是专注于状态空间的“有趣”部分。这些问题通过优先扫描 (Moore & Atkeson, 1993) 和 Queue-Dyna (Peng & Williams, 1993) 得到解决,这是两个独立开发但非常相似的技术。我们将详细描述优先扫描。

该算法类似于 Dyna,除了不再随机选择更新并且值现在与状态相关联（如在值迭代中)而不是状态-动作对（如在 Q-learning 中）。为了做出适当的选择,我们必须在模型中存储额外的信息。每个状态都记住它的前辈:在某些动作下,具有非零转移概率的状态。此外,每个状态都有一个优先级,最初设置为

零。

不是更新 k 个随机状态-动作对,而是优先扫描更新 k 个状态
具有最高优先级。对于每个高优先级状态 s,它的工作方式如下:

记住状态的当前值:Vold = V (s)。

更新状态的值

$$V(s) := \max_{a} \Big[\hat{R}(s,a) + \gamma \sum_{s'} \hat{T}(s,a,s') V(s') \Big]$$

将状态的优先级设置回 0。

计算值变化 = jVold V (s)j。

用于修改 s 的前辈的优先级。

如果我们更新了状态 s 的直接前身的 V 值,则动作 a 它已经改变了数量 $\Delta V(s)$,然后被告
a 使得 $\hat{T}(s,a,s') \neq 0$ 将其优先级提升为已经超过 $\Delta V(s)$ 的任何状态 s,除非它的
该值的优先级。

该算法的全局行为是,当现实世界的转换“令人惊讶”时（例如,代理发生在目标状态上）,然后大量计算被引导以将该新信息传播回相关的前驱状态。当现实世界的过渡是“无聊的”（实际结果与预测结果非常相似）,然后在空间中最值得的部分继续计算。

对图 6 中的问题进行优先级扫描,我们看到比 Dyna 有了很大的改进。达到最佳策略的经验步骤数约为 Dyna 所需的一半,计算量仅为 Dyna 所需的三分之一（因此步骤减少了约 20 倍,Q-learning 的计算量增加了两倍）。

5.4 其他基于模型的方法

为给定模型而提出的解决 MDP 的方法也可以在基于模型的方法的上下文中使用。

RTDP（实时动态规划）（Barto、Bradtke 和 Singh，1995 年)是另一种基于模型的方法，它使用 Q 学习将计算工作集中在代理最有可能占据的状态空间区域上.这是特定于智能体试图实现特定目标状态而其他任何地方的奖励都是 0 的问题。

通过考虑起始状态，它可以找到一条从起始点到目标的短路径，而不必访问状态空间的其余部分。

Plexus 规划系统（Dean、Kaelbling、Kirman 和 Nicholson，1993；Kirman，1994)利用了类似的直觉。它首先制作一个比原始版本小得多的 MDP 的近似版本。近似 MDP 包含一组称为信封的状态，其中包括代理的当前状态和目标状态（如果有的话）。

不在信封中的状态由单个“out”状态进行汇总。规划过程是在近似 MDP 上找到最优策略和向信封中添加有用状态之间的交替。行动可能与规划并行发生，在在这种情况下，不相关的状态也会被剔除。

6. 泛化

之前的所有讨论都默认假设可以枚举状态和动作空间并在它们上存储值表。除了在非常小的环境中，这意味着不切实际的内存需求。它也使经验的使用效率低下。在一个大而平滑的状态空间中，我们通常期望相似的状态具有相似的值和相似的最优动作。因此，当然，应该有一些比表格更紧凑的表示。大多数问题将具有连续或较大的离散状态空间；有些会有大的或连续的动作空间。通过泛化技术解决了大空间学习的问题，该技术允许紧凑地存储学习信息并在“相似”状态和动作之间传递知识。

归纳概念学习中泛化技术的大量文献可应用于强化学习。然而，技术通常需要针对问题的具体细节进行调整。在以下部分中，我们将探讨标准函数逼近技术、自适应分辨率模型和分层方法在强化学习问题中的应用。

上面讨论的强化学习架构和算法包括存储各种映射，包括 S! A（政策），S！ <（价值函数），SA！ <（Q功能和奖励），SA！ S（确定性转换)和 SAS！ [0; 1]（转移概率）。其中一些映射，例如转换和即时奖励，可以使用直接的监督学习来学习，并且可以使用支持嘈杂训练示例的监督学习的各种函数逼近技术中的任何一种来处理。流行的技术包括各种神经网络方法 (Rumelhart & McClelland, 1986)、模糊逻辑 (Berenji, 1991; Lee, 1991)。

CMAC (Albus, 1981) 和基于局部记忆的方法 (Moore, Atkeson, & Schaal, 1995)，例如最近邻方法的推广。其他映射，尤其是策略

映射,通常需要专门的算法,因为输入-输出对的训练集不可用。

6.1 对输入的泛化

强化学习代理的当前状态在其选择奖励最大化动作中起着核心作用。将代理视为无状态黑盒,当前状态的描述是其输入。根据代理架构,其输出要么是动作选择,要么是对可用于选择动作的当前状态的评估。

决定输入的不同方面如何影响输出值的问题有时被称为“结构信用分配”问题。本节检查生成动作或评估的方法,作为对代理当前的描述的函数状态。

这里介绍的第一组技术专门用于没有奖励的情况
延迟;第二组更普遍适用。

6.1.1 即时奖励

当代理的动作不影响状态转换时,由此产生的问题变成了根据代理的当前状态选择最大化即时奖励的动作。这些问题与第 2 节中讨论的老虎机问题相似,只是智能体应根据当前状态选择其动作。出于这个原因,这类问题被描述为关联强化学习。

本节中的算法解决了从即时布尔强化中学习的问题,其中状态是向量值,动作是布尔向量。此类算法可以并且已经在延迟强化的上下文中使用,例如,作为第 4.1 节中描述的 AHC 架构中的 RL 组件。它们也可以通过奖励比较方法推广到实际价值奖励（Sutton,1984）。

CRBP 互补强化反向传播算法 (Ackley & Littman, 1990) (crbp) 由一个前馈网络组成,将状态编码映射到动作编码。动作是根据输出单元的激活概率确定的:如果输出单元 i 具有激活 yi,则动作向量的位 i 的值为 1,概率为 yi,否则为 0。任何神经网络监督训练过程都可用于如下调整网络。如果生成动作 a 的结果是 r = 1,则使用输入-输出对 hs 训练网络;艾。如果结果是 r = 0,则使用输入-输出对 hs 训练网络； ai,其中 a = (1 a1; : : : ; 1 an)。

这个训练规则背后的想法是,每当一个动作未能产生奖励时,crbp 将尝试产生一个与当前选择不同的动作。虽然看起来算法可能会在一个动作和它的补码之间摇摆不定,但这并没有发生。训练网络的一步只会稍微改变动作,并且由于输出概率将趋向于 0.5,这使得动作选择更加随机并增加了搜索。希望随机分布会产生一个效果更好的动作,然后该动作将得到加强。

ARC 关联强化比较 (arc) 算法 (Sutton, 1984) 是用于布尔动作情况的 ahc 架构的一个实例,由两个提要组成

前向网络。一个学习情境的价值,另一个学习策略。这些可以是简单的线性网络,也可以有隐藏单元。

在最简单的情况下,整个系统只学习优化即时奖励。首先,让我们考虑学习策略的网络的行为,从描述 s 的向量到 0 或 1 的映射。如果输出单元有激活 yi,那么如果 y + >,则生成的动作 a 将为 1 0,其中为正常噪声,否则为 0。

在最简单的情况下,输出单元的调整是

$$e = r(a\ 1=2)\ ;$$

其中第一个因素是采取最近行动获得的奖励,第二个因素编码采取了哪些行动。动作被编码为 0 和 1,因此 1=2 始终具有相同的幅度;如果奖励和动作具有相同的符号,则动作 1 的可能性更大,否则动作 0 的可能性更大。

如前所述,网络将倾向于寻求给予积极回报的行动。为了扩展这种方法以最大化奖励,我们可以将奖励与某个基线进行比较,b。这会将调整更改为

$$e = (rb)(a\ 1=2)\ ;$$

其中 b 是第二个网络的输出。第二个网络在标准监督模式下进行训练,以估计 r 作为输入状态 s 的函数。

这种方法的变体已被用于各种应用中（Anderson,1986;
巴托等人,1983;林,1993b;萨顿,1984）。

强化算法 Williams (1987, 1992) 研究了选择行动以最大化即时奖励的问题。他确定了一类广泛的更新规则,它们对预期奖励执行梯度下降,并展示了如何将这些规则与反向传播相结合。此类称为强化算法,包括作为特例的线性奖励不作为（第 2.1.3 节）。

参数 wij 的通用强化更新可以写成

$$wij = \quad ij\ (r\ bij)\ ln(gj)@wij$$

其中 ij 是非负因子,r 是当前强化,bij 是强化基线,gi 是用于基于单元激活随机生成动作的概率密度函数。和 bij 都可以为每个 wij 取不同的值,然而,当 ij 在整个系统中保持不变时,预期更新正好在预期奖励梯度的方向上。否则,更新与梯度在同一个半空间中,但不一定在最陡峭的方向上。

Williams 指出基线 bij 的选择会对算法的收敛速度产生深远的影响。

基于逻辑的方法 强化学习中泛化的另一种策略是将学习问题简化为学习布尔函数的关联问题。

布尔函数具有一个布尔输入向量和一个布尔输出。从主流机器学习工作中汲取灵感,Kaelbling 开发了两种从强化学习布尔函数的算法:一种使用 k-DNF 的偏差来驱动

泛化过程（Kaelbling,1994b）;另一个使用简单的生成和测试方法（Kaelbling,1994a)搜索函数的句法描述空间。

对单个布尔输出的限制使得这些技术难以应用。在非常良性的学习情况下,可以扩展这种方法以使用一组学习器来独立地学习构成复杂输出的各个位。然而,一般来说,这种方法存在强化非常不可靠的问题:如果单个学习器生成不适当的输出位,所有学习器都会收到低强化值。级联方法（Kaelbling,1993b)允许对一组学习者进行集体训练,以产生适当的联合输出;它要可靠得多,但可能需要额外的计算工作。

6.1.2 延迟奖励

另一种允许在大型状态空间中应用强化学习技术的方法是以值迭代和 Q 学习为模型。这里,函数逼近器用于通过将状态描述映射到值来表示值函数。

许多研究人员已经尝试过这种方法:Boyan 和 Moore (1995) 将基于局部记忆的方法与值迭代结合使用； Lin (1991) 使用反向传播网络进行 Q 学习； Watkins (1989) 使用 CMAC 进行 Q 学习； Tesauro (1992, 1995) 使用反向传播来学习双陆棋中的价值函数（在第 8.1 节中描述）； Zhang 和 Dietterich (1995) 使用反向传播和 TD(·) 来学习作业车间调度的良好策略。

尽管有一些积极的例子,但总的来说,函数逼近和学习规则之间的相互作用是不幸的。在离散环境中,保证任何更新价值函数的操作（根据贝尔曼方程)只能减少当前价值函数和最优价值函数之间的误差。当使用泛化时,这个保证不再成立。 Boyan 和 Moore (1995) 讨论了这些问题,他们给出了一些简单的例子,即当泛化与值迭代一起使用时,值函数误差会变得任意大。

他们对此的解决方案仅适用于某些类别的问题,通过仅允许通过一系列蒙特卡洛实验证明其估计值接近最优的更新来阻止这种分歧。

Thrun 和 Schwartz (1993) 认为,价值函数的函数逼近也是危险的,因为在价值函数的定义中,由于泛化导致的价值函数误差可能会因 \max 运算符而变得复杂。

最近的几个结果（Gordon,1995; Tsitsiklis&van Roy,1996)表明,函数近似器的选择性选择如何保证收敛,尽管不一定是最佳值。 Baird 的残差梯度技术 (Baird, 1995) 保证收敛到局部最优解。

也许这些反例的悲观情绪是错误的。 Boyan 和 Moore (1995) 报告说,尽管可证明在离散域中收敛的未调整算法不可靠,但他们的反例可以用于针对特定问题的手动调整。

Sutton (1996) 展示了 Boyan 和 Moore 示例的修改版本如何成功地收敛。一个悬而未决的问题是,在理论上得到理论支持的一般原则是否可以帮助我们理解价值函数逼近何时会成功。在萨顿的 com

以博彦和摩尔的反例进行对比实验,他改变了实验的四个方面:

1.任务规范的小改动。

2. 一种非常不同的函数逼近器 (CMAC (Albus, 1975)),它具有弱
概括。

3. 一种不同的学习算法:SARSA (Rummery & Niranjan, 1994) 而不是 value
迭代。

4. 不同的培训制度。 Boyan 和 Moore 在状态空间中均匀地采样状态,而 Sutton 的方法沿经验轨迹采样。

有直观的理由相信第四个因素特别重要,但需要更仔细的研究。

自适应分辨率模型在许多情况下,我们想做的是将环境划分为状态区域,这些状态区域可以被认为是相同的,以便学习和生成动作。如果没有详细的环境先验知识,很难知道分区的粒度或位置是合适的。使用自适应分辨率的方法克服了这个问题。在学习过程中,构建适合环境的分区。

决策树 在以一组布尔值或离散值变量为特征的环境中,可以学习紧凑的决策树来表示 Q 值。 G 学习算法 (Chapman & Kaelbling, 1991) 的工作原理如下。它首先假设不需要分区,并尝试学习整个环境的 Q 值,就好像它是一个状态一样。在此过程中,它会根据各个输入位收集统计信息;它询问状态描述中是否存在某个位 b 的问题,使得 b = 1 的状态的 Q 值与 b = 0 的状态的 Q 值显着不同。如果找到这样的位,它是用于分割决策树。然后,在每个叶子中重复该过程。这种方法能够在存在大量不相关、嘈杂的状态属性的情况下学习 Q 函数的非常小的表示。它在简单的视频游戏环境中通过反向传播优于 Q 学习,并被 McCallum (1995) 使用(结合其他处理部分可观察性的技术)来学习复杂驾驶模拟器中的行为。但是,它不能获取属性仅在组合中显着的分区(例如解决奇偶校验问题所需的分区)。

可变分辨率动态规划 VRDP 算法 (Moore, 1991) 使传统的动态规划能够在实值多元状态空间中执行,其中直接离散化将成为维数诅咒的牺牲品。 kd-tree(类似于决策树)用于将状态空间划分为粗略区域。粗略区域被细化为详细区域,但仅在预测重要的部分状态空间中。这种重要性的概念是通过在状态空间中运行“心理轨迹”来获得的。这种算法在许多问题上证明是有效的,而对于这些问题来说,完整的高分辨率阵列是不切实际的。它的缺点是需要在一个通过状态空间的最初有效轨迹。

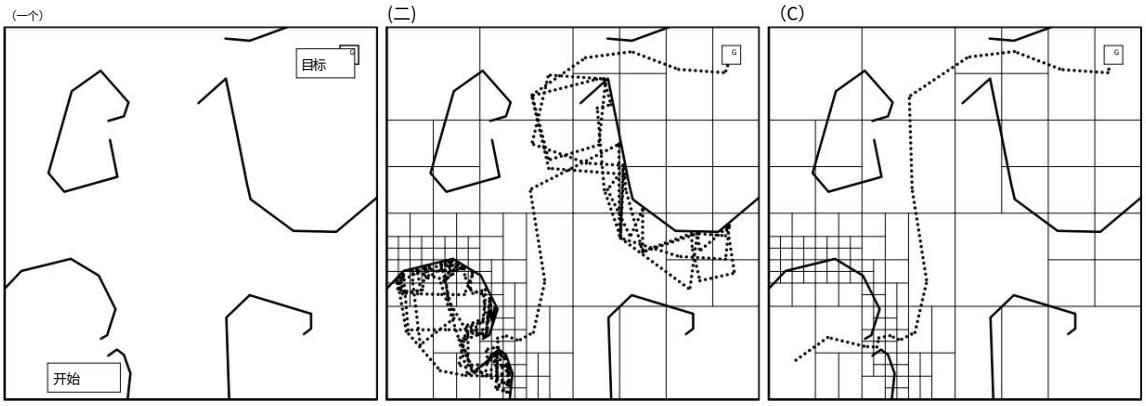

图 7:(a) 一个二维迷宫问题。点位机器人必须找到一条从起点到终点的路径,且不得越过任何障碍线。 (b) PartiGame 在整个第一次审判期间所采取的路径。它从紧张的探索开始,寻找一条从几乎完全封闭的起始区域出来的路线。最终达到足够高的分辨率后,它发现了缝隙,贪婪地朝着目标前进,却被目标的屏障区域暂时挡住了。 (c)

二审。

PartiGame 算法 Moore 的 PartiGame 算法 (Moore, 1994) 是通过学习自适应分辨率模型来解决在确定性高维连续空间中实现目标配置问题的另一种解决方案。它还将环境划分为细胞;但是在每个单元格中,可用的操作包括瞄准相邻的单元格（此瞄准由本地控制器完成,必须作为问题陈述的一部分提供）。单元转换图以在线增量方式求解最短路径,但使用极小极大标准来检测一组单元何时过于粗糙而无法防止障碍物之间的移动或避免限制循环。结束单元格被拆分为更高的分辨率。最终,环境被划分为刚好足以选择适当的行动来实现目标,但没有进行不必要的区分。

一个重要的特点是,除了减少内存和计算需求外,它还以多分辨率方式构建状态空间的探索。给定一个失败,代理最初会尝试一些非常不同的方法来纠正失败,并且只有在所有质量不同的策略都用尽时才诉诸小的局部变化。

图 7a 显示了一个二维连续迷宫。图 7b 显示了在第一次试验期间使用 PartiGame 算法的机器人的性能。图 7c 显示了第二次试验,从稍微不同的位置开始。

这是一种非常快速的算法,可以在不到一分钟的时间内学习多达九个维度的空间中的策略。然而,当前实现对确定性环境的限制限制了它的适用性。 McCallum (1995) 提出了一些相关的树结构

方法。

6.2 对动作的泛化

第 6.1.1 节中描述的网络概括了作为输入呈现的状态描述。它们还以离散的因子表示形式产生输出,因此也可以被视为对动作的泛化。

在这种情况下,当动作被组合描述时,对动作进行概括很重要,以避免为可以选择的大量动作保留单独的统计数据。在连续动作空间中,对泛化的需求更加明显。

当使用神经网络估计 Q 值时,可以为每个动作使用不同的网络,或者为每个动作使用具有不同输出的网络。当动作空间是连续的时,这两种方法都不可能。另一种策略是使用单个网络,将状态和动作作为输入,Q 值作为输出。训练这样的网络在概念上并不困难,但使用网络来找到最佳动作可能是一个挑战。一种方法是对动作进行局部梯度上升搜索,以找到具有高价值的动作(Baird & Klopf,1993)。

Gullapalli (1990, 1992) 开发了一种用于连续动作空间的“神经”强化学习单元。该单元生成具有正态分布的动作;它根据以前的经验调整均值和方差。当所选动作没有执行时好吧,方差很大,导致对选择范围的探索。当一个动作表现良好时,均值向那个方向移动,方差减小,导致在成功的附近产生更多动作值的趋势。这种方法成功用于学习控制具有许多连续自由度的机械臂。

6.3 分层方法

处理大型状态空间的另一种策略是将它们视为学习问题的层次结构。在许多情况下,分层解决方案会在性能上引入轻微的次优性,但可能会在执行时间、学习时间和空间方面获得大量效率。

分层学习器通常被构造为门控行为,如图 8 所示。
有一组将环境状态映射为低级动作的行为和一个门控函数,它根据环境的状态决定应该切换并实际执行哪些行为的动作。 Maes 和 Brooks (1990) 使用了这种架构的一个版本,其中个体行为是先验的,门控功能是从强化中学习的。 Mahadevan 和 Connell (1991b) 使用了双重方法:他们固定了门控功能,并为学习到的个人行为提供了强化功能。 Lin (1993a) 和 Dorigo 和 Colombetti (1995, 1994) 都使用了这种方法,首先训练行为,然后训练门控函数。许多其他分层学习方法都可以在这个框架中使用。

6.3.1 封建 Q 学习

封建 Q 学习 (Dayan & Hinton, 1993; Watkins, 1989) 涉及学习模块的层次结构。在最简单的情况下,有一个高级主控和一个低级从属。主人从外部环境中得到强化。它的动作由命令组成

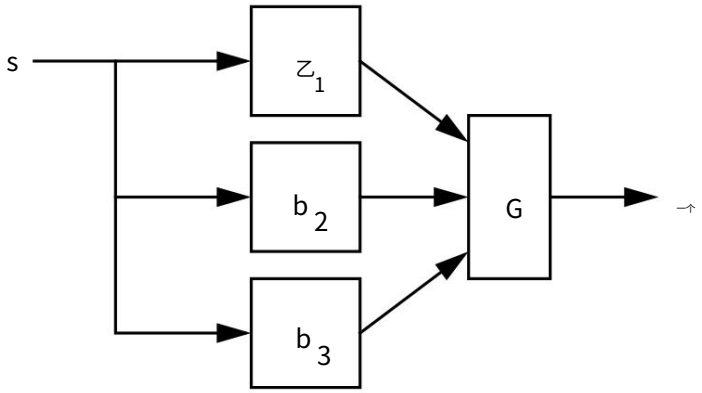

图 8:门控行为的结构。

它可以给低水平的学习者。当主人向奴隶发出特定命令时,它必须奖励奴隶采取满足命令的行动,即使它们不会导致外部强化。然后,主设备学习从状态到命令的映射。从站学习从命令和状态到外部动作的映射。 \commands 的集合及其相关的强化函数是在学习之前建立的。

这实际上是一般“门控行为”方法的一个实例,其中从属可以根据其命令执行任何行为。给出了单个行为（命令)的强化函数,但学习同时发生在两个高水平和低水平。

6.3.2 组合 Q 学习

Singh 的组合 Q 学习 (1992b, 1992a) (C-QL) 由基于子目标时间顺序的层次结构组成。基本任务是达到某种可识别条件的行为。该系统的高级目标是按顺序实现某些条件集。条件的实现为基本任务提供了强化,这些基本任务首先被训练以实现单个子目标。然后,门控函数学习切换元素任务以实现适当的高级顺序目标。 Tham 和 Prager (1994) 使用这种方法来学习控制模拟的多连杆机械臂。

6.3.3 到目标的分层距离

尤其是如果我们将强化学习模块视为更大的代理架构的一部分,重要的是要考虑目标是动态输入给学习器的问题。 Kaelbling 的 HDG 算法 (1993a) 使用分层方法来解决当实现目标（代理应尽快达到特定状态)动态地分配给代理时的问题。

HDG 算法的工作原理与港口中的导航类似。环境被划分（先验,但最近的工作（Ashar,1994)解决了学习划分的情况)为一组区域,其中心被称为 \landmarks。”如果代理是

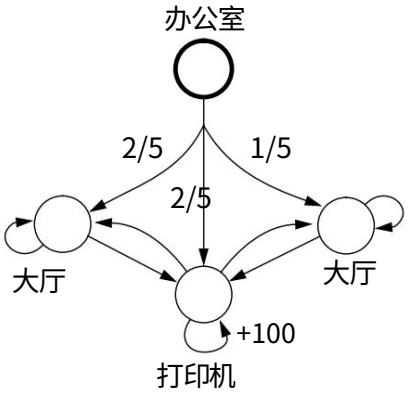

图 9:部分可观察环境的示例。

当前与目标在同一区域,然后它使用低级动作移动到目标。
如果不是,则使用高级信息来确定从代理最近的地标到目标最近的地标的最短路径上的下一个地标。然后,代理使用低级信息瞄准下一个地标。如果动作错误导致路径偏差,则没有问题;每一步都会重新计算最佳瞄准点。

7. 部分可观察的环境

在许多现实世界的环境中,智能体不可能对环境状态有完美和完整的感知。不幸的是,基于 MDP 的学习方法需要完全可观察性。在本节中,我们考虑代理对环境状态进行观察的情况,但这些观察结果可能是嘈杂的并且提供的信息不完整。例如,在机器人的情况下,它可能会观察它是否在走廊、开放的房间、T 字形路口等,而这些观察结果可能容易出错。这个问题也被称为“不完整感知”、“感知混叠”或“隐藏状态”的问题。

在本节中,我们将考虑扩展基本 MDP 框架以解决部分可观察问题。由此产生的形式模型称为部分可观察马尔可夫决策过程或 POMDP。

7.1 无状态确定性策略

处理部分可观察性最天真的策略是忽略它。也就是说,将观察结果视为环境状态并尝试学习表现。图 9 显示了一个简单的环境,其中代理尝试从办公室访问打印机。如果它从办公室移动,那么代理很可能最终会在两个看起来像“大厅”的地方之一,但这需要不同的操作才能到达打印机。如果我们认为这些状态是相同,那么智能体不可能表现得最佳。但它能做到多好呢?

由此产生的问题不是马尔可夫问题,并且不能保证 Q-learning 收敛。 Q-learning 可以很好地处理对 Markov 要求的小的违反,但是可以构建导致 Q-learning 振荡的简单环境(Chrisman &

利特曼,1993）。但是,可以使用基于模型的方法;根据某些策略采取行动并收集有关观察之间转换的统计数据,然后根据这些观察求解最佳策略。不幸的是,当环境不是马尔可夫时,转移概率取决于正在执行的策略,因此这个新策略将引发一组新的转移概率。在某些情况下,这种方法可能会产生合理的结果,但同样不能保证。

但是,询问最佳策略（在这种情况下是从观察到行动的映射)是合理的。找到这个映射是 NP-hard (Littman, 1994b),即使是最好的映射也可能性能很差。例如,在我们的代理试图到达打印机的情况下,任何确定性无状态策略平均需要无数步才能达到目标。

7.2 无状态随机策略

通过考虑随机策略可以获得一些改进;这些是从观察到动作概率分布的映射。如果智能体的动作有随机性,它就不会永远卡在大厅里。 Jaakkola、Singh 和 Jordan (1995) 开发了一种用于寻找局部最优随机策略的算法,但寻找全局最优策略仍然是 NP 难题。

在我们的示例中,事实证明最优随机策略是让智能体在看起来像大厅的状态下以 2 p 2 0:6 的概率向东移动,以 p 2 1 0:4 的概率向西移动。这个策略可以通过求解一个简单的（在这种情况下)二次程序来找到。这样一个简单的例子表明,即使是一个很难准确解决的问题。

7.3 具有内部状态的策略

在广泛的环境中真正有效地表现的唯一方法是使用对先前动作和观察的记忆来消除当前状态的歧义。学习具有内部状态的策略有多种方法。

循环 Q 学习 一种直观简单的方法是使用循环神经网络来学习 Q 值。网络可以通过时间反向传播（或一些其他合适的技术）进行训练,并学习保留“历史特征”以预测价值。这种方法已被许多研究人员使用（Meeden、McGraw 和 Blank,1993;Lin & Mitchell, 1992; Schmidhuber, 1991b). 它似乎对简单问题有效,但在更复杂的问题上可以收敛到局部最优。

分类系统分类系统 (Holland, 1975; Goldberg, 1989) 被明确开发用于解决延迟奖励的问题,包括那些需要短期记忆的问题。通常用于通过决策链将奖励传回的内部机制,称为桶式算法,与 Q-learning 非常相似。尽管早期取得了一些成功,但最初的设计似乎并不能稳健地处理部分观察到的环境。

最近,使用强化学习文献中的见解重新检查了这种方法,并取得了一些成功。 Dorigo 对 Q 学习和分类系统进行了比较研究 (Dorigo & Bersini, 1994)。 Cli 和 Ross (1994) 从 Wilson 的零开始

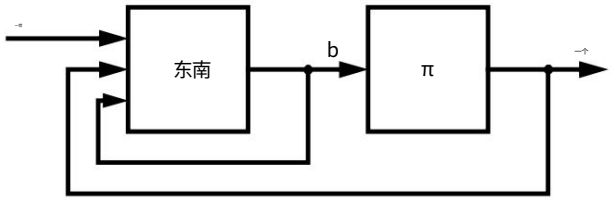

图 10:POMDP 代理的结构。

级别分类系统（Wilson,1995)并添加一位和两位内存寄存器。他们发现,尽管他们的系统可以有效地学习使用短期记忆寄存器,但这种方法不太可能扩展到更复杂的环境。

Dorigo 和 Colombetti 将分类系统应用于从即时强化学习机器人行为的中等复杂问题（Dorigo,1995；Dorigo 和 Colombetti,1994）。

有限历史窗口方法 恢复马尔可夫属性的一种方法是允许决策基于最近观察的历史和可能的行动。 Lin 和 Mitchell (1992) 使用固定宽度的有限历史窗口来学习极点平衡任务。

McCallum (1995) 描述了“\utile su x memory”,它学习了一个可变宽度窗口,该窗口同时用作环境模型和有限内存策略。该系统在非常复杂的驾驶模拟领域取得了出色的结果（ McCallum, 1995). Ring (1994) 有一种使用可变历史窗口的神经网络方法,在必要时添加历史以消除情况的歧义。

POMDP 方法 另一种策略包括使用隐马尔可夫模型 (HMM) 技术来学习环境模型,包括隐藏状态,然后使用该模型构建完美的内存控制器 (Cassandra, Kaelbling, & Littman, 1994; Lovejoy, 1991 年;莫纳汉，1982 年）。

Chrisman (1992) 展示了学习 HMM 的前向后向算法如何适应学习 POMDP。他和后来的 McCallum (1993) 也给出了启发式状态分裂规则,试图为给定环境学习最小的可能模型。然后可以使用生成的模型来整合来自代理观察的信息,以便做出决策。

图 10 说明了完美内存控制器的基本结构。左边的组件是状态估计器,它计算代理的信念状态 b 作为旧信念状态、最后一个动作 a 和当前观察 i 的函数。在这种情况下,信念状态是环境状态的概率分布,表明在给定代理过去的经验的情况下,环境实际上处于这些状态中的每一个的可能性。

状态估计器可以使用估计的世界模型和贝叶斯规则直接构建。

现在我们剩下的问题是如何将策略映射信念状态付诸行动。
这个问题可以表述为 MDP,但使用前面描述的技术很难解决,因为输入空间是连续的。 Chrisman 的方法（1992)没有考虑未来的不确定性,而是在经过少量计算后得出一个策略。运筹学文献中的一个标准方法是解决

最优策略（或其近似值)基于其表示为信念空间上的分段线性和凸函数。这种方法在计算上难以处理,但可以作为进一步近似方法的灵感（Cassandra 等人,1994;Littman、Cassandra 和 Kaelbling,1995a）。

8. 强化学习应用

强化学习流行的一个原因是它可以作为一种理论工具来研究智能体学习行动的原理。但不足为奇的是,它也被许多研究人员用作一种实用的计算工具,用于构建自主系统,并通过经验改进自身。这些应用范围从机器人技术到工业制造,再到计算机游戏等组合搜索问题。

实际应用提供了对学习算法有效性和实用性的测试。
它们也是决定强化学习框架的哪些组件具有实际重要性的灵感。例如,具有真正机器人任务的研究人员可以为以下问题提供数据点：

最优探索有多重要?我们可以将学习期分为探索阶段和开发阶段吗?

什么是最有用的长期奖励模型:有限视野?打折?
无限地平线?

代理决策之间有多少计算可用,应该如何使用?

我们可以在系统中构建哪些先验知识,哪些算法能够使用这些知识?

让我们检查强化学习的一组实际应用,同时牢记这些问题。

8.1 玩游戏

自从该领域诞生以来,游戏就作为一个问题领域主导了人工智能世界。两人游戏不属于已建立的强化学习框架,因为游戏的最优性标准不是在固定环境下最大化奖励之一,而是针对最优对手（极小极大)最大化奖励之一。尽管如此,强化学习算法可以适用于非常通用的游戏类别（Littman,1994a）,并且许多研究人员已经在这些环境中使用了强化学习。 Samuel 的跳棋游戏系统（Samuel,1959)是一个远远超前于时代的应用。这学习了一个由线性函数逼近器表示的值函数,并采用了类似于值迭代、时间差异和 Q 学习中使用的更新的训练方案。

最近,Tesauro (1992, 1994, 1995) 将时间差异算法应用于西洋双陆棋。西洋双陆棋有大约 1020 个州,因此无法进行基于表格的强化学习。相反,Tesauro 使用了基于反向传播的三层

	训练 游戏	隐 单位	结果
基本 TD 1.0	300,000	80	较差的 51场失13分 游戏
TD 2.0	800,000	40	38场失7分 游戏
TD 2.1	1,500,000	80	40 分输 1 分 游戏

表 2:TD-Gammon 在与顶级人类职业选手的比赛中的表现。
西洋双陆棋锦标赛涉及玩一系列游戏以获得积分,直到一个
玩家达到设定的目标。 TD-Gammon 没有赢得这些比赛,但来了
足够接近以至于它现在被认为是世界上最好的少数球员之一。

神经网络作为价值函数的函数逼近器

董事会职位!当前玩家获胜的概率:

使用了两个版本的学习算法。第一个,我们称之为基本 TD Gammon,使用了很少的游戏知识,并且表示

棋盘位置实际上是一种原始编码,其功能强大到只允许神经
网络来区分概念上不同的位置。第二个,TD-Gammon,
提供了相同的原始状态信息,并辅以一些手工制作的双陆棋棋盘位置特征。在此提供手工制作的功能

方式是一个很好的例子,说明人类对任务知识的归纳偏见如何
提供给学习算法。

两种学习算法的训练都需要几个月的计算机时间,并且
是通过不断的自我游戏来实现的。未使用任何探索策略|系统始终
贪婪地选择了预期获胜概率最大的着法。事实证明,这种幼稚的探索策略完全适合这种环境,这可能令人惊讶

鉴于强化学习文献中的大量工作已经产生
许多反例表明贪婪的探索会导致学习效果不佳。然而,双陆棋有两个重要的特性。首先,无论什么政策

其次,每场比赛都保证在有限时间内结束,这意味着有用的奖励
信息被相当频繁地获取。其次,状态转换充分
随机的,独立于策略,偶尔会访问所有状态|一个错误
初始值函数几乎没有让我们无法访问状态关键部分的危险
可以从中获取重要信息的空间。

TD-Gammon 的结果(表 2)令人印象深刻。它已经在最顶端竞争
国际人类游戏水平。基本的 TD-Gammon 打得不错,但不是
专业标准。

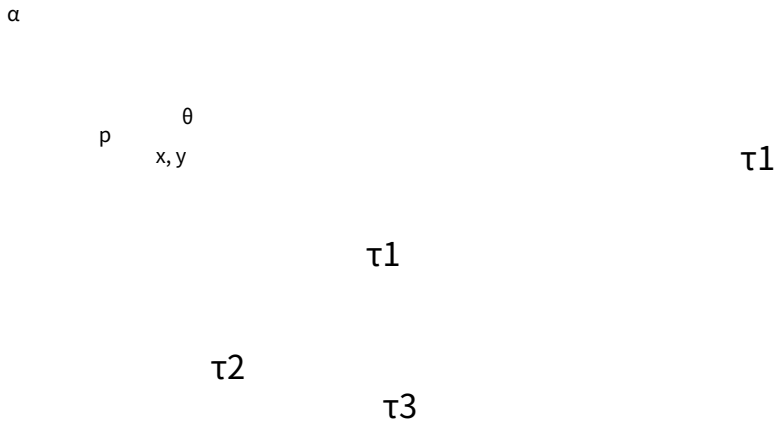

图 11：Schaal 和 Atkeson 的魔鬼贴机器人。锥形棒由两个手棒中的每一个交替击打。任务是尽可能多地防止恶魔棒掉落。机器人具有三个由扭矩矢量表示的电机

1个；2；3.

尽管对其他游戏的实验在某些情况下产生了有趣的学习行为，但没有重复接近 TD-Gammon 的成功。其他已经研究过的游戏包括围棋（Schraudolph、Dayan 和 Sejnowski，1994）和国际象棋（Thrun，1995）。 TD-Gammon 的成功是否以及如何在其他领域重复，这仍然是一个悬而未决的问题。

8.2 机器人与控制

近年来，已经有许多机器人和控制应用程序使用了强化学习。在这里，我们将专注于以下四个示例，尽管许多其他有趣的正在进行的机器人研究正在进行中。

1. Schaal 和 Atkeson (1994) 构建了一个双臂机器人，如图 11 所示，它学会了玩一种被称为魔鬼棒的装置。这是一个复杂的非线性控制任务，涉及六维状态空间，每个控制决策不到 200 毫秒。在大约 40 次初始尝试后，机器人学会了继续杂耍数百次击球。一个典型的人类学习任务需要更多的练习才能在几十次点击中达到熟练程度。

   杂耍机器人从经验中学习了一个世界模型，该模型通过称为局部加权回归的函数近似方案推广到未访问状态（Cleveland & Delvin, 1988; Moore & Atkeson, 1992）。在每次试验之间，使用一种特定于线性控制策略和局部线性转换的动态规划形式来改进策略。动态规划的形式被称为线性二次调节器设计（Sage & White，1977）。

2. Mahadevan 和 Connell (1991a) 讨论了移动机器人长时间推动大箱子的任务。 Box-pushing是一个众所周知的困难机器人问题,其特点是行动结果的巨大不确定性。 Q-learning 与一些新颖的聚类技术结合使用,旨在实现比表格方法所允许的更高维度的输入。机器人学会了与人类编程解决方案的性能竞争。在第 6.3 节中提到的这项工作的另一个方面是将整体任务描述预先编程分解为一组要学习的较低级别的任务。

3. Mataric (1994) 描述了一个机器人实验,从理论强化学习的角度来看,这是一个不可想象的高维状态空间,包含几十个自由度。四个移动机器人在一个封闭空间内移动,收集小磁盘并将它们运送到目的地区域。对基本 Q 学习算法进行了三项增强。首先,使用称为进度估计器的预编程信号将单一任务分解为子任务。

这是以一种稳健的方式实现的,其中机器人不会被迫使用估计器,而是可以自由地避免它们提供的归纳偏差。
其次,控制是分散的。每个机器人都独立地学习自己的策略,而无需与其他机器人进行明确的交流。第三,根据底层传感器的少量预编程布尔特征的值,将状态空间粗暴地量化为少量离散状态。 Q-learned 策略的性能几乎与简单的手工控制器一样好。

4. Q-learning 已被用于电梯调度任务（Crites & Barto, 1996）。该问题仅在现阶段在模拟中实施,涉及为十层楼提供服务的四部电梯。目标是最小化乘客的平均平方等待时间,折扣到未来时间。该问题可以提出为离散马尔可夫系统,但即使在该问题的最简化版本中也有 1022 个状态。 Crites 和 Barto 使用神经网络进行函数逼近,并对他们的 Q 学习方法与最流行和最复杂的电梯调度算法进行了出色的比较研究。他们的控制器的平方等待时间比最佳替代算法（“清空系统”启发式与后退水平控制器）少了大约 7%,并且不到实际电梯系统中最常用的控制器的平方等待时间的一半。

5. 最后一个示例涉及本调查的一位作者将强化学习应用于食品加工业的包装任务。问题涉及用可变数量的不同产品填充容器。

产品特性也随时间变化,但可以感知。根据任务,对容器填充过程施加了各种约束。以下是三个例子：

一个班次生产的所有容器的平均重量不得低于制造商声明的重量 W。

低于申报重量的集装箱数量必须少于 P%。

不得生产重量低于 W0 的容器。

这些任务由根据各种设定值运行的机器控制。
传统做法是由人工操作员选择设定点,但这种选择并不容易,因为它取决于当前的产品特性和当前的任务限制。依赖关系通常难以建模且高度非线性。

该任务被提出为一个无限期马尔可夫决策任务,其中系统的状态是产品特性、生产班次中剩余的时间量以及迄今为止在班次中宣布的平均浪费和低于百分比的函数。该系统被离散化为 200,000 个离散状态,并使用局部加权回归来学习和泛化过渡模型。当获得每条新的转换信息时,使用优先扫描来保持最佳值函数。在模拟实验中,节省的费用是相当可观的,通常浪费会减少十倍。从那时起,该系统已成功部署在美国的几家工厂。

这些示例揭示了实际强化学习的一些有趣方面。最引人注目的是,在所有情况下,为了使一个真正的系统工作,证明有必要用额外的预编程知识来补充基本算法。

提供额外的知识是有代价的:需要更多的人力和洞察力,随后系统的自主性就会降低。但同样清楚的是,对于诸如此类的任务,无知识的方法在机器人的有限生命周期内不会取得有价值的性能。

这种预先编程的知识采取了哪些形式?它包括对杂耍机器人策略的线性假设,手动将任务分解为两个移动机器人示例的子任务,而 box-pusher 还对 Q 值使用聚类技术,假设 Q 值局部一致。四个磁盘收集机器人还使用了手动离散化的状态空间。打包示例的维度要少得多,因此需要相应较弱的假设,但在过渡模型中,局部分段连续性的假设也可以大量减少所需的学习数据量。

探索策略也很有趣。杂耍者使用仔细的统计分析来判断在哪里可以进行实验。然而,这两个移动机器人应用程序都能够通过贪婪的探索很好地学习|总是在没有刻意探索的情况下利用。面对不确定性,包装任务采用乐观态度。这些策略都没有反映理论上最佳(但计算上难以处理)的探索,但都证明是足够的。

最后,还值得考虑这些实验的计算机制。
它们都非常不同,这表明各种强化学习算法的不同计算需求确实有一系列不同的应用。

杂耍者需要在每次击球之间以低延迟做出非常快速的决策,但在每次试验之间有很长的时间(30 秒或更长时间)以巩固在前一次试验中收集的经验并执行产生新反应所需的更积极的计算控制器在下一次试验。推箱机器人的目的是

自主运行数小时,因此必须以统一的长度控制周期做出决定。除了简单的 Q 学习备份之外,这个周期对于相当大量的计算来说是足够长的。四个磁盘收集机器人特别有趣。每个机器人的寿命都不到 20 分钟（由于电池限制）,这意味着大量的数字运算是不切实际的,任何重要的组合搜索都会占用机器人学习寿命的很大一部分。包装任务很容易受到限制。每隔几分钟就需要做出一个决定。除了对正在学习的转换模型执行大规模的基于交叉验证的优化之外,这还为在每个控制周期之间完全计算 200,000 个状态系统的最优值函数提供了机会。

目前在强化学习的实际实施方面正在进行大量进一步的工作。他们产生的洞察力和任务限制将对塑造未来开发的算法产生重要影响。

9. 结论

有多种强化学习技术可以有效地解决各种小问题。但这些技术中很少有能很好地解决更大的问题。这并不是因为研究人员在发明学习技术方面做得不好,而是因为在一般情况下解决任意问题非常困难。为了解决高度复杂的问题,我们必须放弃白板学习技术,并开始纳入偏见,这将为学习过程提供杠杆作用。

必要的偏见可以有多种形式,包括： 塑形:塑形技术用于训练动物（Hilgard & Bower,
1975）;老师首先提出非常简单的问题来解决,然后逐渐让学习者接触更复杂的问题。整形已用于监督学习系统,可用于自下而上训练分层强化学习系统 (Lin, 1991),并通过减少延迟直到问题得到充分理解来缓解延迟强化问题 (Dorigo & Colombetti,1994 年;Dorigo,1995 年）。

局部强化信号:只要有可能,应为代理提供局部强化信号。在可以计算梯度的应用程序中,奖励代理在梯度上的进步,而不仅仅是为了实现最终目标,可以显着加快学习速度（Mataric,1994）。

模仿:一个代理可以通过“观察”另一个代理执行任务来学习（Lin,1991）。对于真正的机器人,这需要尚不具备的感知能力。但另一种策略是让人类通过操纵杆或方向盘向机器人提供适当的运动指令（Pomerleau,1993）。

问题分解:将一个巨大的学习问题分解为一组较小的问题,并为子问题提供有用的强化信号,这是一种非常强大的偏置学习技术。机器人强化学习的最有趣的例子在一定程度上采用了这种技术（Connell & Mahadevan,
1993）。

re exes:让一无所知的智能体无法学习任何东西的一件事是,他们甚至很难找到空间中有趣的部分;他们徘徊

随机周围永远不会接近目标,否则他们总是会立即“被杀死”。
这些问题可以通过编写一组\re exes 来改善,这些\re exes 导致代理最初以某种合理的方式行动（Mataric，1994;Singh,Barto,Grupen,& Connolly,1994）。这些re exes 最终可以是被更详细和准确的学习知识覆盖，但它们至少让智能体在尝试学习时保持活力并指向正确的方向。Millan (1996) 最近的工作探索了使用 re exes 使机器人学习更安全、更安全有效的。

有适当的偏见,由人类程序员或教师提供,复杂的强化
学习问题最终会得到解决。还有很多工作要做,还有很多有趣的问题需要学习技术,特别是关于近似、分解和将偏差纳入问题的方法。

致谢

感谢 Marco Dorigo 和三位匿名审稿人的意见,这些评论有助于改进本文。还要感谢我们在强化学习社区中完成这项工作并向我们解释的许多同事。

Leslie Pack Kaelbling 部分得到了 NSF 拨款 IRI-9453383 和 IRI 9312395 的支持。Michael Littman 部分得到了 Bellcore 的支持。 Andrew Moore 得到了 NSF 研究启动奖和 3M 公司的部分支持。

参考

Ackley, DH 和 Littman, ML (1990)。强化学习中的泛化和缩放。 Touretzky, DS (Ed.), Advances in Neural Information Processing Systems 2, pp. 550{557 San Mateo, CA.摩根考夫曼。

阿不思,JS (1975)。一种新的机械手控制方法:小脑模型关节控制器 (cmac)。动态系统、测量与控制杂志,97、220{
227.

阿不思,JS (1981)。大脑、行为和机器人。 BYTE Books,新罕布什尔州彼得伯勒的 McGraw Hill 子公司。

安德森,CW（1986 年）。使用多层连接主义系统学习和解决问题。博士论文,马萨诸塞大学,阿默斯特,马萨诸塞州。

阿沙尔,RR (1994)。随机域中的分层学习。硕士论文,布朗
大学,普罗维登斯,罗德岛。

贝尔德,L. (1995)。残差算法:使用函数逼近的强化学习。在 Prieditis, A. 和 Russell, S. (Eds.),第十二届机器学习国际会议论文集,第 30 页{37 San Francisco, CA.摩根考夫曼。

Baird, LC 和 Klopf, AH (1993)。具有高维、连续动作的强化学习。技术。代表。 WL-TR-93-1147,俄亥俄州赖特-帕特森空军基地:赖特实验室。

Barto, AG, Bradtke, SJ 和 Singh, SP (1995)。学习使用实时动态编程来行动。人工智能,72 (1), 81{138.

Barto, AG, Sutton, RS 和 Anderson, CW (1983)。可以解决困难的学习控制问题的类似神经元的自适应元素。
IEEE Transactions on Systems, Man, and Cybernetics, SMC-13 (5), 834{846.

贝尔曼,R. (1957)。动态规划。普林斯顿大学出版社,新泽西州普林斯顿。

贝伦吉,人力资源部（1991 年）。空间智能控制的人工神经网络和近似推理。在美国控制会议上,第 1075 页{1080。

Berry, DA 和 Fristedt, B. (1985)。强盗问题:实验的顺序 Al 位置。
查普曼和霍尔,伦敦,英国。

Bertsekas,DP（1987 年）。动态规划:确定性和随机模型。
Prentice-Hall, Englewood Cli s, NJ。

Bertsekas,DP（1995 年）。动态规划和最优控制。雅典娜科学,
马萨诸塞州贝尔蒙特。第 1 卷和第 2 卷。

Bertsekas, DP, & Casta~non, DA (1989)。无限地平线动态规划的自适应聚合。 IEEE Transactions on
Automatic Control, 34 (6), 589{598.

Bertsekas, DP, & Tsitsiklis, JN (1989)。并行和分布式计算:数字
方法。 Prentice-Hall, Englewood Cli s, NJ。

Box, GEP, & Draper, NR (1987)。经验模型构建和响应曲面。
威利。

Boyan, JA 和 Moore, AW (1995)。强化学习中的泛化:安全地逼近价值函数。在 Tesauro, G., Touretzky, DS 和
Leen, TK
（编辑）,神经信息处理系统的进展 7 剑桥,马萨诸塞州。这
麻省理工学院出版社。

Burghes, D. 和 Graham, A. (1980)。控制理论导论,包括最优控制。埃利斯霍伍德。

Cassandra, AR, Kaelbling, LP 和 Littman, ML (1994)。在部分可观察的随机域中表现最佳。在第十二届全国人工
智能会议论文集上,华盛顿州西雅图。

Chapman, D. 和 Kaelbling, LP (1991)。延迟强化学习中的输入泛化:一种算法和性能比较。在国际人工智能联合会
议论文集上,澳大利亚悉尼。

克里斯曼,L. (1992)。带有感知混叠的强化学习:感知区分方法。在第十届全国人工智能智能会议论文集中,第 183 页
{188,加利福尼亚州圣何塞。 AAAI 出版社。

Chrisman, L. 和 Littman, M. (1993)。隐藏状态和短期记忆。在强化学习研讨会上的演讲,机器学习会议。

Cichosz, P., & Mulawka, JJ (1995)。具有截断时间差异的快速有效的强化学习。载于 Prieditis, A. 和 Russell, S.（编辑），第十二届机器学习国际会议论文集,第 99 页{107,加利福尼亚州旧金山。摩根考夫曼。

克利夫兰,WS 和 Delvin,SJ（1988 年）。局部加权回归:一种通过局部拟合进行回归分析的方法。美国统计协会杂志,83 (403), 596{610.

cli , D., & Ross, S. (1994)。向 ZCS 添加临时内存。适应性行为,3 (2)，101{150.

康登,A. (1992)。随机博弈的复杂性。信息与计算,96 (2), 203{224。

Connell, J. 和 Mahadevan, S. (1993)。真实机器人的快速任务学习。在机器人学习中。Kluwer 学术出版社。

Crites, RH 和 Barto, AG (1996)。使用强化学习提高电梯性能。在 Touretzky, D.、Mozer, M. 和 Hasselmo, M. (Eds.),神经信息处理系统 8。

达扬,P. (1992)。一般 TD( ) 的收敛性。机器学习,8 (3),341{362.

Dayan, P. 和 Hinton, GE (1993)。封建强化学习。在 Hanson, SJ, Cowan, JD, & Giles, CL (Eds.), Advances in Neural Information Processing Systems 5 San Mateo, CA。摩根考夫曼。

Dayan, P. 和 Sejnowski, TJ (1994)。 TD(·) 以 1 的概率收敛。机器学习荷兰国际集团,14 (3)。

Dean, T.、Kaelbling, LP、Kirman, J. 和 Nicholson, A. (1993)。在随机域中规划最后期限。在第十一届全国人工智能会议论文集上,华盛顿特区。

D Epenoux, F. (1963)。一个概率性的生产和库存问题。管理科学, 10, 98{108.

德曼 C. (1970)。有限状态马尔可夫决策过程。学术出版社,纽约。

Dorigo, M. 和 Bersini, H. (1994)。 q-learning 和分类系统的比较。在从动物到动画:第三届适应行为模拟国际会议论文集英国布莱顿。

Dorigo, M. 和 Colombetti, M. (1994)。机器人塑造:开发自主代理通过学习。人工智能,71 (2), 321{370.

多里戈,M. (1995)。 Alecsys 和 AutonoMouse:学习通过分布式分类系统控制真实机器人。机器学习,19。

菲希特,C.-N。 (1994)。高效的强化学习。在第七届 ACM 计算学习理论年度会议论文集中,第 88 页{97。计算机协会。

吉廷斯,JC (1989)。多臂强盗 Al 位置索引。 Wiley-Interscience 系列系统和优化。威利,奇切斯特,纽约。

戈德堡,D. (1989)。搜索、优化和机器学习中的遗传算法。艾迪生-韦斯利,马萨诸塞州。

戈登,GJ (1995)。动态规划中的稳定函数逼近。在 Priedi tis, A. 和 Russell, S. (Eds.),第十二届机器学习国际会议论文集,第 261 页{268,加利福尼亚州旧金山。摩根考夫曼。

Gullapalli, V. (1990)。一种用于学习实值函数的随机强化学习算法。神经网络,3, 671{692.

Gullapalli, V. (1992)。强化学习及其在控制中的应用。博士论文,马萨诸塞大学,阿默斯特,马萨诸塞州。

Hilgard, ER, & Bower, GH (1975)。学习理论（第四版）。 Prentice-Hall, Englewood Cli s, NJ。

何曼,AJ 和卡普,RM（1966 年）。关于非终止随机游戏。管理科学, 12, 359{370.

荷兰,JH (1975)。自然和人工系统的适应。密歇根大学出版社,安娜堡,密歇根。

霍华德,RA（1960）。动态规划和马尔可夫过程。麻省理工学院出版社,马萨诸塞州剑桥市。

Jaakkola, T., Jordan, MI 和 Singh, SP (1994)。关于随机迭代动态规划算法的收敛性。神经计算,6 (6)。

Jaakkola, T., Singh, SP 和 Jordan, MI (1995)。非马尔可夫决策问题中的蒙特卡罗强化学习。在 Tesauro, G., Touretzky, DS 和 Leen, TK（编辑）,神经信息处理系统的进展 7 剑桥,马萨诸塞州。麻省理工学院出版社。

凯尔布林,LP (1993a)。随机域中的分层学习:初步结果。在第十届机器学习国际会议论文集上,马萨诸塞州阿默斯特。摩根考夫曼。

凯尔布林,LP (1993b)。在嵌入式系统中学习。麻省理工学院出版社,马萨诸塞州剑桥市。

凯尔布林,LP (1994a)。关联强化学习:一种生成和测试算法。机器学习,15 (3)。

凯尔布林,LP (1994b)。关联强化学习:k-DNF 中的功能。机器学习,15 (3)。

柯尔曼,J. (1994)。通过域表征预测实时规划器性能。
博士论文,计算机科学系,布朗大学。

Koenig, S. 和 Simmons, RG (1993)。实时强化学习的复杂性分析。在第十一届全国人工智能会议论文集上,
第 99 页{105,加利福尼亚州门洛帕克。 AAAI 出版社/麻省理工学院出版社。

Kumar, PR 和 Varaiya, PP (1986)。随机系统:估计、识别和自适应控制。 Prentice Hall,新泽西州
Englewood Cli s。

李,CC（1991）。采用近似推理和神经网络概念的基于自学习规则的控制器。国际英特尔智能系统杂志,6 (1),
71{93。

林,L.-J。 (1991)。使用强化学习和教学对机器人进行编程。在第九届全国人工智能会议论文集上。

林,L.-J。 （1993a）。通过强化对机器人技能进行分层学习。在诉讼中
神经网络国际会议。

林,L.-J。 (1993b)。使用神经网络的机器人强化学习。博士论文,
宾夕法尼亚州匹兹堡卡内基梅隆大学。

Lin, L.-J. 和 Mitchell, TM (1992)。非马尔可夫域中强化学习的记忆方法。技术。代表。 CMU-CS-92-138,卡
内基梅隆大学计算机科学学院。

利特曼,ML (1994a)。马尔可夫游戏作为多智能体强化学习的框架。在第十一届机器学习国际会议论文集中,
第 157 页{163,加利福尼亚州旧金山。摩根考夫曼。

利特曼,ML (1994b)。无记忆策略:理论限制和实际结果。
在 Cli, D., Husbands, P., Meyer, J.-A., & Wilson, SW (Eds.), 从动物到动画 3:第三届国际适应性行
为模拟会议论文集,马萨诸塞州剑桥。麻省理工学院出版社。

Littman, ML, Cassandra, A. 和 Kaelbling, LP (1995a)。部分可观察环境的学习策略:扩大。在 Prieditis,
A. 和 Russell, S. (Eds.),第十二届机器学习国际会议论文集,第 362 页{370,加利福尼亚州旧金山。摩
根考夫曼。

Littman, ML, Dean, TL 和 Kaelbling, LP (1995b)。关于解决马尔可夫决策问题的复杂性。在第十一届关于
人工智能不确定性 (UAI{95) 蒙特利尔的年度会议论文集中,魁北克省,加拿大。

洛夫乔伊,WS（1991）。部分可观察马尔可夫决策过程的算法方法调查。运筹学年鉴,28、47{66。

Maes, P., & Brooks, RA (1990)。学习协调行为。在 Proceedings 第八届全国人工智能会议论文集上,第 796 页{802。摩根考夫曼。

Mahadevan, S. (1994)。在强化学习中打折或不打折:比较 R 学习和 Q 学习的案例研究。在第十一届国际机器学习会议论文集中,第 164 页{172,加利福尼亚州旧金山。摩根考夫曼。

Mahadevan, S. (1996)。平均奖励强化学习:基础、算法、
和实证结果。机器学习,22 (1)。

Mahadevan, S. 和 Connell, J. (1991a)。使用强化学习对基于行为的机器人进行自动编程。在第九届全国人工智能会议论文集上,加利福尼亚州阿纳海姆。

Mahadevan, S. 和 Connell, J. (1991b)。通过利用包含架构将强化学习扩展到机器人技术。在第八届国际机器学习研讨会论文集上,第 328 页{332。

马塔里克,MJ (1994)。加速学习的奖励功能。在 Cohen, WW 和 Hirsh, H. (Eds.),第十一届机器学习国际会议论文集。摩根考夫曼。

麦卡勒姆,AK（1995）。具有选择性感知和隐藏状态的强化学习。博士论文,计算机科学系,罗切斯特大学。

麦卡勒姆,RA（1993）。用有用的区分记忆克服不完整的知觉。
在第十届机器学习国际会议论文集中,第 190 页{ 196 阿默斯特,马萨诸塞州。摩根考夫曼。

麦卡勒姆,RA（1995）。隐藏状态强化学习的基于实例的实用区别。在第十二届国际会议机器学习论文集上,第 387 页{395,加利福尼亚州旧金山。摩根考夫曼。

Meeden, L.、McGraw, G. 和 Blank, D. (1993)。自动驾驶车辆中的紧急控制和规划。在 Touretsky, D. (Ed.), Proceedings of the Cognitive Science Society of the Cognitive Science Society, pp. 735{740. Lawerence Erlbaum Associates,新泽西州希尔斯戴尔。

米兰,J. d。河（1996）。导航策略的快速、安全和增量学习。 IEEE
系统、人与控制论交易,26 (3)。

通用电气公司的莫纳汉（1982 年）。部分可观察马尔可夫决策过程的调查:理论,
模型和算法。管理科学, 28, 1{16.

摩尔,AW（1991 年）。可变分辨率动态规划:在多元实值空间中有效地学习动作图。在过程中。第八届国际机器学习研讨会。

摩尔,AW（1994 年）。多维状态空间中可变分辨率强化学习的部分博弈算法。在 Cowan, JD, Tesauro, G. 和 Alspector, J.
(Eds.), Advances in Neural Information Processing Systems 6, pp. 711{718 San Mateo, CA. 摩根考夫曼。

Moore, AW 和 Atkeson, CG (1992)。基于记忆的函数逼近器用于学习控制的研究。技术。代表,麻省理工学院人工智能实验室,马萨诸塞州剑桥市。

Moore, AW 和 Atkeson, CG (1993)。优先扫描:用更少的数据和更少的实时性进行强化学习。机器学习,13。

Moore, AW, Atkeson, CG 和 Schaal, S. (1995)。基于记忆的学习控制。
技术。代表。 CMU-RI-TR-95-18,CMU 机器人研究所。

Narendra, K. 和 Thathachar, MAL (1989)。学习自动机:简介。
Prentice-Hall, Englewood Cli s, NJ。

Narendra, KS, & Thathachar, MAL (1974)。学习自动机|一项调查。 IEEE Transactions on Systems, Man, and Cybernetics, 4 (4), 323{334.

Peng, J. 和 Williams, RJ (1993)。 Dyna 框架内的高效学习和规划工作。适应性行为,1 (4), 437{454.

Peng, J. 和 Williams, RJ (1994)。增量多步 Q 学习。在第十一届机器学习国际会议论文集上,第 226 页{232,加利福尼亚州旧金山。摩根考夫曼。

Pomerleau, DA (1993)。用于移动机器人引导的神经网络感知。克鲁威学术出版社。

普特曼,ML (1994)。马尔可夫决策过程|离散随机动态规划。 John Wiley & Sons, Inc.,纽约,纽约。

Puterman, ML 和 Shin, MC (1978)。折扣马尔可夫决策过程的修改策略迭代算法。管理科学, 24, 1127{1137.

环,MB（1994 年）。强化环境中的持续学习。博士论文,
德克萨斯大学奥斯汀分校,德克萨斯州奥斯汀。

鲁德,美国（1993 年）。多级自适应方法的数学和计算技术。宾夕法尼亚州费城工业与应用数学学会。

Rumelhart, DE, & McClelland, JL (Eds.)。 （1986 年）。并行分布式处理:认知微观结构的探索。第 1 卷:基础。麻省理工学院出版社,马萨诸塞州剑桥市。

Rummery, GA 和 Niranjan, M. (1994)。使用联结系统的在线 Q 学习。
技术。代表。 CUED/F-INFENG/TR166,剑桥大学。

鲁斯特,J. (1996)。经济学中的数值动态规划。在计算机手册中
国家经济学。爱思唯尔,北荷兰。

Sage,美联社和怀特,CC（1977 年）。最佳系统控制。普伦蒂斯霍尔。

Salganico, M., & Ungar, LH (1995)。使用多臂老虎机分配指数在实值空间中进行主动探索和学习。在
Prieditis, A. 和 Russell, S.
（编辑）,第十二届机器学习国际会议论文集,第 480 页{487,加利福尼亚州旧金山。摩根考夫曼。

塞缪尔,阿拉巴马州（1959 年）。一些使用跳棋游戏的机器学习研究。 IBM 研究与开发杂志,3, 211{229.重
印于 EA Feigenbaum 和 J. Feldman,编辑,计算机与思想,麦格劳希尔,纽约 1963 年。

Schaal, S. 和 Atkeson, C. (1994)。机器人杂耍:基于记忆的学习的实现。控制系统杂志,14。

Schmidhuber, J. (1996)。一种在不受限制的环境中进行多智能体学习和增量自我改进的通用方法。在
Yao, X. (Ed.), 进化计算:理论与应用。科学出版社有限公司,新加坡。

施米德胡伯,JH (1991a)。好奇的模型构建控制系统。在过程中。国际神经网络联合会议,新加坡,卷。 2,第
1458 页{1463。 IEEE。

施米德胡伯,JH (1991b)。马尔可夫和非马尔可夫环境中的强化学习。 Lippman, DS, Moody, JE, &
Touretzky, DS (Eds.), Advances in Neural Information Processing Systems 3, pp. 500{506
San Mateo, CA.摩根考夫曼。

Schraudolph, NN, Dayan, P., & Sejnowski, TJ (1994)。围棋中位置评价的时间差分学习。在 Cowan,
JD, Tesauro, G., & Alspector, J. (Eds.), Advances in Neural Information Processing Systems
6, pp. 817{824 San Mateo, CA.摩根考夫曼。

Schrijver, A. (1986)。线性和整数规划理论。 Wiley-Interscience,新
纽约州约克。

施瓦茨,A.（1993 年）。一种最大化未折扣奖励的强化学习方法。在第十届机器学习国际会议论文集上,第
298 页{305,马萨诸塞州阿默斯特。摩根考夫曼。

Singh, SP, Barto, AG, Grupen, R. 和 Connolly, C. (1994)。运动规划中的鲁棒强化学习。在 Cowan, JD,
Tesauro, G. 和 Alspector, J. (Eds.), Advances in Neural Information Processing Systems
6, pp. 655{662 San Mateo, CA.
摩根考夫曼。

Singh, SP 和 Sutton, RS (1996)。通过替换资格痕迹来强化学习。
机器学习,22 (1)。

辛格,SP（1992a）。具有抽象模型层次结构的强化学习。在第十届全国人工智能智能会议论文集上,第 202 页{207,加利福尼亚州圣何塞。 AAAI 出版社。

辛格,SP（1992b）。通过组合元素顺序任务的解决方案来迁移学习。机器学习,8 (3),323{340。

辛格,SP（1993 年）。学习解决马尔可夫决策过程。博士论文,计算机科学系,马萨诸塞大学。此外,CMPSCI 技术报告 93-77。

斯滕格尔,RF（1986 年）。随机最优控制。约翰威利父子。

萨顿,RS (1996)。强化学习中的泛化:使用稀疏粗编码的成功示例。在 Touretzky, D.、Mozer, M. 和 Hasselmo, M. (Eds.),神经信息处理系统 8。

萨顿,RS (1984)。强化学习中的时间学分分配。博士论文，
马萨诸塞大学,阿默斯特,马萨诸塞州。

萨顿,RS (1988)。学习通过时间差异的方法进行预测。机器学习,3 (1),9{44。

萨顿,RS (1990)。基于近似动态规划的学习、规划和反应的集成架构。在第七届机器学习国际会议论文集上,德克萨斯州奥斯汀。摩根考夫曼。

萨顿,RS (1991)。通过增量动态规划进行规划。在第八届机器学习国际研讨会论文集上,第 353 页{357。摩根考夫曼。

Tesauro, G. (1992)。时间差分学习中的实际问题。机器学习,8，
257{277.

Tesauro, G. (1994)。 TD-Gammon,自学双陆棋程序,达到大师级
水平发挥。神经计算,6 (2), 215{219.

Tesauro, G. (1995)。时间差分学习和 TD-Gammon。 ACM 通讯,38 (3), 58{67.

Tham, C.-K. 和 Prager, RW (1994)。用于操纵器任务分解的模块化 q-learning 架构。在第十一届机器学习国际会议论文集上,加利福尼亚州旧金山。摩根考夫曼。

Thrun, S. (1995)。学习下国际象棋。在 Tesauro, G.、Touretzky, DS 和 Leen, TK (Eds.), Advances in Neural Information Processing Systems 7 Cambridge,
嘛。麻省理工学院出版社。

Thrun, S. 和 Schwartz, A. (1993)。使用函数逼近进行强化学习的问题。在 Mozer, M.、Smolensky, P.、Touretzky, D.、Elman, J. 和 Weigend, A.
（Eds.），1993 年连接主义模型暑期学校论文集，新泽西州希尔斯代尔。
劳伦斯·厄尔鲍姆。

特伦，某人（1992 年）。探索在学习控制中的作用。在 White, DA, & Sofge, DA (Eds.)，英特尔智能控制手册：神经、模糊和自适应方法。 Van Nostrand Reinhold，纽约，纽约。

齐齐克利斯，JN（1994 年）。异步随机逼近和 Q 学习。机器学习，16 (3)。

Tsitsiklis, JN 和 Van Roy, B. (1996)。基于特征的大规模动态规划方法。机器学习，22 (1)。

勇敢的，LG（1984）。可学习的理论。 ACM 通讯，27 (11)，
1134{1142.

沃特金斯，CJCH (1989)。从延迟奖励中学习。博士论文，英国剑桥国王学院。

Watkins, CJCH 和 Dayan, P. (1992)。 Q-学习。机器学习，8 (3)，279{292。

怀特黑德，SD（1991）。 Q 学习中的复杂性和合作性。在第八届机器学习国际研讨会论文集上，伊利诺伊州埃文斯顿。摩根考夫
曼。

威廉姆斯，RJ（1987 年）。一类用于神经网络中强化学习的梯度估计算法。在 IEEE 第一届神经网络国际会议论文集上，加利福尼亚州圣地亚哥。

威廉姆斯，RJ（1992 年）。连接主义的简单统计梯度跟随算法
强化学习。机器学习，8 (3)，229{256.

Williams, RJ 和 Baird, III, LC (1993a)。政策迭代的一些增量变体的分析：理解演员批评学习系统的第一步。技术。代表。
NU-CCS-93-11，东北大学计算机科学学院，马萨诸塞州波士顿。

Williams, RJ 和 Baird, III, LC (1993b)。基于不完美价值函数的贪婪策略的严格性能界限。技术。代表。
NU-CCS-93-14，东北大学计算机科学学院，马萨诸塞州波士顿。

威尔逊，S. (1995)。基于准确性的分类适应度。进化计算，3 (2)，
147{173.

Zhang, W. 和 Dietterich, TG (1995)。作业车间调度的强化学习方法。在国际人工智能联合会议论文集上。