强化学习与随机优化：序贯决策的通用框架

[美] 沃伦 • B. 鲍威尔(Warren B. Powell) 著
郭 涛 译

清華大學出版社
北 京

北京市版权局著作权合同登记号 图字：01-2024-3396

图书在版编目(CIP)数据

强化学习与随机优化 : 序贯决策的通用框架 /
(美) 沃伦·B.鲍威尔 (Warren B.Powell) 著 ; 郭涛译.
北京 : 清华大学出版社, 2025. 7. -- ISBN 978-7-302-69714-5

Ⅰ. TP181

中国国家版本馆CIP数据核字第2025LY6673号

责任编辑：王　军　刘远菁
封面设计：高娟妮
版式设计：恒复文化
责任校对：马遥遥
责任印制：沈　露

出版发行：清华大学出版社
网　　址：https://www.tup.com.cn，https://www.wqxuetang.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-83470000　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：涿州汇美亿浓印刷有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　印　　张：50.25　　字　　数：1070 千字
版　　次：2025 年 9 月第 1 版　　印　　次：2025 年 9 月第 1 次印刷
定　　价：256.00 元

产品编号：097949-01

译者序

强化学习是一种重要的机器学习范式，智能体通过与环境的交互，根据环境给予的奖励信号不断优化其动作策略，从而最大化累积回报。这一范式的兴起，推动了大模型与智能体时代的到来。近年来，基于人类反馈的强化学习(Reinforcement Learning from Human Feedback，RLHF)成为关键算法之一，它通过优化大模型的奖励模型，将人类的价值观与偏好纳入人工智能系统的学习过程，极大地提升了模型对人类意图的对齐能力。在此基础上，Google Research团队进一步提出了基于人工智能反馈的强化学习(Reinforcement Learning from AI Feedback，RLAIF)，这一方法为强化学习提供了新的可扩展途径，不再依赖高昂且耗时的人类标注收集，却依然能够获得与人类反馈相当的性能表现。值得一提的是，在DeepSeek-R1-Zero和DeepSeek-R1模型中，研究者直接应用了强化学习以及群体相对策略优化(Group Relative Policy Optimization，GRPO)等新型算法，显著增强了大模型的推理能力，标志着大模型的发展进入了新的阶段。

1. 为什么向读者推荐本书

在翻译、出版《深度强化学习图解》之后，我被强化学习中蕴含的数学建模思想深深吸引，由此萌生了进一步研读相关著作的念头。一次偶然的契机使我在Warren B. Powell教授团队CASTLE Labs的主页上发现了这本堪称“宝藏”的著作。在深入了解Powell教授的学术经历后，我更为其深厚的学识与卓越的贡献所折服。

Warren B. Powell教授曾任普林斯顿大学教授，是CASTLE Labs和PENSA的创始人。四十余年来，他在强化学习与随机优化领域作出了开创性的贡献。本书正是其长期研究与不断探索的结晶，历经十余年，不断发展与完善。其源头可追溯至2011年出版的*Approximate Dynamic Programming: Solving the Curses of Dimensionality*与2012年出版的*Optimal Learning*。这两部著作为动态优化与学习领域的重要成果奠定了基础。

在随后的十余年中，作者不断探索、思考并总结研究成果，提出了以“先建模、后求解”为核心的全新理念，并构建了“序贯决策”的通用框架。书中系统阐述了四类通用

策略(PFA、CFA、VFA、DLA)的设计与学习方法，涵盖混合学习与优化、机器学习与序贯决策的桥接、从确定性优化到随机优化、从单智能体到多智能体等广阔主题。近年来，Powell教授还在丰田北美总部的演讲中提出人工智能的七个层级，其中将“序贯决策”定位为第六级人工智能，认为其是支撑大模型(第四级人工智能)推理与智能决策的核心方法。他再次强调了“先建模、后求解”的理念及四类通用策略的价值与实现路径。

本书不仅在内容上具有经典性与权威性，也在方法论与实践路径上为计算随机优化与学习、大模型智能体优化提供了系统而深刻的框架。其所提出的决策与推理建模思想，既是学术探索的指引，也是实践落地的指南。无论是致力于大模型、智能体推理与优化等前沿领域的研究者，还是希望夯实理论、拓展视野的读者，都能从本书中获益良多。强烈推荐！

2. 如何使用本书

本书内容涵盖广泛的理论与数学公式，难度较高，常令读者在面对密集的推导与表达时汗流浃背甚至望而却步。为帮助读者更好地理解并掌握书中的思想，我在此提供学习路径与相关资源，协助读者循序渐进地进入本书的知识体系，并将其有效运用于学业与工作中。

1) 学习方法

(1) 整体把握，建立框架：建议读者首先关注作者提出的建模思想、框架、策略与实现路径，从整体上理解其技术体系与方法论。

(2) 专题研读，结合实践：在总体理解的基础上，选择某一专题深入研究，准确把握公式的理论含义，并通过Python或MATLAB编写代码，将理论与实践紧密结合起来。

(3) 迁移应用，发挥价值：结合自身研究方向或工作实践，将相关的建模思想与理论方法加以运用，力求真正发挥其价值。

2) 学习路线

(1) 通读本书，掌握范式：建议先通读本书，从整体上掌握作者的技术体系和方法论。

(2) 专题拓展，研读资源：针对感兴趣的专题，进一步学习作者提供的在线资源。①

(3) 延伸阅读，代码实践：重点推荐阅读作者的两部著作——*Sequential Decision Analytics and Modeling*与*A Modern Approach to Teaching an Introduction to Optimization*(扫描下方二维码即可延伸阅读)，并结合作者在GitHub发布的源码进行实战操作。

(4) 前沿动态，实时更新：若读者希望及时了解人工智能大模型与强化学习的最新技术进展，可扫描下方二维码，查看我整理的《AI大模型强化学习技术进展》PPT，以获得持续更新的参考资料。

① 扫描书中二维码，即可查看相应的拓展资源。

(5) 获取学习资源：本书作者提供了配套的PPT资源以及370个代码示例，扫描下方二维码即可下载。

PPT资源

代码示例

3. 为什么要翻译本书

读完本书后，我对作者的学术经历、研究成果以及全书的宏阔视野深感震撼。心念至此，既然作者已将自己与团队多年来在该领域积累的心血汇聚成书，我为何不将这一宝贵成果译为中文，使更多读者能够领略其中的思想与方法呢？

“道阻且长，行则将至；行而不辍，未来可期。”这句话正是本书翻译过程的真实写照。三年半的时间里，翻译之路漫长而艰辛。反复重译、不断打磨的过程几近无数；为追求术语的准确性与表达的地道性，不得不多次向业内学者请教。时而在心理压力下濒临崩溃，时而又从作者网站的思想与精神中汲取力量，不断重建信心，最终才得以完成本书的翻译工作。

原著堪称鸿篇巨制，也是迄今为止我翻译过篇幅最长、耗时最久的一部著作。由于书中涉及的强化学习与随机过程的理论体系宏大、术语纷繁，而国内学界对部分概念的译法尚存分歧，为确保本书术语的一致性，我在充分研读相关文献的基础上，参考了以下出版物的译法与表述：北京大学董豪等著的《深度强化学习：基础、研究与应用》、上海交通大学俞勇教授团队编写的《动手学强化学习》、中国科学院计算技术研究所赵地研究员团队翻译的《强化学习》、上海交通大学计算机科学与工程系俞凯教授翻译的《强化学习》(第2版)、刘次华编写的《随机过程》(第5版)、胡奇英等编写的《随机过程》(第2版)、刘澍编写的《随机过程》。此外，谨向清华大学出版社的编辑、校对与排版团队致以诚挚谢意，他们为保证本书的高质量出版付出了大量心血。

本书内容宏富而深刻，而译者水平有限，译文中难免有不足之处，诚恳期待各位读者批评、指正。

致谢

本书主要讲述序贯决策问题的建模框架，涉及搜索4类决策策略。我们之所以需要所有4类策略，是因为我们处理的问题涉及的领域广泛，包括货物运输(几乎所有模式)、能源、卫生、电子商务、金融，甚至材料科学。

关于序贯决策的研究涉及大量的计算工作，离不开CASTLE实验室的许多学生和工作人员的努力。在普林斯顿大学任教的39年生涯中，我与70名研究生和博士后同事以及9名专业人员朝夕相处，受益匪浅。衷心地感谢这群才华横溢的研究者所作的贡献，是他们使我有机会加入这个“通过计算方法解决广泛问题”的挑战。也正是这种问题的多样性激励着我研究解决问题的不同方法。在这个过程中，我遇到了来自“丛林”对面的研究者，通过阅读他们的论文，与他们交谈，甚至帮助他们攻克难题，我学会了他们的语言。

我还要感谢我在指导两百余篇高级论文的过程中所获得的感悟。虽然本科生的研究较为浅显，但他们确实帮助我接触到了更广泛的问题，这些主题涵盖了体育、卫生、城市交通、社交网络、农业、制药，甚至希腊货船优化等领域。2008年，正是这些本科生加快了我进入能源领域的步伐，使我得以尝试建模并解决各种问题，包括微电网、太阳能电池阵列、储能、需求管理和风暴应对等。这段经历使我接触到了新的挑战、新的方法，而最重要的是帮助我接触到了工程和经济的新领域。

鉴于CASTLE实验室的学生和工作人员太多，无法在这里全数列出，我特意在实验室网站中列出了一幅学术谱系图，并在此向名单上的每一个人致以最诚挚的感谢！

特别感谢CASTLE实验室的资助者，其中不乏众多的政府资助机构，如美国国家科学基金会、美国空军科学研究办公室、DARPA、美国能源部(经由哥伦比亚大学和特拉华大学引荐)和劳伦斯利弗莫尔国家实验室(我的首位能源领域的资助者)。特别感谢AFOSR的优化和离散数学项目，该项目为我提供了近30年的持续资助。我要向ODM的项目经理表示感谢，他们是Neal Glassman(帮助我启动了该项目)、Donald Hearn(向我引荐了材料科学项目)、Fariba Fahro(其对这项研究的热情决定了该项研究在AFOSR的生死存亡)和Warren Adams。感谢这些项目经理多年来所发挥的举足轻重的作用，正是他们将学术研究人员和

决策者(将研究成果出售给美国国会的人)连接起来。

我想感谢业界赞助商以及助力这项研究的专业人员。CASTLE实验室最鲜明的特点之一是，不仅撰写学术论文和运行计算机模拟，还在实地开展研究。我们会与某家公司合作，找出问题，建立一个模型，然后观察它是否有效，但它通常无效。这才是真正的研究，我曾经在一本名为*From the Laboratory to the Field, and Back*(《从实验室到现场，再回到实验室》)的小册子中记录了整个过程。正是这个反复的过程让我学会了如何建模和解决实际问题。我们在早期取得过一些成功，随后在解决更困难的问题时又经历了一段失败。但在21世纪初，我们取得了两项惊人的成功：在诺福克南部铁路使用近似动态规划实现了机车优化系统，并为施耐德公司(美国最大的卡车运输公司之一)提供了战略车队模拟器。该软件后来被授权给Optimal Dynamics，由其在卡车装载行业实施该技术。业界赞助商在助力我们的研究时没有得到任何保证，但是他们对我的(有时甚至是错位的)信心在我们的学习过程中发挥了至关重要的作用。

大学(尤其是普林斯顿这样的大学)研究实验室与业界合作，会带来少有人理解的管理方面的挑战。普林斯顿大学的资助官员John Ritter愿意就公司资助研究并获得软件授权的合同进行谈判，才使我能与业界达成合作。正是因为他们使用了软件，我才能了解哪些有效，哪些无效。John十分清楚，大学的首要任务是支持教师及其研究任务，而非提高许可费用。我想我可以自豪地说，我职业生涯中的5000万美元的研究经费给普林斯顿大学带来了不错的回报。

最后，我还要感谢一些专业人员的付出，是他们的努力使这些工业项目变得可能。其中表现最突出的是Hugo Simao，他是我指导的第一个博士生，毕业后在巴西任教，并于1990年回美国帮助创办了CASTLE实验室。Hugo贡献颇多，其中最重要的是其作为许多重大项目的首席开发人员，为实验室的发展奠定基础，尤其是与Yellow Freight System/YRC维持了长达数十年的关系。他也是为施耐德公司开发的获奖模型的首席开发人员，该模型后来被授权给Optimal Dynamics公司；此外，他还带领团队开发了用于模拟PJM电网的大型能源模型SMART-ISO，这已远远超出了研究生的能力范畴。而且从20世纪90年代开始，在工具还较粗糙的时期，Hugo就能把他的天赋应用于开发复杂系统。Hugo还曾在指导学生(研究生和本科生)处理软件项目的过程中发挥了重要作用，那时恰逢20世纪90年代，当许多人从Fortran语言过渡到C语言之时，我退出了编程界。Hugo的天赋、耐心和崇高的职业操守为CASTLE实验室的壮大奠定了良好的基础。后来加入实验室与Hugo并肩作战的还有Belgacem Bouzaiene Ayari，他在实验室工作了近20年，是诺福克南部铁路获奖项目的首席开发人员，作出过许多贡献。与业界赞助人合作所带来的价值无法用言语衡量，但可以肯定的是，如果没有像Hugo和Belgacem这样的天才研究人员，这项研究是万万不可能实现的。

Warren B. Powell

前言

本书浓缩了我毕生对序贯决策问题的研究，这可以追溯到1982年，当时我初次接触卡车(如优步/来福的卡车)装载运输中出现的问题，考虑到未来客户需求的高度随机性，包括运输整车货物的请求，我们必须权衡分配哪个司机来运输货物，以及哪些货物需要被运走。

我花了20年的时间才找到解决这个问题的实用算法，由此才出版了我的第一本关于近似动态规划的书(2007)，其主要突破是引入了决策后状态，并使用分层聚合来近似价值函数以解决这些高维问题。然而，我现在想说的是(当时我已意识到了这一点)，书中最重要的第5章仅仅提及了如何针对这些问题建模，而并没有提及解决问题的算法。当时，我确定了序贯决策问题的5个要素，从而得出了如下的目标函数：

$$\max_{\pi} \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^{\pi}(S_t))|S_0\right\}$$

直到该书第2版发行(2011)，我才意识到近似动态规划(具体来说是基于价值函数的策略)不是解决这些问题的唯一方法；相反，4类策略中只有一类使用价值函数。该书的2011年版列出了本书中描述的4类策略中的3类，但该书的大部分内容仍然侧重于近似价值函数。在2014年的论文*Clearing the Jungle of Stochastic Optimization*(《扫除随机优化“丛林”之障碍》)中，我才首次确定了现在使用的4类策略。之后，在2016年，我意识到这4类策略可以分成两种主要策略：策略搜索策略——搜索一系列函数以找到最有效的那个；前瞻策略——通过近似当前决策的下游影响来做出好的决策。

最后，我在2019年发表于*European Journal for Operational Research*(《欧洲运筹学杂志》)的一篇论文*A Unified Framework for Random Optimization*(《随机优化的统一框架》)中整合了这些想法，并且更充分地理解了以下主要问题：状态无关问题(包括基于导数的随机搜索和无导数随机搜索的纯学习问题)和更一般的状态相关问题；累积回报和最终回报目标函数；“任何自适应搜索算法都是一个序贯决策问题”。2019年论文中的材料实际上是本书的提纲。

本书以我2011年出版的聚焦于近似动态规划的书为基础，收录了上一本书的很多章节(部分章节改动巨大)，因此也可以称本书为“第3版”。不过，两个版本的框架完全不同。“近似动态规划”(approximate dynamic programming，ADP)这一术语仍然用于指代基于“近似处于某状态的下游价值”的理念来做决策。经过对此方法(其在本书中所占篇幅长达5章)的几十年的研究，我现在可以满怀信心地说，尽管价值函数近似(value function approximation)备受关注，但仅能处理极少的决策问题。

相反，我终于可以肯定：这4类策略具有普适性。这意味着任何决策方法都归属于这4类中的一类，或者算是两类或更多类的混合体。这将重点从算法(决策方法)转移到模型(特别是上述优化问题，以及状态转移函数和外生信息过程模型)上。这意味着，在设计决策策略之前，要先列出问题的要素。我称之为：

先建模，后求解。

研究序贯决策问题的各领域非常关注方法，我以前研究近似动态规划时也是如此。问题是，任何特定的方法本质上都局限于一类问题。在本书中，我演示了如何处理一个简单的库存问题，然后调整数据，以使4类策略中的每一类都能最有效地发挥作用。

这开辟了一种全新的方法来处理问题类。因此，在撰写本书的最后一年，我开始称之为“序贯决策分析”(sequential decision analytics)，其可以是由以下序列组成的任何问题：

决策、信息、决策、信息……

决策包括二元选择(出售资产)、离散选择(在计算机科学中备受青睐)，乃至运筹学中流行的高维资源分配问题。这种方法从一个问题开始，转移到建模不确定性这一挑战性任务，最后设计策略以做出优化某些指标的决策。该方法实用、可扩展且应用广泛。

能够创建一个跨越15个不同领域的通用框架，并使其代表解决序贯决策问题的所有可能方法，这无疑是令人兴奋的。有一种通用语言来模拟任何序贯决策问题，并结合4类策略的一般方法，这显然是有价值的，但这个框架是基于前人的成果而开发的。我不得不选择最优的符号和建模约定，但我的框架包含了为解决这些问题而开发的所有方法。我曾经与大量研究人员一样，只推广特定的算法策略。但我如今的目标是提升所有方法的知名度，从而使试图解决实际问题的人能够尽可能地使用最全的工具箱，而不是局限于某个特定领域开发的工具。

本书书名中的“强化学习”(reinforcement learning，RL)一词必须拿出来单独地讲一讲。在本书的撰写期间，人们对“强化学习”产生了极大的兴趣，它最初是以近似动态规划的形式出现的(我曾将ADP和RL喻为美式英语和英式英语)。然而，随着RL领域不断发展并开始致力于解决更棘手的问题，该领域人员与我和其他ADP研究人员得出了相同的结论：价值函数近似不是万能的——通常无法发挥作用。因此，RL领域开始尝试其他方法(正如我所做的那样)，如“策略梯度法”(英文为policy gradient method，我称之为策略函数近似)、上置信区间(英文为upper confidence bounding，成本函数近似的一种形式)、Q学习

(英文为Q-learning，基于价值函数近似生成策略)，以及蒙特卡洛树搜索(英文为Monte Carlo tree search，基于直接前瞻近似的策略)。所有这些方法都可以在Sutton和Barto的里程碑式代表作*Reinforcement Learning: An introduction*(《强化学习：导论》)的第2版中找到，但仅作为特定方法，而非一般的类。相较之下，本书更深入，并确定了一般类。

这种从一种核心方法到所有4类策略的演变正在“随机优化丛林”的其他领域中重复进行。随机搜索、模拟优化和老虎机问题的所有方法都来自这4类策略。随着时间的推移，我越来越清楚地意识到所有这些领域(包括强化学习)都在追随前人的研究，即最优控制(和随机控制)。最优控制领域率先引入并认真探索了价值函数近似(他们称之为代价函数，英文为cost-to-go function)、线性决策规则(策略函数近似的一种形式)和主力“模型预测控制”(简单的滚动时域法的“大名”，本书称之为“直接前瞻近似”)。我还发现，我的建模框架与最优控制相关文献中使用的框架最为接近，相较于其他大多数领域对转移函数概念的视而不见，最优控制是第一个引入这一功能强大的建模方法的领域。我做了一些小调整，例如使用状态S_t而非x_t；使用决策x_t(其广泛用于数学规划领域)而非u_t。

随后，我又引入了一个大的变化，以充分利用所有的4类策略。也许本书最重要的创新是打破了优化策略之间近乎自动的联系，然后假设将根据贝尔曼(Bellman)方程或哈密顿-雅可比(Hamilton-Jacobi)方程来计算最优策略。这些方程几乎不可用于计算实际问题，于是人们认为下一步自然是近似这些方程。然而，几十年的研究证明了这一点是错误的，人们已经开发出了不依赖于HJB方程的方法。我意识到，本研究的主体是通过将所有4类策略原理写入上述优化问题的原始语句来开发不同的策略。

不同领域的人需要花费一些时间学习这种通用语言。更有可能的是，现有的建模语言将适应这个框架。例如，最优控制领域可以保留该领域的符号，但要学会像前面展示的那样编写其目标函数，并意识到对策略的搜索需要跨越所有4个类(需要指出的是，该领域已经在使用了)。我希望采用离散动作符号a的强化学习领域能学会使用更通用的x(就像老虎机问题领域目前所做的那样)。

本书旨在吸引该领域的新手，以及具有处理决策和不确定性的一个或多个子领域背景知识的人；在撰写本书时，我意识到满足这两个广泛的群体无疑是最大的挑战。本书篇幅很长。我通过在许多章节中标记*来标明首次阅读时可以跳过的章节，从而方便新手阅读。我还希望本书能够获得各应用领域研究者的青睐。然而，本书主要面向意图通过建模应用程序并在软件中加以实现来解决实际问题的人。设计符号是为了便于编写计算机程序，其中数学模型和软件之间应该有直接的关系。在对信息流进行建模时，这一点尤为重要；不过，在主流强化学习相关论文中，这一点经常被忽视。

Warren B. Powell
新泽西州普林斯顿
2021年8月

目录

第Ⅰ部分 导 论

本书共有20章，分为6个部分。第Ⅰ部分包括4章，为本书的其余部分奠定了基础。

- 第1章介绍“序贯决策分析”的广泛应用领域及通用建模框架，该框架可将序贯决策问题简化为一种寻找决策的方法(规则)，我们称其为策略。
- 第2章介绍不同行业用到的15个主要的典型建模框架。这些行业都从不同的角度处理不确定条件下的序贯决策问题：使用8种不同的建模系统，通常集中于一个主要问题类别，并采用特定的解决方法。我们的建模框架将覆盖所有这些行业。
- 第3章介绍在线学习，重点是序贯学习与批量学习。此章可以被视为机器学习的简介，但几乎完全专注于自适应学习，这也是整本书的重点。
- 第4章将序贯决策问题分为3类，从而为本书的其余部分奠定基础：①可以使用确定性数学解决的问题；②可以使用样本合理近似随机性的问题(然后使用确定性数学解决)；③只能使用自适应学习算法解决的问题，这是本书其余部分的重点。

第1章概述了一个通用建模框架，涵盖所有序贯决策问题；提供了用于建模和解决序贯决策问题的整个框架的图景，这对于任何读者来说都应该是有价值的，无论他们对不确定条件下的决策掌握得如何；描述了问题的范围，简要介绍序贯决策问题的建模，并概述了用来解决这些问题的4类策略(决策方法)。

第2章总结了每个领域的典型建模框架，使用该领域的符号来解决某种形式的序贯决策问题。对该领域完全陌生的读者可以略读此章以初步了解已经采用的各种方法。具有更深背景知识的读者将在一个或多个典型问题上拥有一定程度的专业知识，这将有助于将该类问题与我们的框架联系起来。

第3章深入介绍了在线学习。可以略读此章，然后根据需要选择性地参考其中的内容。可以先阅读3.1节，然后略读其余小节的标题。本书将重复引用此章中的方法。

第4章将随机优化问题分为3类：

(1) 可以使用确定性数学精确解决的随机优化问题。

(2) 可以用固定样本表示不确定性的随机优化问题。这些问题仍然可以用确定性数学来解决。

(3) 只能使用序贯、自适应学习算法解决的随机优化问题。这将是本书其余部分的重点。

此章旨在提醒我们，有一些特殊的问题可以通过用采样近似值替换原始期望值来精确地解决。此章最后会介绍与学习问题相关的一些基本概念，包括对在线问题和离线问题进行重要的区分，并确定用于设计自适应学习策略的不同策略。

第1章

序贯决策问题

简单而言，一个序贯决策问题由以下序列组成：

决策、信息、决策、信息、决策……

做决策时，会产生成本或获得回报。我们面临的挑战是，如何表示将要到达的信息，以及如何在现在和将来做决策。本书的目标是对这些问题进行建模，并在新信息存在不确定性的情况下做出有效的决策。

解决序贯决策问题的第一步是了解正在做的决策。令人惊讶的是，从实验室的科学家到试图治疗重大疾病的人，在面对复杂问题时，往往无法确定自己面临的决策。

因此，我们想找到一种决策方法。英语中至少有45个词的意思相当于“决策方法”，但我们选用的是“策略”(policy)一词。该词常见于马尔可夫决策过程和强化学习等领域，但其解释范围比我们将使用的要窄得多。其他领域根本不使用该术语。设计有效的策略将是本书大部分内容的重点。

更难的是识别不确定性的不同来源。确定潜在的决策可能很难，但对于你正在管理的事情(无论是治疗疾病、管理库存还是进行投资)，要考虑所有可能带来影响的随机事件，几乎是不可能的。不确定性不仅来源广泛，其形式也千差万别。

在不确定的情况下做决策涉及一系列的分析问题，这些问题出现在工程、科学、商业、经济、金融、心理学、卫生、交通和能源等领域，既包括实验科学、医学决策、电子商务和体育领域出现的主动学习问题，即收集信息的决策，也包括机器学习中的随机搜索的迭代算法(找到最适合数据的模型或使用模拟器查找装配线的最优布局)，还包括双智能体博弈和多智能体系统。事实上，任何人类活动都包含序贯决策问题。

在不确定的情况下做决策是十分常见的事，这是每个人自两岁时第一次尝试新食物以来都必须做到的。以下是几个日常问题示例，我们必须处理这些问题的不确定性。

- 个人决策——包括从自动取款机中取多少钱，发现找工作的最优途径，以及决定何

时出发赴约。

- 食品购买——所有人都必须进食，而且不可能每天都去商店购物，所以必须提前决定什么时候去购物，以及购物时各类商品的购买量。
- 健康决策——例如，设计饮食和锻炼计划、进行年度体检、进行乳房X光检查和结肠镜检查。
- 投资决策——应该选择哪种共同基金？应该如何分配投资？应该为退休存多少钱？应该租房还是买房？

序贯决策问题无所不在、千姿百态。不确定情况下的决策几乎涵盖了所有主要领域。表1.1列出了问题领域和每个领域中可能出现的问题示例。毫不奇怪，已经出现了许多不同的分析领域来解决这些问题，通常使用不同的符号系统，并提出了适合每个特定环境中问题特征的解决方法。

本书将使用“先建模，后求解”的理念为分析序贯决策问题提供基础。虽然这对于确定性优化和机器学习等领域是标准的，但对于需要在不确定的情况下做决策的领域却完全相反。研究序贯决策问题的领域倾向于提出解决问题的方法，然后加以应用。这如同拿着锤子找钉子。

这种方法的局限性在于，已经开发的不同方法只能解决一部分问题。假设有一个最简单、最经典的序贯决策问题：管理产品库存以满足需求。设R_t是在t时的库存，x_t是订购的量(即时到达)，可满足需求$\hat{D}_{t+1}$(在t时未知)。库存R_t随时间的变化如下：

$$R_{t+1} = \max\{0, R_t + x_t - \hat{D}_{t+1}\} \tag{1.1}$$

表1.1 应用领域及对应的问题列表

领域	问题
商业	应该销售具有哪些功能的产品？应该使用哪种补给？应该要价多少？应该如何管理运载工具？哪个菜单最吸引顾客？
经济	鉴于经济状况，美联储应该拟定多高的利率？应提供何种水平的市场流动性？应该对投资银行实施什么指导准则？
金融	应该组合投资哪些股票？交易者应该如何针对潜在下跌风险进行套期保值？应该何时购入或出售资产？
互联网	应该展示哪些广告以实现广告点击量最大化？哪些电影最受青睐？何时/如何发送大规模通知？
工程	如何设计各种设备(从喷雾罐到电动汽车、桥梁到运输系统、晶体管到计算机)？
材料科学	应该使用什么样的温度、压力和浓度组合来创建具有最高强度的材料？
公共卫生	应该如何通过测试来评估疗效？应该如何分配疫苗？应针对哪些人群？
医学研究	什么样的分子结构会构成杀死最多癌细胞的药物？生产单壁纳米管需要采取哪些步骤？
供应链管理	应该什么时候从中国采购货物来补充库存？应该使用什么运输方式？应选择哪个供应商？
货物运输	应该让哪个驾驶员运输货物？卡车承运商应负责运送哪些货物？驾驶员应定居在何处？
信息采集	应该派无人机去何处收集有关野火或入侵物种的信息？应该测试什么药物来对抗疾病？

续表

领域	问题
多智能体系统	在寡头垄断的市场中，一家大公司应该如何竞标合同并预期竞争对手的反应？面对敌方的潜艇，我方潜艇应该如何应对？
算法	应该在搜索算法中使用什么步长规则？如何确定评估代价函数的下一个点？

对于上述库存管理问题，可以使用以下策略：当库存低于值$\theta^{\min}$时，订购足够的存货，使其增至$\theta^{\max}$。我们只需要确定参数向量$\theta=(\theta^{\min},\theta^{\max})$。策略很简单，但可能难以找到$\theta$的最优值。

接下来，考虑一系列复杂性逐渐增长的库存问题，先来看美国东南部一个仓库的库存采购问题。

(1) 采购的货物来自中国，可能需要90~150天才能到达。

(2) 必须满足季节性变化(以及圣诞节前后的巨大变化)的需求。

(3) 特殊订单可以空运，从而将运输时间缩短30天。

(4) 正在出售高价礼服，如果生产延迟或港口卸货延迟，必须特别警惕缺货风险。

(5) 礼服有不同的款式和颜色。如果某种颜色缺货，客户可能愿意接受其他颜色。

(6) 可以调整商品价格，但不清楚市场会有什么反应。可以一边调整价格一边观察市场反应来学习，以此指导未来的定价决策。

以上的每一项修改都会影响决策，这意味着在某种程度上修改了原始策略。

式(1.1)中的简单库存问题只有一个决策，x_t指定现在要订购多少货物。在面对实际问题时，可能会考虑一系列下游决策，包括：

- 订购量及交货承诺，如加急订单、正常订单、宽限期订单。
- 等待新货物到达时当前货物的定价。
- 未来货舱预订。
- 货船的行驶速度。
- 是否通过空运紧急增加库存，以填补因延误造成的空缺。
- 通过公路还是铁路将货物从港口运入仓库。

然后，对于至少要90天才能送达的产品，必须考虑其不同形式的不确定性：

- 完成制造的时间。
- 影响船速的天气性延误。
- 陆运延误。
- 到货时的产品质量。
- 汇率变化。
- 从现在到新货物抵达这段时间的库存需求。

如果你设置了一个简单的问题，例如式(1.1)，就永远不会考虑所有这些不同的决策和不确定性的来源。我们的演示将以丰富的建模框架为特色，再次强调我们的理念：

先建模，后求解。

本书将首次介绍适合所有序贯决策问题的通用建模框架，探讨4种被称为“策略”的、涵盖学术文献中或实践中使用的所有方法的决策类别。因为评估策略有多个维度(计算复杂性、透明度、灵活性、数据要求)，所以我们的目标并非总是选择性能最好的策略。然而，我们选择策略时仍将始终关注性能，这意味着目标函数的表述将是标准的。并非所有研究序贯决策问题的领域都是如此。

1.1 目标读者

本书的目标读者是那些想为存在不同形式不确定性的序贯决策问题开发实用、灵活、可扩展且可实现的模型的读者。终极目标是创建能够解决实际问题的软件工具。在本书中，精细的数学建模是将实际问题转化为软件的必要步骤。同时拥有这两个目标的读者将从我们的演示中获益最多。

鉴于此，我们发现，多个领域的专业人员可以使用本书，他们来自各种应用领域(工程、经济学和科学)及更注重方法论的领域(如机器学习、计算机科学、最优控制和运筹学)，对概率和统计、线性代数以及计算机编程都有一定程度的了解。

我们的演示强调建模和计算，尽量避免深入理论细节。若具备概率、统计学及线性代数的基础知识，则可轻松阅读本书的绝大多数内容。有时，我们会转向诸如资源分配问题(如管理不同血型的血液库存或资产投资组合)的更高维度的应用，对于此类问题，建议先熟悉一下线性、整数和(或)非线性规划。然而，这些问题都可以使用功能强大的求解器来解决，而不必了解这些算法的实际工作原理。

也就是说，对于数学背景知识深厚的高级博士生来说，不存在算法挑战和理论问题。

1.2 序贯决策问题领域

图1.1列出了序贯决策领域各个方法学派的一些著名书籍。表1.2按照大致诞生的时间顺序分别列出了多个领域，第2章中将作更深入的讨论。我们注意到，有两个不同的领域——基于导数的随机搜索和无导数随机搜索，都可以追溯到1951年发表的论文。

这些领域中的每一个都使用大约8个符号系统和一组重叠的算法策略来处理某种类型的序贯决策问题。每个领域都至少有一本代表作(通常是几本)和数千篇论文(更有甚者，每年会有数千篇论文)。每个领域都有最适合该领域的开发工具。

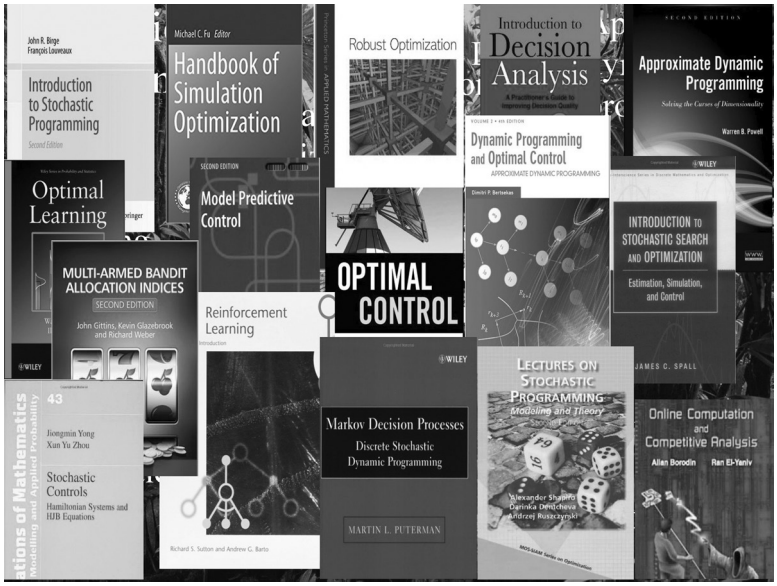

图1.1 代表随机优化中不同领域的主要书籍样本

表1.2 处理不确定性下的序贯决策的领域

(1) 基于导数的随机搜索	(9) 随机规划
(2) 无导数随机搜索	(10) 多臂老虎机问题
(3) 决策树	(11) 模拟优化
(4) 马尔可夫决策过程	(12) 主动学习
(5) 最优控制	(13) 机会约束规划
(6) 近似动态规划	(14) 模型预测控制
(7) 强化学习	(15) 鲁棒优化
(8) 最优停止	

领域的分化(以及不同的符号系统)不仅使得大家忽略了不同实践领域开发的共通性，还阻碍了思想的交叉融合。一个起初很简单的问题(如式(1.1)中的库存问题)可以使用诸如动态规划的特定策略来解决。但是，随着问题的现实性(和复杂性)不断增长，原始技术将束手无策，还需要到其他领域中寻找合适的方法。

我们将所有这些领域都归入“强化学习与随机优化”的范畴。“随机优化”通常指处理不确定性决策的分析领域。从“强化学习与随机优化”中的“强化学习”可以看出该领域正日益流行，并且该术语将被应用于解决序贯决策问题的一套不断扩展的方法。本书的目标是提供一个通用框架，以涵盖致力于解决这些问题的所有领域，而不是仅支持某一特定的方法。我们将这个更广泛的领域称为序贯决策分析(sequential decision analytics)。

序贯决策分析需要整合来自数学科学的3个核心领域的工具和概念。

(1) 统计机器学习——囊括了统计、机器学习和数据科学领域。这些工具的大部分(但不是全部)应用都涉及递归学习。我们还将涉足频率论和贝叶斯(Bayesian)统计领域，但仅限于此处提及的这些材料。

(2) 数学规划——该领域涵盖基于导数和无导数搜索算法的核心方法，用于从计算策略到优化策略参数的各种目的。有时，我们会遇到向量值决策问题，这些问题需要利用线性、整数和非线性规划工具。同理，所有这些方法都是在不假设具有随机优化背景的情况下介绍和提出的。

(3) 随机建模和模拟——在存在不确定性的情况下优化问题时通常需要对影响过程性能的不确定量进行精细的建模。我们会对蒙特卡洛(Monte Carlo)模拟方法进行基本介绍，但希望你具有概率和统计学的背景知识，包括贝叶斯定理的使用。

我们的演示不要求读者深入理解高级数学知识或任何方法领域，但我们将融合上述3个领域的概念和方法。与确定性问题的处理相比，不确定性问题的处理本质上更微妙，并且需要比机器学习更复杂的建模。

1.3 通用建模框架

整本书的核心是通用建模框架的使用，与确定性优化和机器学习中的做法相同。我们的框架主要基于最优控制中广泛使用的框架。这已被证明是最实用、最灵活的，并提供了数学模型与它在软件中的实现之间的明确关系。虽然大部分演示将侧重于建模序贯决策问题和开发决策的实用方法，但我们也认识到开发不同的不确定性来源模型的重要性(这一主题可以独立著书)。

此处仅概述通用建模框架，详细讨论参见第9章。通用建模框架的核心要素如下。

- 状态变量S_t。状态变量包含了我们需要知道的一切信息，以便我们做决策并对问题进行建模。状态变量包括物理状态变量R_t(无人机的位置、库存、股票投资)、我们完全了解的参数和数量的其他信息I_t(如当前价格和天气)，以及信念B_t，并以概率分布的形式描述我们不完全知道的参数和数量(这可能用来估计一种药物能将新患者的血糖降低多少，或者市场对价格会有怎样的反应)。
- 决策变量x_t。决策变量可以是二元的(持有或出售)、离散集合(药物、产品、路径)、连续变量(如价格或剂量)，以及离散和连续变量的向量。决策所受约束为$x_t \in \mathcal{X}_t$，我们用一种称为策略$X^\pi(S_t)$的方法来做决策，并为策略引入了符号，但将在完成模型之后再设计策略。这便是我们所说的“先建模，后求解”的基础。
- 外生信息W_{t+1}。这是在做出决策后所了解到的信息(市场对价格的反应、患者对药物的反应、穿越路径的时间)，而在做出决策时并不知道这些信息。外生信息来自正在建模的任何系统的外部。与此相对的是，决策是在过程内部做出的一种信息形

式，因此决策可以被认为是一个内生信息过程。

- 转移函数$S^M(S_t, x_t, W_{t+1})$。由更新状态变量的每个元素所需的公式组成。它涵盖了系统的所有动态，包括对序贯学习问题的估计和信念的更新。广泛应用于控制理论的转移函数用符号$f(x,u,w)$表示(针对状态x、控制u和信息w)；而我们的代表“状态转移模型”或“系统模型”的符号则替代了常用的字母$f(\cdot)$。
- 目标函数。它首先包括每个时间段的贡献(或回报[①]、成本等)，表示为$C(S_t, x_t)$，其中$x_t = X^\pi(S_t)$由策略决定，S_t是当前状态，由转移函数计算。正如本书后部将演示的那样，有多种不同的编写目标函数的方法，但最常用的方法是最大化累积贡献，其表达式为：

$$\max_\pi \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t))|S_0\right\} \tag{1.2}$$

其中，期望$\mathbb{E}$指“对所有类型的不确定性求平均值”，它可能是药效或者市场对价格做出反应的不确定性(在初始状态S_0下获得)，以及随着时间的推移带来的信息的不确定性$W_1, \ldots, W_t, \ldots$。策略的最大化仅仅意味着想要找到最优的决策方法。本书的大部分内容都致力于应对搜索策略的挑战。

确定了上述5个要素后，还有以下两个步骤要完成。

- 随机建模(也称为不确定性量化)。状态变量(包括初始状态S_0)中的参数和数量，以及外生信息过程$W_1, W_2, \ldots, W_t, \ldots$可能存在不确定性。某些情况下，可通过观察物理系统来避免对W_t过程建模。否则，将需要一个可能的W_{t+1}实现的数学模型，并且条件是给定S_t和决策x_t(任何一个都会影响W_{t+1})。
- 设计策略。只有完成建模之后，才能转而解决设计策略$X^\pi(S_t)$的问题。这是本书与随机优化丛林中所有书籍的出发点。我们不会在建立模型之前选择策略；相反，建模完成后，方才提供奔赴每个可能策略的路线图并教你如何在其中选择策略。

策略π由某种类型的函数$f \in \mathcal{F}$组成，可能具有可调参数$\theta \in \Theta^f$，其与函数f关联，其中策略将状态映射到决策。该策略通常包含函数中嵌入的优化问题。这意味着可以将式(1.2)改写成：

$$\max_{\pi\,=\,(f\in\mathcal{F},\theta\in\Theta^f)} \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t))|S_0\right\} \tag{1.3}$$

这就留下了一个问题：如何搜索函数？本书的大部分内容都致力于准确描述如何做到这一点。

---

① 译者注：在强化学习领域中，reward通常被译作“回报”或“奖励”，在本书中统一译作“回报”。

可以用上述符号修改序贯决策问题的原始表征——本章开头将该问题描述为“决策、信息、决策、信息……”，如以下序列所示：

$$(S_0, x_0, W_1, S_1, x_1, W_2, \ldots, S_t, x_t, W_{t+1}, \ldots, S_T)$$

其中使用三元组“状态、决策、新信息”来读取已知信息(状态变量S_t)、所做的决策x_t，及做出决策后了解的外生信息W_{t+1}。根据决策x_t赚取贡献$C(S_t, x_t)$(即获得回报或产生成本)，其中决策来自策略$X^\pi(S_t)$。

若有很多问题，则倾向于使用计数器n(第n个实验，第n个客户)，这种情况下，应将序贯决策问题写作：

$$(S^0, x^0, W^1, S^1, x^1, W^2, \ldots, S^n, x^n, W^{n+1}, \ldots, S^N)$$

在一些场景下甚至可以同时使用这两者，如$(S_t^n, x_t^n, W_{t+1}^n)$。例如，第$n$周第$t$小时的决策。

注意，存在由“决策、信息、停止”“决策、信息、决策、停止”和“信息、决策、信息、决策……”组成的问题，以及在无限时域内进行排序的问题。我们将有限序列用作默认模型。

可以使用前面的简单库存问题来讲述建模框架。

- 状态变量S_t。状态变量在最简单的问题中指的是库存R_t。
- 决策变量x_t。这是在t时所订购的数量。现在，假设它马上就能送达。我们还引入了策略$X^\pi(S_t)$，其中$x_t = X^\pi(S_t)$，将在创建模型后进行设计。
- 外生信息W_{t+1}。这是出现在t和$t+1$之间的需求$\hat{D}_{t+1}$。
- 转移函数$S^M(S_t, x_t, W_{t+1})$。这将是库存R_t的演变，由下式给出：

$$R_{t+1} = \max\{0, R_t + x_t - \hat{D}_{t+1}\} \tag{1.4}$$

- 目标函数。例如，在观察信息W_{t+1}后，倾向于编写单周期贡献函数，因为这包含需求$\hat{D}_{t+1}$，我们将以t周期内订购的库存x_t来满足该需求。因此，可将贡献函数写作：

$$C(S_t, x_t, W_{t+1}) = p \min\{R_t + x_t, \hat{D}_{t+1}\} - cx_t$$

其中，p是产品的销售价格，c是每件产品的成本。目标函数如下：

$$\max_\pi \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t), W_{t+1}) | S_0\right\}$$

其中，$x_t = X^\pi(S_t)$，且必须给出外生信息过程$W_1, \ldots, W_T$的一个模型。由于外生信息是随机的，因此必须取贡献总和的期望$\mathbb{E}$，以对信息过程中的所有可能结果求平均值。

接下来，利用第10章介绍的工具建立需求分布$\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_t, \ldots$的数学模型。

设计策略$X^\pi(S_t)$时可以参考学术文献，简单的库存问题的策略具有如下补充库存(order-up-to)的结构：

$$X^{Inv}(S_t|\theta)=\begin{cases}\theta^{\max}-R_t & R_t<\theta^{\min},\\ 0 & \text{其他情形}\end{cases} \tag{1.5}$$

这是一个参数化策略，需要求解下式以查找$\theta=(\theta^{\min},\theta^{\max})$：

$$\max_{\theta}\mathbb{E}\left\{\sum_{t=0}^{T}C(S_t,X^{Inv}(S_t|\theta),W_{t+1})|S_0\right\} \tag{1.6}$$

此处选择一个特定的策略类，然后在该类中进行优化。

注意，我们使用建模方法在数学模型和计算机软件之间建立了直接的关系。上面的每个变量都可以在计算机程序中直接翻译为变量名，唯一的例外是，期望算子必须替换为基于模拟的估计值(我们展示了如何做到这一点)。数学模型和计算机软件之间的这种关系在当前用于不确定性决策的大多数建模框架中并不存在，但最优控制除外。

前面曾对这个简单的库存问题进行过归纳。在继续阅读本书的过程中，我们将会展示如何使用5步通用建模框架为更复杂的问题建模。此外，我们将介绍涵盖解决更复杂问题的所有方法的4类策略。换言之，不仅我们的建模框架可用于对任何序贯决策问题进行建模，我们总结的4类策略也具备通用性：它们涵盖研究文献中研究过的或实践中使用过的所有方法。1.4节将概述这4类策略。

1.4　序贯决策问题的策略设计

用于区分随机优化领域的是解决问题所用的策略类型。本书中统一框架的最重要方面可能是如何识别和组织不同类别的策略。初步介绍参见第7章，详细介绍参见第11章——这也是本书其余内容的基础。本节仅简要介绍设计策略的方法。

关于如何在不确定性条件下做决策的全部文献可大致分为两种策略制定类型。

(1) 策略搜索。这包括需要搜索的所有策略：

- 用于做决策的不同类型的函数$f\in\mathcal{F}$。例如，式(1.5)中的补充库存策略就是一种非线性参数函数。
- 任何由函数f引入的可调参数$\theta\in\Theta^f$。式(1.5)中的$\theta=(\theta^{\min},\theta^{\max})$便是一个例子。

如果选择包含参数的策略，就必须找到参数θ的集以最大化(或最小化)诸如式(1.6)的目标函数。

(2)前瞻近似。此类策略的制定旨在允许我们根据决策的下游影响近似值做出最优决策。这些都是最受研究社区关注的策略种类。

补充库存策略$X^{Inv}(S_t|\theta)$是一个必须优化(也可以说是调优)的策略的好例子。可以使用模拟器(如式(1.6)所示)或在现场实际操作时进行优化。

这两种策略中的每一种都分别产生了两类子策略，从而生成了4类策略。下面详细介

绍这4类策略。

1.4.1 策略搜索

策略搜索类中的策略可以分为以下两个子类。

- 策略函数近似(policy function approximation，PFA)——这些是将状态(包括我们可以获得的所有信息)映射到决策的分析函数(式(1.5)中的补充库存策略为PFA)。详见第12章。
- 成本函数近似(cost function approximation，CFA)——CFA策略是参数化优化模型(通常是确定性优化模型)，其已被修改过，可帮助模型在不确定的情况下更好地随时间而响应。CFA策略中有一个嵌入的优化问题。此处仅将CFA当作主要的新策略类别来介绍，详细介绍请参阅第13章。

PFA涵盖将我们在状态变量中知晓的信息映射到决策的所有分析函数。这些分析函数有以下3种类型。

(1) 查找表。查找表用于离散状态S可以映射到离散动作的情况，例如：

- 如果患者是男性，60岁以上，血糖高，就开二甲双胍。
- 如果你的车在某个特定的十字路口，则向左转。

(2) 参数函数。参数函数可描述由参数向量θ作参数的任何分析函数。前面的补充库存策略就是一个简单例子。也可将其写作线性模型，例如：

$$X^{PFA}(S_t|\theta)=\theta_1\phi_1(S_t)+\theta_2\phi_2(S_t)+\theta_3\phi_3(S_t)+\theta_4\phi_4(S_t)$$

其中，$\phi_f(S_t)$是从状态变量信息中提取的特征。神经网络是另一种选项。

(3) 非参数函数。非参数函数包括局部线性近似函数或深度神经网络函数。

可以使用策略搜索优化的第二类函数被称为成本函数近似，或CFA，它们是参数化优化问题。在学习问题中使用的简单的CFA被称作区间估计(interval estimation)，可以用来确定哪个广告在网站上的点击量最大。设$\mathcal{X}=\{x_1,\ldots,x_M\}$为一组广告(可能有数千个)，而$\bar{\mu}_x^n$是目前对“(对所有广告)进行$n$次观察后单击广告$x$”概率的最优估计。然后设$\bar{\sigma}_x^n$是估计值$\bar{\mu}_x^n$的标准差。区间估计将使用以下策略选择下一个广告：

$$X^{CFA}(S^n|\theta)=\arg\max_{x\in\mathcal{X}}\left(\bar{\mu}_x^n+\theta\bar{\sigma}_x^n\right) \tag{1.7}$$

其中，$\arg\max_x$意味着找到使括号中的表达式最大化的x值。CFA的显著特征在于它需要解决一个嵌入的优化问题(广告的最大点击量)，并且有一个可调参数θ。

一旦引入了在策略中解决优化问题的想法(正如在式(1.7)中对策略的处理)，就可以解决任何参数化优化问题。不再局限于“x必须是一组离散选择中的一个”；它可以是一个大的整数规划，例如预留冗余时间以应对可能的天气性延误的航班时刻表规划，或者为预防电机故障而为明天的发电和电能储备制定的规划(这两个规划都是实践中使用的CFA的真实例子)。

1.4.2　基于前瞻近似的策略

做决策时倾向于考虑现在所做决定的下游影响，为此，可以采用以下两种方法。

(1) 价值函数近似(value function approximation，VFA)。这是解决序贯决策问题的一种流行方法，应用了动态规划(或马尔可夫决策过程)领域的原理。假设状态变量会告知必须在网络的何处做出决策，或者告知所持有的库存量。假设有人告知，如果在$t+1$时处于状态S_{t+1}(即处于网络中的某个节点或将拥有某个量级的库存)，也就是说$V_{t+1}(S_{t+1})$是处于状态S_{t+1}的“价值”，可以将其视为到达目的地的最短路径的成本，或从$t+1$时起获得的预期收益。

现在假设在t时处于状态S_t，并试图决定应该做何种决策x_t。做出决策x_t之后，观察随机变量W_{t+1}，可以得到$S_{t+1}=S^M(S_t,x_t,W_{t+1})$(例如，上面示例中的式(1.4))。假设已知$V_{t+1}(S_{t+1})$，可以通过求解下式找到状态$S_t$对应的值：

$$V_t(S_t)=\max_{x_t}\left(C(S_t,x_t)+\mathbb{E}_{W_{t+1}}\{V_{t+1}(S_{t+1})|S_t\}\right) \tag{1.8}$$

其中，最好将期望算子$\mathbb{E}_{W_{t+1}}$视作W_{t+1}所有结果的平均值。优化式(1.8)的x_t^*值是状态S_t的最优决策。第一期贡献$C(S_t,x_t^*)$加上未来的贡献$\mathbb{E}_{W_{t+1}}\{V_{t+1}(S_{t+1})|S_t\}$，得到现在处于状态$S_t$的价值$V_t(S_t)$。知道所有时间段和所有状态对应的价值$V_t(S_t)$时，就得到了一个基于VFA的策略，如下所示：

$$X_t^{VFA}(S_t)=\arg\max_{x_t}\left(C(S_t,x_t)+\mathbb{E}_{W_{t+1}}\{V_{t+1}(S_{t+1})|S_t\}\right) \tag{1.9}$$

其中，$\arg\max_{x_t}$返回使式(1.9)最大化的值x_t。

式(1.9)是计算最优策略的一个很不错的方法，但在实际问题中很少使用它来计算(第14章给出了一些可以精确求解的问题类)。出于此原因，许多领域都开发出了以近似动态规划、自适应动态规划或最明显的强化学习等命名的近似价值函数的方法。这些领域采用通过机器学习估计的近似函数$\overline{V}_{t+1}(S_{t+1})$来替换精确价值函数$V_{t+1}(S_{t+1})$。

基于VFA的策略引起了研究者的极大关注，并且可能是4类策略中最难的一个。本书将用4章(第15~18章)的篇幅讲解“近似”。

(2) 直接前瞻近似(direct lookahead approximation，DLA)。前瞻策略的最简单示例是导航系统，用来规划到目的地的路径，告知在下一个路口转向何处。当信息更新时，路径也会更新。

这是一个随机问题的确定性前瞻的例子。虽然确定性前瞻在某些应用中很有用，但很多情况下，做决策时必须明确考虑不确定性，这意味着必须在直接前瞻策略中解决随机优化问题！整个研究领域都非常关注处理不确定性条件下的直接前瞻近似模型的具体方法。建模和处理直接前瞻策略的通用框架参见第19章。

1.4.3 混合和匹配

可以通过混合多个类的策略来创建混合策略。可以创建前瞻策略的H个未来周期，然后使用价值函数近似来估算出规划时域结束时的状态。可以使用确定性前瞻，但须引入可调参数以使其在不确定性条件下的表现更强。可以结合PFA(将其视为建议决策的某种分析函数)，对来自PFA的决策的偏差进行加权，并将其添加到其他基于优化的策略中。在第19章中使用随机前瞻时，可能会同时用到所有4个类。

混合策略的一个例子是确定驶向目的地的路径和出发时间的策略。导航系统使用确定性前瞻，借助网络每条链路(即不同规划路径)的行程时间的“点估计”来解决最短路径问题。这条路径可能会产生40分钟的预估行程时间，但你什么时候才动身？现在你意识到了交通状况的不确定性，因此你可能决定添加一个缓冲区。当你重复行程时，你可以在评估预估方案的准确性时，向上或向下调整缓冲区。这是一种组合的直接前瞻(因为它计划了一条通向未来的路径)，具有可调的出发时间参数(使其成为PFA的一种形式)。

我们无法告诉你如何解决所有具体问题(多样性巨大)，但会给你一个完备的工具箱，并提供一些指导方针，帮助你做出选择。

1.4.4 4类的最优性

学术研究文献中普遍存在一种误解，认为式(1.8)(称为贝尔曼方程或哈密顿-雅可比方程)是创建最优策略的基础，任何好的(即接近最优的)策略的设计都必须从贝尔曼方程开始。这显然不正确。

4类策略中的任何一类都可能包含特定问题类的最优策略。出现的问题纯粹是计算问题。例如，对于绝大多数实际应用，贝尔曼方程(式(1.8))是不可计算的。尝试将式(1.8)中的真价值函数$V_{t+1}(S_{t+1})$替换为近似函数$\overline{V}_{t+1}(S_{t+1})$，效果可能很好，但许多情况下，它无法产生有效的策略。此外，一旦你开始使用价值函数近似，就会发现其他3类策略中的任何一种都可能同样有效或(通常)更好。这就是为什么随着时间的推移，许多人在存在新信息的情况下做决策时，没有使用过(甚至没有听说过)贝尔曼方程。

1.4.5 概述

我们认为，4类策略(PFA、CFA、VFA和DLA)是通用的，涵盖了前面列出的所有领域提出的所有方法，以及实践中使用的所有方法。

在这4个类别中，学术界主要关注VFA和各种形式的DLA(确定性和随机性)。相比之下，我们认为PFA和CFA在实践中的应用要广泛得多。特别是在学术界，CFA被广泛忽视了，却在实践中以一种特定的方式被广泛使用(通常不被调优)。PFA和CFA(即策略搜索类)在实践中是首选的，因为它们更简单，但正如你将反复看到的：

简单性的代价是参数可调，而调优很难！

1.5　学习

决策分析的一个重要部分涉及学习。传统的机器学习要求提供由输入x^n以及相关的响应y^n组成的数据集，然后找到一个可能是线性模型的函数$f(x|\theta)$，例如：

$$f(x|\theta)=\theta_0+\theta_1\phi_f(x)+\theta_2\phi_f(x)+...+\theta_F\phi_F(x)$$

其中，函数$\phi_f(x)$从x中的数据提取特征。输入x可能是文档中的单词、患者病历、天气数据或客户数据，例如个人数据和最近的购买历史，还可能是非线性模型、分层模型，甚至神经网络。之后，必须通过解决优化问题来拟合模型：

$$\min_{\theta}\frac{1}{N}\sum_{n=1}^{N}(y^n-f(x^n|\theta))^2$$

这就是经典的批量学习。

按顺序做决策时，也在进行序贯学习。假设有一个有病史h^n的病人；决定采用策略$X^\pi(S^n)$(其中S^n包括患者病史h^n)确定治疗方案$x^{treat,n}$。选择治疗方案后，静候以观察疗效，用y^{n+1}对其进行索引，其原因与做出决策x^n后观察W^{n+1}的原因相同。索引“n+1”表示这是未包含在任何用n索引的变量中的新信息。

状态变量S^n内的信念状态B^n包含用新观察y^{n+1}更新估计θ^n需要的所有信息。所有这些更新都隐藏在以下转移中：

$$S^{n+1}=S^M(S^n,x^n,W^{n+1})$$

正如y^{n+1}包含在W^{n+1}中。进行这种自适应更新的方法参见第3章“在线学习”，“在线学习”是机器学习领域使用的一个术语，用于序贯学习而非批量学习或分组学习。

在序贯决策分析中使用在线学习的条件如下。

(1) 计算函数的近似期望值$\mathbb{E}F(x,W)$，使其最大化。

(2) 创建近似策略$X^\pi(S|\theta)$。

(3) 计算处于状态S_t的近似价值，通常将其表示为$\overline{V}_t(S_t)$。

(4) 学习动态系统中的任何基础模型。其中包括：

① 用于描述过去如何影响未来活动的转移函数$S^M(S_t,x_t,W_{t+1})$。

② 成本或贡献函数：如果试图复刻人类的行为，则函数可能是未知的。

(5) 参数化成本函数近似计算，使用学习来修改策略中嵌入的目标函数和(或)约束。

评估这些函数的工具详见第3章，这些不同问题的具体设置则贯穿全书。

1.6 主题

我们的演示由一系列贯穿全书的主题组成。本节将介绍其中的一部分主题。

1.6.1 混合学习和优化

我们的应用程序通常会混合多个决策，其中有的决策会直接或间接地影响学习过程，有的决策会影响学习内容，有的决策会同时影响学习过程和学习内容。不妨思考以下三大类问题。

- 纯粹的学习问题——在这类问题中，决策只控制我们为学习而获取的信息。这可能出现在实验、计算机模拟甚至市场测试中。
- 无学习的状态相关问题——我们偶尔会遇到决策影响物理系统但不涉及学习的问题。使用导航系统告知转向就是这样的例子，即决策会影响物理系统(规划汽车的路径)，但没有学习。
- 混合问题——许多情况下，决策既会改变物理系统，又会影响我们为学习获取的信息。此外，还有一些多决策系统，例如分配疫苗的物理决策和指导信息收集(如疾病传播或药物疗效信息的收集)的测试决策。

1.6.2 将机器学习桥接到序贯决策

找到最优策略等同于找到实现最低成本、最高利润或最优性能的最优函数。这个类似的随机优化问题常见于统计学和机器学习中，其中一个常见的问题是使用数据集(x^n, y^n)，其中$x^n = (x_1^n, \ldots, x_K^n)$用于预测$y^n$。例如，可以指定以下形式的线性函数：

$$y^n = f(x^n|\theta) = \theta_0 + \theta_1 x_1^n + \ldots + \theta_K^n x_K^n + \epsilon^n \tag{1.10}$$

其中，ϵ^n是一个随机误差项，通常假设为正态分布，平均值为0，方差为σ^2。

求解下式，可以得到参数向量$\theta = (\theta_1, \ldots, \theta_K)$：

$$\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \left(y^n - f(x^n|\theta)\right)^2 \tag{1.11}$$

因此，将模型与数据拟合的问题涉及两个步骤。第一步是选择函数$f(x|\theta)$，这可以通过在式(1.10)中指定线性模型来完成(注意，这个模型之所以被称为“线性”，是因为它在θ中呈线性)。第二步涉及求解式(1.11)中给出的优化问题。唯一的区别是表现指标的具体选择。

接下来考虑如何处理序贯决策问题。假设正在最小化成本$C(S^n, x^n)$，该支出取决于我们的决策x^n以及状态变量S^n中涉及的其他信息。决策是根据策略$x^n = X^\pi(S^n|\theta)$(其参数为θ)做出的，该策略类似于在得知y^{n+1}之前用于预测(或估计)y^{n+1}的统计模型$f(x^n|\theta)$。目标函数应为：

$$\min_{\theta} \mathbb{E} \sum_{n=0}^{N-1} C(S^n, X^{\pi}(S^n|\theta)) \tag{1.12}$$

其中，$S^{n+1} = S^M(S^n, X^{\pi}(S^n), W^{n+1})$，此处的序列来源$(S^0, W^1, \ldots, W^N)$已知。

比较式(1.11)和式(1.12)时，不难发现两者都在搜索一组函数，以最小化某些指标。在统计建模中，指标需要数据集$(x^n, y^n)_{n=1}^N$，而决策问题只需要贡献(或成本)函数$C(S, x)$、转移函数$S^{n+1} = S^M(S^n, x^n, W^{n+1})$及外生信息过程的来源$W^1, \ldots, W^N$。用于搜索$\theta$以求解式(1.11)或式(1.12)的工具相同，但输入要求(训练数据集或物理问题模型)不同。

我们的统计模型可以采用多种形式中的任何一种，但它们都属于广泛的分析模型类别，可能是查找表、参数或非参数模型。所有这些类别的函数都属于4类策略中的一类，我们称之为策略函数近似。

表1.3简要对比了统计学习中的一些经典问题与随机优化中的类似问题。第一行对比了标准批量处理机器学习问题与典型随机优化问题(针对状态无关问题)。第二行对比了在线学习(必须在数据到达时适应数据)与在线决策。这两种情况都使用了期望，因为我们的目标是在下一次观察后立即做出期望有效的决策。最后，第三行清楚地表明，我们正在机器学习和随机优化中搜索函数，在这里我们使用的是典型的基于标准期望的目标函数形式。截至本书撰写之时，我们认为学术界才刚刚开始注意到这些关联，因此恳请读者帮忙留意将机器学习和序贯决策联系在一起的机会。

表1.3　统计学习中的经典问题与随机优化中的类似问题的比较

统计学习	随机优化
批量估计： $\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} (y^n - f(x^n\|\theta))^2$	样本平均近似： $\min_{x \in \mathcal{X}} \frac{1}{N} \sum_{n=1}^{N} F(x, W(\omega^n))$
在线学习： $\min_{\theta} \mathbb{E} F(Y - f(X\|\theta))^2$	随机搜索： $\min_{\theta} \mathbb{E} F(X, W)$
搜索函数： $\min_{f \in \mathcal{F}, \theta \in \Theta^f} \mathbb{E} F(Y - f(X\|\theta))^2$	策略搜索： $\min_{\pi} \mathbb{E} \sum_{t=0}^{T} C(S_t, X^{\pi}(S_t))$

1.6.3　从确定性优化到随机优化

我们的方法展示了如何将确定性问题推广到随机问题。假设我们正在解决前面的库存问题，尽管是从确定性模型开始，仍要使用标准的矩阵-向量数学来使符号尽可能保持紧凑。由于问题是确定性的，因此需要在不同的时段做出相应的决策$x_0, x_1, \ldots, x_t, \ldots$($x_t$可以是标量或向量)。设$C_t(x_t)$是在$t$时的贡献，由下式给出：

$$C_t(x_t) = p_t x_t$$

其中p_t是在t时的(已知)价格。此外，决策x_t还需要满足一组约束条件，通常写作：

$$A_t x_t = R_t \tag{1.13}$$

$$x_t \geq 0 \tag{1.14}$$

$$R_{t+1} = B_t x_t + \hat{R}_{t+1} \tag{1.15}$$

我们希望对下式求解：

$$\max_{x_0,\ldots,x_T} \sum_{t=0}^{T} C_t(x_t) \tag{1.16}$$

目的是满足式(1.13)~式(1.15)的约束。这是一个可以用多个包求解的数学规划。

现在假设希望$\hat{R}_{t+1}$是一个随机变量，这意味着它在$t+1$以前是未知的。此外，假设价格p_t随时间随机变化，这意味着直到$t+1$时才知晓p_{t+1}。这些变化将问题转化为不确定性条件下的序贯决策问题。

一些简单的步骤可将这个确定性优化问题转化为不确定性下的序贯优化问题。先将贡献函数写作：

$$C_t(S_t, x_t) = p_t x_t$$

其中，价格p_t是状态S_t中的随机信息。将目标函数改写成：

$$\max_{\pi} \mathbb{E}\left\{\sum_{t=0}^{T} C_t(S_t, X^{\pi}(S_t))|S_0\right\} \tag{1.17}$$

其中，$X^{\pi}(S_t)$必须做出满足约束条件式(1.13)~式(1.14)的决策。式(1.15)由转移函数$S^M(S_t, x_t, W_{t+1})$表示，其中W_{t+1}包括$\hat{R}_{t+1}$和更新的价格p_{t+1}。我们现在有一个正确建模的序贯决策问题。

可通过4项更改，将确定性优化公式转换为随机优化公式：

- 用函数(策略)$X^{\pi}(S_t)$替换每个x_t。
- 使贡献函数$C_t(x_t)$取决于状态S_t，以获取随时间随机演变的信息(如价格p_t)。
- 现在，取贡献总和的期望，因为变化$S_{t+1} = S^M(S_t, x_t, W_{t+1})$取决于随机变量$W_{t+1}$。可将期望算子$\mathbb{E}$视作信息过程的所有可能结果$W_1, \ldots, W_T$的平均值。
- 用$\max_{\pi}$替换$\max_{x_0,\ldots,x_T}$，这意味着从寻找最优决策集转向寻找最优策略集。

当存在不确定性时，须审慎地将确定性问题的约束转换为需要的格式。例如，若要分配资源，并且必须在一段时间内强加某项预算，可以用下式表示：

$$\sum_{t=0}^{T} x_t \leq B$$

其中，B是所有时间段使用的预算。这个约束不能直接用于随机问题，因为它假设我们同时“决定”所有变量——$x_0, x_1, \ldots, x_T$。当面对某个序贯决策问题时，必须按顺序做出这些决策，反映每个时间点的可用信息。必须递归地施加预算约束，如：

$$x_t \leq B - R_t \tag{1.18}$$

$$R_{t+1} = R_t + x_t \tag{1.19}$$

这种情况下，R_t将用作状态变量，策略$X^\pi(S_t)$必须反映约束式(1.18)，而约束式(1.19)由转移函数捕获。每个决策$x_t = X^\pi(S_t)$必须在做出决策时反映已知的情况(由S_t捕获)。

在实践中，期望值是很难计算的(通常是不可能计算的)，因此可采用熟知的蒙特卡洛模拟方法。这些方法详见第10章。这给我们遗留了设计策略的常见问题。为此，先来回顾一下1.4节的内容。

所有优化问题都涉及建模和算法的混合。对于整数规划(尤其是对于整数问题)，建模是很重要的，但算法设计往往比建模更重要。现代算法的强大之处在于，它们通常擅长处理建模策略，能够得到不错的效果(对于问题类)。

序贯决策问题则与此不同。

表1.4列举了确定性优化问题和随机优化问题的处理方法的一些主要差异。

表1.4　确定性优化与随机优化

处理方法	确定性优化	随机优化
(1) 模型	公式组	复杂函数、数值模拟、物理系统
(2) 目标	最小化成本	表现指标、风险指标
(3) 寻找之物	实值向量	函数(策略)
(4) 难点	设计算法	① 建模 ② 设计策略

(1) 模型。确定性模型是方程组。随机模型通常是复杂的方程组、数值模拟器，甚至是具有未知动态的物理系统。

(2) 目标。确定性模型会最小化或最大化某些定义明确的指标，如成本或利润。随机模型要求处理统计性能指标和风险等不确定性算子。许多随机动态问题非常复杂(如管理供应链、卡车运输公司、能源系统、医院、战疫)，涉及多个目标。

(3) 寻找之物。在确定性优化中，寻找确定性标量或向量。在随机优化中，几乎总是在寻找称为策略的函数。

(4) 难点。确定性优化的挑战是设计一个有效的算法。相比之下，随机优化最难的部分是建模。随机模型的设计和校准都可能非常困难。最优策略是罕见的，如果模型不正确，则策略不是最优的。

1.6.4　从单个智能体到多个智能体

本书最后将把这些想法扩展到多智能体系统。多智能体建模对于分解复杂系统(如不同供应商独立运营的供应链)以及大型运输网络(如卡车运输和铁路运输的主要运营商)非常有

效。多智能体建模在军事、对抗性环境(如国土安全)、具有少量竞争对手的寡头垄断市场及许多其他应用中至关重要。

多智能体建模在机器人、无人机和水下航行器等相关问题中非常重要，这些设备通常用于分布式信息收集。例如，无人机可以辨识野火受灾区域，以指导飞机和直升机投放阻燃剂。机器人可以探测地雷，水下航行器可以收集鱼类种群的信息。

知识领域分化的必然性致使多智能体设置几乎总是需要学习。这反过来又引入了通信和协调的维度，协调可以通过中央智能体进行，也可通过设计鼓励智能体合作的策略来解决。

本章对比了建模策略与应用最广泛的学习系统建模和算法框架(称为部分可观察的马尔可夫决策过程，或POMDP)。这是一个非常复杂的数学理论，不能推出可扩展的算法。我们将使用多智能体框架来说明转移函数的知识，然后利用上述4类策略来开发实用、可扩展、可实施的解决方案。

1.7 建模方法

建模框架中的5个元素(参见1.3节)可用于建模任何序贯决策问题，可以使用多种目标函数(稍后将进行介绍)。1.4节中的4类策略涵盖了序贯决策问题中可能用来做决策的所有方法。

这4类策略是1.3节中建模框架的核心。我们认为，用于为序贯决策问题(我们指的是任何序贯决策问题)做出决策的所有方法都将使用这4类中的一类(或两类及以上的混合)。1.2节中列出的领域使用的方法通常与特定的解决方案(有时不止一种)相关。与之相比，我们的方法更具通用性。

我们注意到，我们的方法与确定性优化中使用的方法非常相似，在确定性优化中，人们在搜索解决方案之前写出一个优化模型(带有决策变量、约束和目标)。这正是我们正在做的：在不指定策略的情况下写出模型，然后寻找有效的策略。我们称这种方法为：

先建模，后求解。

4类策略的通用性使得我们能够将设计模型的过程(参见1.3节)与模型的解决方案(即找到可接受的策略)分开。第7章将初次讲解这种方法在纯学习问题中的应用。接下来，第8章将介绍更丰富的应用，第9章会给出建模框架的大幅扩展版本。第10章介绍建模不确定性，第11章将更详细地回顾这4类策略。第12章至第19章将详细描述4类策略中的每一类，最后，第20章将过渡到多智能体系统。

1.8 如何阅读本书

本书的所有主题均参照概念从简单到复杂的逻辑顺序精心编排。本节实为本书的阅读指南。

1.8.1 主题编排

本书分为6个部分。

第 I 部分——引言和基础。首先总结了一些最常见的典型问题，然后介绍了贯穿全书的近似策略。

- 典型问题及其应用(第2章)——首先列出一系列不同领域所熟悉的典型问题，此过程中主要使用这些领域所熟悉的符号。不熟悉随机优化这一领域的读者可以略读此章。
- 在线学习(第3章)——大多数关于统计学习的书籍都侧重于批量处理应用程序，其中的模型适合静态数据集。本书中的学习主要是序贯的，在机器学习领域中被称为“在线学习”。我们对在线学习的使用完全是内生的，因为不需要外部数据集进行训练。
- 随机搜索简介(第4章)——从一个基本随机优化问题(该问题为大多数随机优化问题提供了基础)入手，还提供了其他一些准确解决某些问题的示例。随后，又介绍了一些处理采样模型的方法，然后过渡到自适应学习方法，这将是本书其余部分的重点。

第 II 部分——与状态无关的问题。有很多(无论出于何种原因)始终都不会随时间变化的优化问题。所有的“与状态无关的问题”都是纯粹的学习问题，因为我们的决策所导致的一切变化都源于我们对问题的信念。这些问题也称为随机搜索问题。本书第III部分研究更一般的状态相关问题，其中包括大规模的动态资源分配问题(决策改变资源的分配)，以及其他环境因素(例如变化的天气、市场价格、房间温度等)，其中，问题本身随时间演变。

- 基于导数的随机搜索(第5章)——基于导数的算法是最早为随机优化提出的自适应方法之一。这些方法构成了经典的(基于导数的)随机搜索或随机梯度算法的基础。
- 步长策略(第6章)——基于采样的算法需要使用通常所称的步长(或学习率)在新旧估计之间进行平滑化处理。步长策略在基于导数的随机搜索中起着关键作用，其中随机梯度决定了参数向量的改进方向，步长决定了梯度方向上移动的距离。
- 无导数随机搜索(第7章)——无导数随机搜索包含各种领域，如排名和选择(用于离线学习)、响应面方法和多臂老虎机问题(用于在线形式)。这一章展示了所有4类策略，用于决定下一步用何种策略对试图优化的函数进行(通常是有噪声的)观察。

第 III 部分——状态相关问题。这一部分将转而讲解更多类别的序贯问题，其中被优化的问题随着时间的推移而演变，这意味着此问题取决于随时间变化的信息或参数。这意味着目标函数和(或)约束取决于状态变量中的动态数据，其中该动态数据可以取决于正在做出的决策(例如库存或无人机的位置)，也可以只是根据外部因素(例如市场价格或天气)而演变。这些问题可能有(也可能没有)信念状态。

- 状态相关问题(第8章)——首先给出一系列与状态相关的问题。状态变量可能出现在目标函数(例如价格)或约束中，这常见于涉及物理资源管理的问题。然后介绍包含进化信念的问题，并引入主动学习维度(第7章首次提及)。
- 序贯决策问题建模(第9章)——此章全面总结了如何为一般(状态相关)序贯决策问题建模。首先通过简单问题演示建模框架，然后剖析深奥的复杂问题的建模框架。
- 不确定性建模(第10章)——好的策略需要一个好的不确定性模型，后者可以说是建模中最细微的层面。这一章确定了12种不同的不确定性来源，并讨论了如何对它们进行建模。
- 策略设计(第11章)——此章对创建策略的不同策略进行了更全面的阐释，从而引出了在本书第 I 部分中首次针对学习问题引入的4类策略。这一章还将指导如何针对特定问题挑选这4个类别，并介绍了一系列关于能量存储问题变化的实验结果，这些结果表明，可以根据数据的特性，使4类策略中的每一个都发挥最优作用。

第IV部分——基于策略搜索的策略。这一部分的内容描述了“策略搜索”类中必须在模拟器或现场的操作中进行调整的策略。

- PFA(policy function approximation)——策略函数近似(第12章)。这一章讲解了直接从状态变量映射到决策而不必解决嵌入优化问题的参数函数的使用(及其变化)。这是唯一一个不解决嵌入优化问题的问题类。可以在良好定义的参数空间中搜索，以找到在离线或在线环境中都能随时间产生最优表现的策略。PFA非常适用于具有标量动作空间或低维连续动作的问题。
- CFA(cost function approximation)——成本函数近似(第13章)。该策略涵盖了解决最优学习问题(也称为多臂老虎机问题)的有效策略，以及需要使用线性、整数或非线性规划求解器的高维问题的策略。这一类策略在研究文献中常被忽视，但在工业中被广泛使用(启发性地)。

第V部分——基于前瞻近似的策略。基于前瞻近似的策略与基于策略搜索的策略不相上下。这一部分将通过了解当前决策对未来的影响来设计好的策略。可以通过发现(通常是近似地)处于某种状态的价值，或者在某个时域内进行规划来做到这一点。

- VFA(value function approximation)——价值函数近似。这一类策略常见于以下文献：列出了针对不同特殊情况的各种精确方法的文献以及基于近似价值函数而作的文献，其中，近似价值函数由近似动态规划、自适应(或神经)动态规划和(最初)强化学习等术语描述。鉴于这一领域研究的深度和广度，本书将分5章介绍这类策略。

- 精确动态规划(第14章)。某些类别的序贯决策问题可以精确解决。其中最著名的特征是离散状态和动作(称为离散马尔可夫决策过程)，这是我们深入研究的主题。我们还会简要介绍最优控制文献中的一个重要问题——线性二次调节(linear quadratic regulation)，以及一些可以分析和解决的简单问题。
- 后向近似动态规划[①](第15章)。后向近似动态规划类似于经典的后向动态规划(第14章)，但不需要通过蒙特卡洛采样来枚举状态或计算期望值，并避免使用机器学习来近似估计价值函数。
- 前向近似动态规划[②]I：策略价值(第16章)。这是使用机器学习方法近似策略价值作为启动状态函数的第一步，也是被称为近似(或自适应)动态规划或强化学习的广泛方法的基础。
- 前向近似动态规划II：策略优化(第17章)。这一章基于以下基本算法：Q学习、价值迭代和策略迭代(在第14章中首次介绍)，尝试基于价值函数近似找到高质量的策略。
- 前向近似动态规划III：凸函数(第18章)。这一章重点讨论凸问题，特别强调在动态资源分配中应用的随机线性规划，利用凸性来构建价值函数的高质量近似。

- DLA(direct lookahead approximation)——直接前瞻近似(第19章)。直接前瞻策略在一定时域内进行优化，但我们允许引入各种近似，以使其更易于处理，而非优化原始模型。标准的近似是使模型具有确定性，这对某些应用非常有效。对于不太有效的应用，我们重新审视了解决随机优化问题的整个过程，但更加强调计算。

第VI部分——多智能体系统和学习。本书最后展示了如何将框架扩展到处理多智能体系统，而这本身就需要学习。

- 多智能体建模与学习(第20章)——首先展示了如何将学习系统建模为两个智能体问题(一个控制智能体观察一个环境智能体)，并展示了这如何为部分可观察的马尔可夫决策过程(partially observable Markov decision process，POMDP)生成替代框架。然后，扩展到多个控制智能体的问题，特别是需要通信建模的问题。

1.8.2　如何阅读每一章

本书主题范围甚广，因此内容繁多。然而，有些部分可以略读。章节中标有*的小节在初读时可以跳过。

① 译者注：backward approximate dynamic programming通常被译作“后向近似动态规划”或“反向近似动态规划”，在本书中统一译作“后向近似动态规划”。

② 译者注：forward approximate dynamic programming通常被译作“前向近似动态规划”或“正向近似动态规划”，在本书中统一译作“前向近似动态规划”。

标有**的小节表示材料涉及复杂的数学知识。数学功底不错的读者(尤其是具有测度-理论概率背景知识的读者)大多可以利用所学的全部知识来理解这一材料。尽管我们偶尔会略微提及这一材料，但毕竟本书不是为这些读者设计的。然而，我们的许多符号样式都是在理解概率论者如何思考和处理序贯决策问题的基础上设计的。本书将为希望以此为出发点进行更多理论研究的读者打下良好的基础。

初次接触序贯决策问题(指的是任何形式的动态规划、随机规划和随机控制)的读者应该从相对简单的“入门”模型开始。学习如何为相对简单的问题建模很容易。相比之下，为复杂的问题建模很难，特别是在开发随机模型时。重要的是找到处理起来得心应手的问题，然后由此精进。

本书将详细讨论4类策略。其中，两个相对简单(PFA和CFA)，两个较为复杂(VFA和随机DLA)。你不必马上成为所有这些策略的专家。随着时间的推移，每个人都要根据不断变化的信息做决策，而这些人中的绝大多数都从未听说过贝尔曼方程(基于VFA的策略)。此外，虽然确定性DLA(想想规划路径的导航系统)也相对易于理解，但随机DLA却是另一回事。理解策略的概念和调整策略(可以使用PFA和CFA实现)比立马翻阅学术文献中流行的更复杂的策略(VFA和随机DLA)更重要。

1.8.3 练习分类

每一章结尾都附有一系列练习，这些练习大致分为以下几类。

- 复习问题。这些问题相对简单，直接从各章中提取，不需要创造性地解决问题。
- 建模问题。这些都是描述应用程序的问题，之后必须放入前面提及的建模框架中。
- 计算练习。这些练习要求执行与各章所述方法相关的特定计算。
- 理论问题。我们会不时地提出经典理论问题。大多数关于随机优化的论著都会强调这些问题。本书强调建模和计算，因此理论问题的作用相对较小。
- 求解问题。这些问题将给出一个环境，需要你完成建模和策略设计。
- 来自*Sequential Decision Analytics and Modeling*(《序贯决策分析和建模》)的阅读材料——这是一本通过示例方式来进行教学的在线图书。每一章(第1章和第7章除外)都在讲述如何建模以及解决特定的决策问题，以展示不同类别策略的特点。这些练习中基本都有Python模块，以便读者进行计算。这些练习通常要求读者自开始时便使用Python模块，但需要额外的编程。
- 每日一问——这是一个自选问题，你将在每章结束时以此为背景来回答问题。这就好比“记日记”，因为你会基于全书的材料积累答案，但使用的是与你相关的问题设置。

并非每章的练习都包含上述所有类别。

1.9 参考文献注释

1.2节——第2章将探讨随机优化的不同领域，并概述相关文献。需要反复强调的是，我们的通用框架基于所有这些领域。

1.3节——Powell(2011)的著作(第5章)首次阐述了通用框架的5个要素(扫描右侧二维码即可查看)，它基于第1版的初始模型，初始模型具有6个元素(Powell(2007))。框架很大程度上借鉴了长期以来用于最优控制的框架(有很多相关书籍，请参阅Lewis & Vrabie(2012)的论著，这是该领域的一本受欢迎的参考文献)，但存在一些差异。将我们的框架与Powell(2021)的最优控制框架以及马尔可夫决策过程(现在是强化学习)中使用的框架进行比较。它们之间关键的区别在于，最初基于确定性控制的最优控制框架通常对控制$u_0, u_1, \ldots, u_T$进行优化，即使问题是随机的，也是如此。我们的符号表明，如果问题是随机的，则u_t是一种被我们称为策略的函数(控制人员称之为控制律)，我们总是对策略π进行优化。

1.4节——Powell(2011)似乎首次公开引用了用于解决动态规划的“4类策略”，但未列出本书中使用的4类(一类是短视策略，忽略了成本函数近似)。首次列出本书使用的4类策略的是Powell(2014)的*Clearing the Jungle of Stochastic Optimization*教程，但其没有意识到这4类策略可以(也应该)分为两个主要策略。Powell(2016)的教程给出了第一篇确定“策略搜索”和“前瞻策略”两种策略的论文。Powell(2019)提出了所有这些想法，将4类策略与状态无关和状态相关的问题类别以及累积回报和最终回报等不同类型的目标相结合。这篇论文为本书奠定了基础。

练习

复习问题

1.1 3类状态变量是什么？

1.2 序贯决策问题的5个要素是什么？

1.3 “先建模，后求解”是什么意思？

1.4 简单性的代价是什么？请举个例子，摘自本章或自行选择一个问题皆可。

1.5 为序贯决策问题设计策略的两种策略是什么？试简述每种策略的原则。

1.6 4类策略分别是什么？试简述每类策略。

建模问题

1.7 选择3个序贯决策问题的例子。试简述背景，并列出：

(1) 正在做的决策。

(2) 在做出决策后得到的可能与该决策相关的信息。

(3) 至少一个可用于评估该决策执行情况的指标。

1.8 对3种状态变量中的每一种执行以下操作：

(1) 给出3个物理状态变量的例子。

(2) 给出3个完全了解的参数或数量的信息示例，但这些信息不会被视为物理状态变量。

(3) 给出3个不完全知道但可以用概率分布估计的参数或数量的例子。

1.9 1.3节介绍了如何为简单的库存问题建模。重复这个模型，并假设以价格 p_t 销售产品。根据方程，价格在每个时间段都会发生变化：

$$p_{t+1} = p_t + \varepsilon_{t+1}$$

其中，ε_{t+1} 是平均值为0且方差为 σ^2 的正态分布随机变量。

求解问题

1.10 考虑资产出售问题，需要决定何时出售资产。设 p_t 是资产在 t 时出售的价格，并假设使用下式对资产价格的变化建模：

$$p_{t+1} = p_t + \theta(p_t - 60) + \varepsilon_{t+1}$$

假设噪声项 ε_t，$t = 1, 2, ...$ 是独立的并且随时间均匀分布，其中 $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$。设：

$$R_t = \begin{cases} 1 & \text{若 } t \text{ 时仍持有资产} \\ 0 & \text{其他} \end{cases}$$

进一步设：

$$x_t = \begin{cases} 1 & \text{若 } t \text{ 时出售资产} \\ 0 & \text{其他} \end{cases}$$

当然，只能在仍持有资产的情况下出售资产。现在需要一个规则来决定是否应该出售资产。假设：

$$X^\pi(S_t|\rho) = \begin{cases} 1 & \text{若 } p_t \geq \bar{p}_t + \rho \text{ and } R_t = 1 \\ 0 & \text{其他} \end{cases}$$

其中：

$$S_t = \text{可用于做决策的信息(必须设计)},$$

$$\bar{p}_t = .9\bar{p}_{t-1} + .1p_t$$

(1) 这个问题的状态变量 S_t 的元素是什么？

(2) 不确定性是什么？

(3) 假设在电子表格中运行一个模拟，在该模拟中，可以获得噪声项在T时间段的示例实现：$(\hat{\varepsilon})_{t=1}^{T} = (\hat{\varepsilon}_1, \hat{\varepsilon}_2, \ldots, \hat{\varepsilon}_T)$。注意，我们将$\hat{\varepsilon}_t$视作数字，例如$\hat{\varepsilon}_t = 1.67$，而$\varepsilon_t$是正态分布的随机变量。给定序列$(\hat{\varepsilon})_{t=1}^{T}$，编写用于计算策略价值的表达式$X^{\pi}(S_t|\rho)$。给定这一序列，我们可以评估$\rho$的不同值，如$\rho = 0.75$、2.35或3.15，以查看哪个效果最好。

(4) 实际上，并不会给定序列$(\hat{\varepsilon})_{t=1}^{T}$。假设$T = 20$个时间段，且：

$$\begin{aligned} \sigma_{\varepsilon}^2 &= 4^2, \\ p_0 &= \$65, \\ \theta &= 0.1 \end{aligned}$$

试列出策略价值作为期望(见1.3节)。

(5) 使用前面的参数，开发电子表格以创建序列的10个样本路径$((\varepsilon_t),\ t = 1, \ldots, 20)$。可以使用函数`NORM.INV(RAND(),0,`$\sigma$`)`生成$\varepsilon_t$的一个随机观察。设决策规则$X^{\pi}(S_t|\rho)$的表现取决于它决定的出售价格(如果它决定出售)，在所有10个样本路径上取平均值。现在测试$\rho = 1, 2, 3, 4, \ldots, 10$，并找到效果最好的$\rho$值。

(6) 重复(5)，但这次要解决以下问题：

$$\max_{x_0,\ldots,x_T} \mathbb{E} \sum_{t=0}^{T} p_t x_t$$

为此，在看到任何信息之前选择出售的时间t(即$x_t = 1$)。评估解决方案$x_2 = 1, x_4 = 1, \ldots, x_{20} = 1$。哪个最好？其表现与最优$\rho$值的表现$X^{\pi}(S_t|\rho)$有何区别？

(7) 最后，重复(6)，但现在可以看到所有的价格，然后选择最好的价格。这被称为后验界限(posterior bound)，因为它可以看到未来的所有信息，以便现在做出决策。(5)和(6)部分中的解与后验界限相比如何？(随机优化中有一整个领域都使用这种策略作为近似。)

(8) 根据1.5节中描述的分类，对(5)、(6)和(7)中的策略进行分类(是的，(7)是一类策略)。

1.11　库存问题描述了一种策略：如果库存低于$\theta^{\min}$，则进行订购，购至$\theta^{\max}$。这代表4个类别中的哪一个？写出为找到θ最优值所必须使用的目标函数。

序贯决策分析和建模

以下练习摘自在线书籍*Sequential Decision Analytics and Modeling*(《序贯决策分析和建模》)。扫描右侧二维码，即可查看该书。

1.12　阅读上述书籍中关于资产出售问题的第2章(2.1~2.4节)。

(1) 本书1.4节中介绍的4类策略中的哪一类用于解决此问题？

(2) 策略中使用了哪些可调参数？

(3) 试描述使用历史数据调整策略的过程。

每日一问

“每日一问”是你自行设计的一个将被用于本书其他章中“每日一问”的问题。

1.13 为本章选择一个问题背景。理想的问题应具有丰富性(例如，不同类型的决策和不确定性来源)，但最好的问题是你熟悉或特别感兴趣的问题。如果该序贯决策问题涉及某种形式的学习，将会有助于展示建模和算法框架的丰富性。目前，可以只准备一到两段的背景摘要，再在后面的章节中提供更多的信息。

参考文献

第2章

典型问题及其应用

大量的序贯决策问题至少出现在15个不同的领域(1.2节中已列出了这些领域)，这些领域各自都有针对这些问题的建模方法和解决方法。正如书面语和口语的起源不同，这些领域除了来自核心系统的专业用语符号，还有大约8种完全不同的符号系统。

隐藏在这些不同符号“语言”中的方法有些是真正的原创，还有一些是在原有方法之上进行了创造性改进，而其他的只是换了一个名称而已。这些不同方法均来源于激发各领域想象力的各种问题。不出意料，各个研究团体都会不断遇到新问题，从而诞生新想法。

本章2.1节将概述这些不同的领域及其各自的建模风格。全章采用各领域的符号简要介绍各领域最重要的典型模型。某些情况下，还将补充说明如何更换视角。2.2节将概述本书中使用的通用建模框架，该框架可用于对2.1节中的每个典型问题进行建模。最后，2.3节将概述不同应用的设置。

2.1 典型问题

随机优化中的每个领域都有一个典型的建模框架，以说明各领域的问题。通常，典型的问题都倾向于寻找一种巧妙的解决方法，就好比寻找钉子的锤子。虽然这些工具通常仅限于某个特定的问题类，但通常都阐释了一些重要思想，为强大近似方法奠定了基础。因此，理解这些典型问题有助于为不确定性条件下的所有序贯决策问题打下解题基础。

对上述所有领域都陌生的读者初读本书时可以略过这些典型问题。重要的是认识到，所有这些领域都在研究某种形式的序贯决策问题，而这个问题可以使用1.3节中首次提及的通用建模框架进行建模，详见2.2节。

2.1.1 随机搜索——基于导数和无导数

如果有一个问题能够概括几乎所有随机优化问题(至少能概括所有使用期望的问题)，那么这个问题通常被称为随机搜索，写作：

$$\max_x \mathbb{E}F(x,W) \tag{2.1}$$

其中，x是一个确定性变量，或者是一个向量(或者，正如将要展示的那样，是一个函数)。该期望与随机变量W相关，W可以是一个向量，也可以是一组随时间而变化的随机变量序列$W_1,\ldots,W_t,\ldots,W_T$。我们将式(2.1)中的期望所使用的符号形式称作简化式(compact form)，该符号并没有表明期望对象。

我们倾向于使用以下表达式，以明确该期望与随机变量的相关性：

$$\max_x \mathbb{E}_W F(x,W) \tag{2.2}$$

我们将式(2.2)中使用的形式称作期望的扩展式(expanded form)，该形式标明了对什么随机变量求期望。虽然概率论学者不赞成这种惯例，但终归应该提倡符号清晰化。我们还将介绍一些有助于表述与初始状态变量S^0的相关性的问题，状态变量可能包括有关市场如何响应价格变化之类的不确定参数的概率信念。我们通过下式来表达这种相关性：

$$\max_x \mathbb{E}\{F(x,W)|S^0\} = \max_x \mathbb{E}_{S^0}\mathbb{E}_{W|S^0}F(x,W) \tag{2.3}$$

初始状态变量可以表示问题与确定性或概率信息(例如，关于未知参数的分布)的相关性。例如，可以假设W呈正态分布，平均值为μ，其中μ也不确定(它可能均匀分布在0和10之间)。这种情况下，式(2.3)中的第一个期望——$\mathbb{E}_{S^0}$基于μ的均匀分布，而第二个期望——$\mathbb{E}_{W|S^0}$基于给定平均值μ的W正态分布。我们看到式(2.3)中的形式更好地传达了所涉及的不确定性。

每次解决问题时，初始状态S^0都可能发生变化，这本身就是问题。例如，S^0可能会捕捉患者的病历，之后必须选择一个治疗方案，然后观察疗效。我们有时会采用式(2.1)的简化式，但打算将式(2.3)中的扩展式用作默认式(当你开始处理实际应用时，会倾向于这么做)。

基于以下原因，这个基本问题类常以不同的形态出现。

- 初始状态S^0。初始状态将包括任何确定性参数，以及不确定参数的初始分布。S^0可能是一组固定的确定性参数(例如水沸腾时的温度)，或者它可能在每次解决问题时都会发生变化(可能包括实验室中的温度和湿度)，它还可能包括描述未知参数(例如市场对价格的反应)的概率分布。
- 决策x。x可以是二元的、离散的(有限且不太大)、分类的(有限但极可能有非常多的选择)、连续的(标量或向量)，也可以是离散向量。
- 随机信息W。W的分布可能已知，也可能未知，分布可能是正态的或指数型的，也

可能具有重尾、尖峰和罕见事件。W可能是一次实现的单个变量或向量，也可能是变量(或向量)序列$W_1,\dots,W_t,\dots,W_T$。

- 函数$F(x,W)$可以由几个维度来表征：
  - 函数评估的成本。函数$F(x,W)$可能极易评估(零点几秒到几秒)，或者成本较高(几分钟、几小时、几天或几周)。
  - 搜索预算。可能有限(例如，仅限于函数或其梯度的N个评估)，或无限(显然这纯粹出于分析目的——实际预算总是有限的)。甚至还有一些问题，其中规则决定了何时停止，这可能是外生的，也可能取决于我们所学的(这些问题被称为随时问题)。
  - 噪声级(以及噪声的性质)。有些应用的函数求值中的噪声最小(或不存在)，而其他应用的噪声级非常高。

本书的大部分内容将集中于更实用的有限成本版本，并在其上运行一个算法(称为π，原因稍后阐明)，迭代N次，以产生一个随机变量的解$x^{\pi,N}$，因为它取决于一段时间内对W的观察。

这个问题有两种类型。

- 最终回报目标。为此运行算法π，迭代N次，生成解$x^{\pi,N}$。我们只关心最终解的表现，而不关心在执行搜索时的表现。在找到$x^{\pi,N}$后必须对其进行评估，并引入一个用于测试的随机变量$\widehat{W}$(与训练相反)。最终回报目标函数(扩展式)如下：

$$\max_{\pi} \mathbb{E}_{S^0} E_{W^1,\dots,W^N|S^0} E_{\widehat{W}|S^0,x^{\pi,N}} F(x^{\pi,N},\widehat{W}) \tag{2.4}$$

- 累积回报目标。在此设置中，我们在执行搜索时关注总回报，这会产生目标函数：

$$\max_{\pi} \mathbb{E}_{S^0} \mathbb{E}_{W^1,\dots,W^N|S^0} \sum_{n=0}^{N-1} F(X^{\pi}(S^n),W^{n+1}) \tag{2.5}$$

基于算法策略，随机搜索的一般问题一直被视为两个不同的领域，它们分别是基于导数的随机搜索和无导数随机搜索。这两个领域的起源都可以追溯到1951年，但都作为完全独立的研究领域各自发展。

1. 基于导数的随机搜索

我们接受这样一个现实：不能对期望求导，这会阻止对$F(x) = \mathbb{E}F(x,W)$求导。然而，我们会在许多问题中观察W，然后对$F(x,W)$求导，并将其写作随机梯度：

$$\nabla_x F(x,W(\omega))$$

最常见的方法是用报童问题阐释随机梯度：

$$F(x,W) = p\min\{x,W\} - cx$$

随机梯度很容易被验证为：

$$\nabla_x F(x,W)=\begin{cases} p-c & x<W \\ -c & x>W \end{cases}$$

如你所见，可以在观察W之后计算$F(x,W)$的梯度，再在随机梯度算法中使用该梯度：

$$x^{n+1}=x^n+\alpha_n\nabla_x F(x^n,W^{n+1}) \tag{2.6}$$

其中，α_n称为步长。Robbins和Monro在1951年发表的一篇著名论文中证明了随机梯度算法(式(2.6))渐近收敛到目标函数(式(2.4))的最优值，这可以表示为：

$$\lim_{n\to\infty} x^n=x^*=\arg\max_x \mathbb{E}F(x,W)$$

70年后，这种算法的热度依然不减。第5章将详细介绍这个重要的类别，第6章则专门介绍如何设计α_n的步长公式。

2. 无导数随机搜索

有很多问题能计算随机梯度$\nabla_x F(x,W)$，但还有更多的问题无法计算它。相反，我们假设只能对函数$F(x,W)$进行随机观察，因此有：

$$\hat{F}^{n+1}=F(x^n,W^{n+1})$$

其中，索引表示首先选择x^n，然后观察W^{n+1}，之后计算函数$\hat{F}^{n+1}=F(x^n,W^{n+1})$的采样观察值。最后，使用采样观察值$\hat{F}^{n+1}$更新$\mathbb{E}F(x,W)$的估计值$\bar{F}^n(x)$，以获得$\bar{F}^{n+1}(x)$。

无导数随机搜索包括以下两个核心部分。

- 创建信念$\bar{F}^n(x)$。这可以使用第3章介绍的一系列机器学习工具中的任何一种来完成。
- 选择要观察的点x^n。这通常被称为算法，但在本书中，我们称其为策略。对于这个问题，第7章进行了较深入的讨论。

无导数随机搜索是一个非常丰富的问题类别，以至于有很多领域都在研究特定的算法策略，而不承认竞争方法。

2.1.2 决策树

无论是否存在不确定性，决策树显然都是描述序贯决策问题的最常见的方法之一。图2.1展示的是一个决定持有或出售资产的简单问题。如果决定持有，则会观察资产价格的变化，然后做出持有或出售的决策。

图2.1列出了决策树的基本元素。方形节点表示做出决策的点，而圆形节点表示显示随机信息的点。通过回滚计算每个节点的值来求解决策树。在结果节点，对所有下游节点取平均值(因为不控制转移到哪个节点)；而在决策节点，则基于一个周期的回报与下游价值的和来选择最优决策。

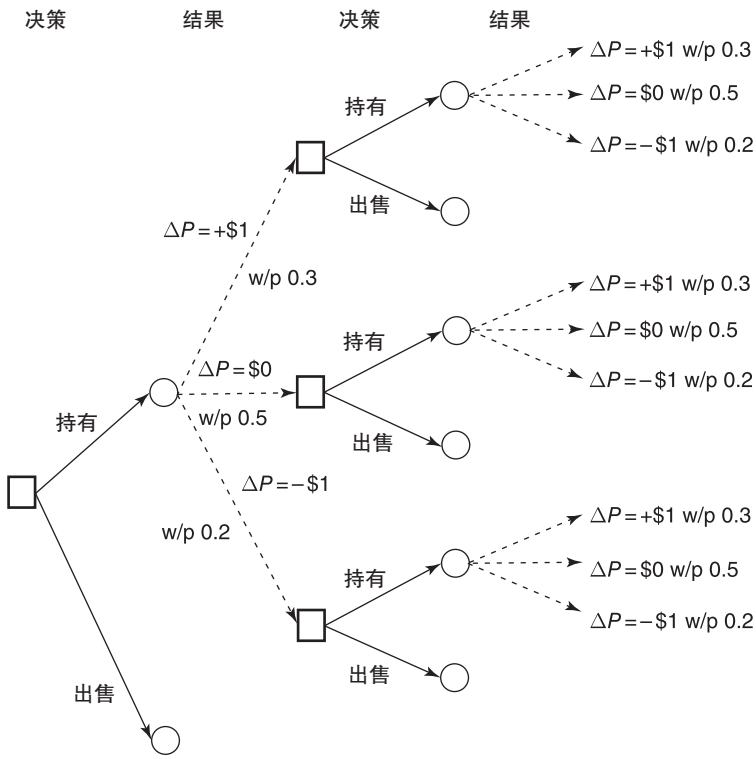

图2.1 决策树展示了决策(持有或出售资产)和新信息(价格变化)的序贯问题

几乎任何具有离散状态和动作的动态规划都可以建模为决策树。问题在于，决策树呈爆炸式增长，即使是对于较小的问题，也是如此。设想一个场景，其中有3个决策(购买、出售、持有资产)和3个随机结果(如价格变化：+1、−1或0)。价格变化及决策组成的每个序贯使树成长为原来的9倍。现在想象一个交易问题，每分钟做一次决策。仅仅一小时后，决策树的分支就扩大到$9^{60} \approx 1.8 \times 10^{57}$个！

2.1.3 马尔可夫决策过程

马尔可夫决策过程使用非常标准的框架进行建模，如图2.2所示。注意，这是在没有索引时间的情况下建模的，因为标准典型模型适用于稳定状态下的问题。在计算状态变量并选择动作时，一些研究者还考虑一组“决策迭代周期”，即时间点，通常建模为$t=1, 2, ...$。

例如，假设有$s \in \mathcal{S}$单位的库存，再采购$a \in \mathcal{A}_s$个单位，然后随机出售数量$\hat{D}$，用下式计算更新后的库存：

$$s' = \max\{0, s + a - \hat{D}\}$$

一步转移矩阵(one-step transition matrix)可以根据下式进行计算：

$$P(s'|s, a) = Prob[\hat{D} = \max\{0, (s + a) - s'\}]$$

状态空间——$\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$是系统可能占据的一组(离散状态)

动作空间——$\mathcal{A}_s = \{a_1, ..., a_M\}$是状态$s$下可以采取的一系列动作

转移矩阵——假设已给定了包含元素的一步状态转移矩阵

$P(s'|s, a)$ = 给定状态S_t等于s并采取动作a时状态S_{t+1}等于s'的概率

回报函数——设$r(s, a)$是当我们处于状态s并采取动作a时得到的回报

图2.2 马尔可夫决策过程的典型模型

回报函数可能是：

$$r(s, a) = p \min\{s + a, \hat{D}\} - ca$$

其中，c是购买库存物品的单位成本，p是为满足尽可能多的需求而定的销售价格。

如果要解决有限时域的问题，可设$V_t(S_t)$是处于状态S_t并从t时开始表现最优的最优价值。如果给定$V_{t+1}(S_{t+1})$，可用下式计算$V_t(S_t)$：

$$V_t(S_t) = \max_{a \in \mathcal{A}_s} \left(r(S_t, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|S_t, a) V_{t+1}(S_{t+1} = s') \right) \tag{2.7}$$

其中，γ是一个折扣因子(用于捕捉金钱的时间价值)。注意，要计算$V_t(S_t)$，就必须循环检查$S_t \in \mathcal{S}$的每个可能值，然后解决最大化问题。

式(2.7)可能看起来平淡无奇，但在其首次提出时，曾引起了巨大轰动，被称为运筹学和计算机科学中的贝尔曼最优性方程(Bellman's optimality equation)，或控制论中的哈密顿-雅可比方程(Hamilton-Jacobi equations)(尽管该领域通常将其用于连续状态和动作/控制)。

式(2.7)是一类主要策略的基础方程，我们称之为基于价值函数近似的策略(或VFA策略)。具体来说，如果知道$V_{t+1}(S_{t+1})$，就可以通过求解下式在t时和状态S_t下做出决策：

$$X_t^{\pi}(S_t) = \arg\max_{a \in \mathcal{A}_s} \left(r(S_t, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|S_t, a) V_{t+1}(S_{t+1} = s') \right)$$

如果使用式(2.7)精确计算价值函数，那么这将是一个罕见的最优策略实例。

如果一步转移矩阵$P(s'|S_t, a)$可以计算(并存储)，则非常容易从T时起开始计算式(2.7)(假设给定$V_T(S_T)$，则通常使用$V_T(S_T) = 0$)，并在时间上往回推进。

该领域对稳态问题表现出极大的兴趣，假设随着$t \to \infty$，$V_t(S_t) \to V(S)$。这种情况下，式(2.7)可改写为：

$$V(s) = \max_{a \in \mathcal{A}_s} \left(r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V(s') \right) \tag{2.8}$$

现在，得到一个方程组，必须解出来才能求得$V(s)$。详见第14章。

贝尔曼方程在首次提出时曾被视为一项重大的计算突破，因为它避免了决策树的爆炸性增长。然而，人们(包括贝尔曼本人)很快意识到，当状态s是一个向量(即使它仍然是离散的)时仍存在问题——状态空间的大小会随着维数的增加呈指数级增长，因此通常

将这种方法限制在状态变量最多具有3个或4个维度的问题上。这就是众人所知的“维数灾难”。

事实上，贝尔曼方程遭受了三种“维数灾难”。除了状态变量之外，随机信息W(藏在一步转移$P(s'|s,a)$中)也可以是向量；动作a也可能是向量x。人们通常会因为“维数灾难”而忽视“动态规划”(但它们意味着离散的马尔可夫决策过程)，但真正的问题是查找表的使用。有一些策略可以应对“维数灾难”，但较为困难，这也是本书如此之厚的原因。

2.1.4　最优控制

最优控制领域最为人熟知的是控制问题的确定性形式，通常用“系统模型”(转移函数)来描述：

$$x_{t+1} = f(x_t, u_t)$$

其中，x_t是状态变量，u_t是控制(或动作或决策)。一个典型的工程控制问题可能涉及火箭的控制(如使SpaceX在起飞后着陆)，状态x_t是火箭的位置和速度(每个都是三维的)，而控制u_t将是火箭所有维度上的力。力对火箭的位置和速度(即其状态x_t)的影响都包含在转移函数$f(x_t, u_t)$中。

转移函数$f(x_t, u_t)$是一个特别强大的符号，将在全书使用(我们将转移写作$S_{t+1}=S^M(S_t, x_t, W_{t+1})$)。它捕捉决策$x_t$(例如移到某个地点、添加库存、进行治疗或对车辆施加压力)对状态x_t的影响。注意，2.1.3节图2.2中描述的典型MDP框架使用了一步转移矩阵$P(S_{t+1}|S_t, a_t)$；第9章将详细讲解为何必须使用转移函数计算一步转移矩阵。在实践中，一步转移矩阵往往不可计算，而转移函数很容易计算。

问题是要找到u_t，以便求解：

$$\min_{u_0,\ldots,u_T} \sum_{t=0}^{T} L(x_t, u_t) + J_T(x_T) \tag{2.9}$$

其中，$L(x,u)$是“损失函数”，$J_T(x_T)$是终端成本。式(2.9)可以递归表示为：

$$J_t(x_t) = \max_{u_t} \left(L(x_t, u_t) + J_{t+1}(x_{t+1})\right) \tag{2.10}$$

其中，$x_{t+1} = f(x_t, u_t)$。这里，$J_t(x_t)$被称为代价(cost-to-go)函数，它只是2.1.3节中价值函数$V_t(S_t)$的不同表示法。

一个标准的解策略通常描述为模型的一部分，即将转移$x_{t+1} = f(x_t, u_t)$视为一种可以放松的约束，产生目标：

$$\min_{u_0,\ldots,u_T} \sum_{t=0}^{T} \left(L(x_t, u_t) + \lambda_t(x_{t+1} - f(x_t, u_t))\right) + J_T(x_T) \tag{2.11}$$

其中，λ_t是一组拉格朗日乘子，称为“共态变量”(co-state variable)。函数

$$H(x_0,u)=\sum_{t=0}^{T}\left(L(x_t,u_t)+\lambda_t(x_{t+1}-f(x_t,u_t))\right)+J_T(x_T)$$

被称为哈密顿量(Hamiltonian)。

式(2.9)中目标的一种常见形式是在状态x_t和控制u_t下的二次目标函数：

$$\min_{u_0,\ldots,u_T}\sum_{t=0}^{T}\left((x_t)^TQ_tx_t+(u_t)^TR_tu_t\right) \tag{2.12}$$

尽管这需要相当多的代数运算，但可以证明式(2.12)的最优解可以写作函数$U^\pi(x_t)$的形式，如下：

$$U^*(x_t)=-K_tx_t \tag{2.13}$$

其中，K_t是取决于矩阵$(Q_{t'},R_{t'}),t'\le t$的适当维度的矩阵。

这个理论的一个局限性是它很容易被推翻。例如，只需要添加一个非负约束$u_t\ge 0$，此结果即会失效。对目标函数进行任何更改，都可以得到相同的结论。这里还存在很多问题，其中目标在状态变量和决策变量中不是二次的。

有许多问题需要我们对流程如何随时间演变的不确定性进行建模。引入不确定性的最常见方法是使用转移函数，通常写作：

$$x_{t+1}=f(x_t,u_t,w_t) \tag{2.14}$$

其中，w_t在t时是随机的(这是最优控制相关文献中的标准符号，通常在连续时间内对问题进行建模)。w_t可能代表库存系统中的随机需求、从一个地点移到另一个地点时的随机成本，或确诊族群是否患病时的噪声。w_t经常被建模为附加噪声，写作：

$$x_{t+1}=f(x_t,u_t)+w_t \tag{2.15}$$

其中，w_t就像使火箭偏离轨道的风。

引入噪声时，通常将优化问题写作：

$$\min_{u_0,\ldots,u_T}\mathbb{E}\sum_{t=0}^{T}\left((x_t)^TQ_tx_t+(u_t)^TR_tu_t\right) \tag{2.16}$$

以上式子的问题是，必须认识到在t时的控制u_t是取决于状态x_t的随机变量，而状态x_t又取决于噪声项$w_0,\ldots,w_{t-1}$。

为将最初的确定性控制问题转化为随机控制问题，只须遵循1.6.3节提供的指导。先引入一个控制律(control law，最优控制的术语)，表示为$U^\pi(x_t)$(称之为策略)。现在的问题是找到求解下式的最优策略(“控制律”)：

$$\min_{\pi}\mathbb{E}_{w_0,\ldots,w_T}\sum_{t=0}^{T}\left((x_t)^TQ_tx_t+(U_t^\pi(x_t))^TR_tU_t^\pi(x_t)\right) \tag{2.17}$$

其中，x_t根据式(2.14)进行演化，且必须给定一个模型来描述随机变量w_t。本书将用大量篇幅重点讲述寻找好策略的方法。最优控制问题详见14.11节。

最优控制语言广泛应用于工程(主要面向确定性问题)和金融领域，但仅限于这些领域。然而，最优控制的符号将构成我们自己的建模框架的基础。

2.1.5　近似动态规划

近似动态规划的核心思想是使用机器学习方法代替价值函数$V_t(S_t)$(见式(2.7))，近似值为$\overline{V}_t^n(S_t|\theta)$(假设是在$n$次迭代后)。可以使用第3章中介绍的各种近似策略中的任何一种。设a_t是t时的决策(例如订购多少货物或开什么药)。设$\bar{\theta}^n$是在n次更新之后对θ的估计。假设我们处于状态S_t^n(这可能是在第n次迭代期间t时的库存)，可以使用这种近似来创建处于状态S_t^n的价值的采样观察：

$$\hat{v}_t^n = \max_{a_t}\left(C(S_t^n, a_t) + \mathbb{E}_{W_{t+1}}\{\overline{V}_{t+1}(S_{t+1}^n|\bar{\theta}^{n-1})|S_t^n\}\right) \tag{2.18}$$

其中，$S_{t+1}^n = S^M(S_t^n, a_t, W_{t+1})$，$\bar{\theta}^{n-1}$是在$n-1$次迭代之后对$\theta$的估计。

然后可以使用$\hat{v}_t^n$更新估计$\bar{\theta}_t^{n-1}$，以获得$\bar{\theta}_t^n$。这取决于如何近似$\overline{V}_t^n(S_t|\theta)$(第3章介绍了多种方法)。此外，可用其他方法获得采样观察$\hat{v}_t^n$，详见第16章。

给定价值函数近似$\overline{V}_{t+1}(S_{t+1}^n|\bar{\theta}^{n-1})$，有一种使用下式进行决策的方法(即策略)：

$$A^\pi(S_t^n) = \arg\max_{a_t}\left(C(S_t^n, a_t) + \mathbb{E}_{W_{t+1}}\{\overline{V}_{t+1}(S_{t+1}^n|\bar{\theta}^{n-1})|S_t^n\}\right)$$

其中，$\arg\max_{a_t}$返回的a值使表达式最大化。这就是我们所说的基于VFA的策略。

使用近似价值函数的想法最初由贝尔曼于1959年提出，随后应用于各界，并有了新的发展。20世纪70年代，最优控制领域使用神经网络近似连续价值函数；20世纪80年代和90年代，计算机科学将其称为强化学习。第16章和第17章将深入讨论这些方法。此想法还适用于随机资源分配问题，人们为此开发了利用价值函数凹性(当最大化时)的方法(见第18章)。

2.1.6　强化学习

当控制领域开发使用神经网络近似价值函数的方法时，两位计算机科学家——Andy Barto和他的学生Richard Sutton尝试模拟动物行为，就像老鼠试图找到走出迷宫的路来获得回报一样(见图2.3)。通过捕捉迷宫中从特定点出发的某一路径最终通向成功的概率，可以随着时间的推移而学习成功。

该基本思想与近似动态规划的方法非常相似，但它有自己独特的形式。强化学习的核心算法策略包括学习处于状态s并采取动作a的价值$Q(s,a)$，而非学习处于状态s的价值$V(s)$。通过计算下式进行的基本算法称为Q学习：

$$\hat{q}^n(s^n, a^n) = r(s^n, a^n) + \lambda \max_{a'} \bar{Q}^{n-1}(s', a') \tag{2.19}$$

$$\bar{Q}^n(s^n, a^n) = (1-\alpha_{n-1})\bar{Q}^{n-1}(s^n, a^n) + \alpha_{n-1}\hat{q}^n(s^n, a^n) \tag{2.20}$$

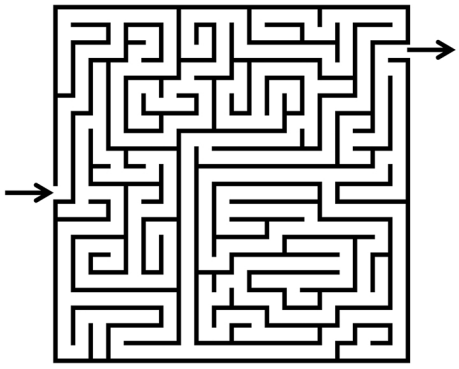

图2.3 寻找走出迷宫的路径

在这里，λ是一个折扣因子，但它不同于我们在解决动态问题(如式(2.7))时(偶尔)使用的折扣因子γ。参数λ是所谓的“算法折扣因子”，因为它有助于为未来犯错的影响打“折扣”，而这些错误会(错误地)降低处于状态s^n并采取动作a^n的价值。

更新式(2.20)有时可写作：

$$\begin{aligned}\bar{Q}^n(s^n,a^n) &= \bar{Q}^{n-1}(s^n,a^n)+\alpha_{n-1}(\hat{q}^n(s^n,a^n)-\bar{Q}^{n-1}(s^n,a^n)) \\ &= \bar{Q}^{n-1}(s^n,a^n)+\alpha_{n-1}\underbrace{(r(s^n,a^n)+\lambda\max_{a'}\bar{Q}^{n-1}(s',a')-\bar{Q}^{n-1}(s^n,a^n))}_{\delta}\end{aligned} \tag{2.21}$$

其中：

$$\delta = r(s^n,a^n)+\lambda\max_{a'}\bar{Q}^{n-1}(s',a')-\bar{Q}^{n-1}(s^n,a^n)$$

被称为“时间差分”(temporal difference)，因为它获得了从一次迭代到下一次迭代的当前估计$\bar{Q}^{n-1}(s^n,a^n)$与更新的估计$(r(s^n,a^n)+\lambda\max_{a'}\bar{Q}^{n-1}(s',a')-\bar{Q}^{n-1}(s^n,a^n))$的差值。式(2.21)被称为时间差分学习(temporal difference learning)，其通过用于选择状态和动作的固定策略来执行。该算法称为“TD(λ)”(反映算法折扣因子λ的作用)，这种方法被称为“TD学习”。第16章和第17章将其称为近似价值迭代。

为了计算式(2.19)，假设给定一个状态s^n，例如图2.3中迷宫内老鼠的位置。使用某种方法(“策略”)来选择一个动作a^n，这会产生一个回报$r(s^n,a^n)$。接下来，选择一个下游状态，其可能因为处于状态s^n并采取动作a^n而得到。有以下两种方法可以做到这一点。

(1) 无模型学习。假设有一个可以观察的物理系统，例如做诊断的医生或从互联网上选择产品的人。

(2) 基于模型的学习。这里假设从一步转移矩阵$p(s'|s,a)$中对下游状态进行采样。实际上，真正要做的是从$s'=S^M(s^n,a^n,W^{n+1})$模拟转移函数，其中函数$S^M(\cdot)$(使用我们的符号)与最优控制的式(2.14)相同，并且W^{n+1}是一个随机变量，必须从某些(已知)分布中采样。

计算机科学家经常研究观察系统的问题，这意味着他们不使用转移函数的显式模型。

一旦有了模拟的下游状态s'，就可以根据目前的估计$\bar{Q}^{n-1}(s',a')$(称为“Q因子”)找到

最佳动作a'。最后，更新处于状态s^n并采取动作a^n的估计价值。当这种逻辑应用于图2.3中的迷宫时，算法会稳定地学习找到出口概率最高的状态-动作对(pair)，但它确实需要足够频繁地对所有状态和动作进行采样。

有许多Q学习变体反映不同规则，这些规则用于选择状态s^n，选择动作a^n，处理更新后的估计$\hat{q}^n(s^n,a^n)$，以及计算估计$\bar{Q}^n(s,a)$。例如，式(2.19)~式(2.21)是查找表的一种表示方式，但仍有相当多的研究正在进行，其中用深度神经网络近似$\bar{Q}(s,a)$。

你将逐渐了解到，近似价值函数并不是万能的算法。随着RL领域扩展到更广泛的问题，研究人员开始引入不同的算法策略，这些策略将在本书中作为4类策略的样本出现(基于价值函数近似的策略只是其中之一)。如今，“强化学习”更多地应用于使用广泛策略处理序贯决策问题的领域，这正是它成为本书研究主题的原因。

如今有很多人把“强化学习”等同于Q学习，其实不然，Q学习只是一种算法，而非问题。然而，该领域的领导者将强化学习描述为：

(1) 一个由智能体(agent)在某个环境中执行动作并获得回报的问题类。

(2) 一个将其工作确定为“强化学习”的领域。

(3) 领域使用自定义的适用于该问题类的“强化学习”方法开发的一组方法。

综上，该表征由这样一个领域构成：该领域将其工作自定义为由解决“智能体在环境中执行动作并接受回报的”问题类的任何方法组成的“强化学习”。实际上，“强化学习”常被描述为问题类而非方法，因为“强化学习”所涵盖的很多工作不需要Q学习(或用于近似价值函数的任何方法)。强化学习是一个问题还是一种方法，这个问题在本书撰写之时仍然悬而未解。

我们的论点是：该问题类的更一般表征是序贯决策问题，包括任何由智能体在环境中执行动作的问题，但也包括智能体只观察环境的问题(这是RL领域中的一个重要问题类)。此外，我们并非只关注基于VFA的策略(例如Q学习)，还试图将我们的讨论泛化至所有4类策略。我们注意到，RL领域已经在研究分属4类策略的算法，因此我们认为，我们的通用模型不仅描述了RL领域目前正在研究的所有问题，还描述了RL领域可能会生成的所有问题类和方法。

2.1.7　最优停止

随机优化中的经典问题称为最优停止问题。假设有一个随机过程W_t(这可能是资产的价格)，它决定了我们在t时停止的回报$f(W_t)$(停止并出售资产时收到的报价)。设$\omega\in\Omega$是$W_1,\dots,W_T$的一个样本路径，若把讨论局限于有限期限问题，则这可能代表金融期权的到期日。设：

$$X_t(\omega)=\begin{cases}1 & \text{若在}t\text{时停止}\\ 0 & \text{其他情形}\end{cases}$$

设τ是$X_t = 1$时的时间t(假设$t > \tau$时$X_t = 0$)。这种表示法产生了一个问题，因为ω指定了完整的样本路径，这似乎表明可以在t时做出决策之前展望未来。不要大意——当使用历史数据对策略进行回溯测试时，很容易犯这种错误。它实际上是随机规划领域的一个相当标准的近似值，详见第19章(特别是19.9节中的“两阶段随机规划”)。

为了解决这个问题，需要构建函数X_t并使其只依赖于历史$W_1, \ldots, W_t$。在这种情况下，τ称为停止时间(stopping time)。优化问题可以表述为：

$$\max_{\tau} \mathbb{E} X_\tau f(W_\tau) \tag{2.22}$$

其中，τ为“停止时间”。通常，数学家会规定式中的τ(等价于X_t)为“$\mathcal{F}_t$——可测量函数”，这只是另一种表述“τ不由晚于τ的时间点来计算”的说法。

经过测度—理论概率训练的读者非常熟悉这一术语，不过这一术语对于开发随机优化的模型和算法却不是必需的。9.13节将会介绍这些概念，并解释为什么不需要使用这些术语。

更确切地说，解决式(2.22)中的停止问题的一个方法是，创建一个函数$X^\pi(S_t)$，使其取决于t时系统的状态。假设需要一个用于出售资产的策略。设持有资产，则$R_t = 1$，否则为0。假设$p_1, p_2, \ldots, p_t$是历史价格走势，如果在t时出售，则得到p_t。通过下式，进一步假设我们创建了一个平滑的过程$\bar{p}_t$：

$$\bar{p}_t = (1 - \alpha)\bar{p}_{t-1} + \alpha p_t$$

在t时，状态变量为$S_t = (R_t, \bar{p}_t, p_t)$。出售策略可能如下：

$$X^\pi(S_t|\theta) = \begin{cases} 1 & \text{若 } \bar{p}_t > \theta^{\max} \text{ 或 } \bar{p}_t < \theta^{\min} \\ 0 & \text{其他情形} \end{cases}$$

找到最优策略意味着通过求解下式找到最优$\theta = (\theta^{\min}, \theta^{\max})$：

$$\max_{\theta} \mathbb{E} \sum_{t=0}^{T} p_t X^\pi(S_t|\theta) \tag{2.23}$$

那么，停止时间是最早的时间$\tau = t$，其中$X^\pi(S_t|\theta) = 1$。

最优停止问题十分常见。部分示例如下。

(1) 美式期权。美式期权允许在指定日期或之前出售资产。17.6.1节将以示例阐释如何使用近似动态规划的美式期权。该策略可应用于任何停止问题。

(2) 欧式期权。金融资产的欧式期权允许在未来的指定日期出售该资产。

(3) 机器更换。在监控一台(通常是复杂的)机器的状态时，需要制定一项策略，告知何时停止、维修或更换。

(4) 临床试验。经营药物临床试验的医药公司必须知道何时停止试验并宣布成功或失败。更完整的临床试验模型，参见Powell的著作(第14章)，扫描右侧二维码即可查看。

状态变量的简单性使得最优停止看起来似乎是一个容易解决的问题。然而，在实际应用中，几乎总是需要考虑额外的信息。例如，资产出售问题可能取决于一篮子指数或证券，其大大扩展了状态变量维度；机器更换问题可能涉及多个测量值，做决策时需要综合考虑这些测量值；临床实验结果则总是取决于每个患者特有的多项因素。

2.1.8　随机规划

假设我们是在线零售商，必须将库存分配给不同的配送中心，并满足存放库存的配送中心的需求。调用初始决策x_0(这是“当下”的决策)来分配库存。然后可以看到对产品的需求D_1和零售商将收讫的付款p_1。

设$W_1 = (D_1, p_1)$为这个随机信息，而ω为W_1的样本实现，因此$W_1(\omega) = (D_1(\omega), p_1(\omega))$是需求和价格的一种可能实现。我们在看到这些信息后做出决策x_1，且对于需求的每个可能实现ω都有一个出货决策$x_1(\omega)$。随机规划领域通常将每个结果ω作为一个场景。

假设此时$\Omega = (\omega_1, \omega_2, \ldots, \omega_K)$是需求$D_1(\omega)$和价格$p_1(\omega)$的一组(不太大的)可能结果(“场景”)，那么第二阶段的决策$x_1(\omega)$将受第一阶段x_0做出的初始库存决策的约束。这两个约束写作：

$$\begin{aligned} A_1 x_1(\omega) &\leq x_0, \\ B_1 x_1(\omega) &\leq D_1(\omega) \end{aligned}$$

设$\mathcal{X}_1(\omega)$是$x_1(\omega)$的可行域，由以上约束定义，则这两个阶段的问题可写作：

$$\max_{x_0} \left(-c_0 x_0 + \sum_{\omega \in \Omega} p(\omega) \max_{x_1(\omega) \in \mathcal{X}_1(\omega)} (p_1(\omega) - c_1) x_1(\omega) \right) \tag{2.24}$$

在随机规划术语中，第二阶段的决策变量$x_1(\omega)$被称为“追索权变量”(recourse variable)，因为它们代表了当新信息可用时可能会做出的反应(这就是“追索权”的定义)。两阶段随机规划基本上是确定性优化问题，但它们可以是非常大的确定性优化问题(尽管具有特殊结构)。

例如，假设允许第一阶段决策x_0“查看”第二阶段的信息，这种情况下，我们将其写作$x_0(\omega)$，并得到一系列较小的问题，每个ω对应一个问题。然而，现在我们通过展望未来允许x_0作弊。可通过引入非预期约束(nonanticipativity constraint)来解决这一点，如下所示：

$$x^0(\omega) - x^0 = 0 \tag{2.25}$$

现在，有了一系列第一阶段变量$x_0(\omega)$(每个ω对应一个$x_0(\omega)$)，还有单个变量x_0，我们试图强制每个$x_0(\omega)$保持一致(在这一点上，可以称x_0表示“非预期”)。算法专家可以放宽式(2.25)的非预期约束，然后解决一系列较小的问题(可能是并行的)，然后引入链路机制，从而使整个过程收敛到满足非预期约束的解。

将式(2.24)中的优化问题(以及时间段0和1的相关约束)称为随机优化问题。在实践中，

这些应用往往诞生于序贯决策问题的背景之下，我们将在其中寻找“t时考虑了不确定的未来”(称为$t+1$，不过可以是多个时间段$t+1,\dots,t+H$)的最优决策x_t，得出以下策略：

$$X_t^\pi(S_t) = \arg\max_{x_t\in\mathcal{X}_t}\left(-c_t x_t + \sum_{\omega\in\Omega} p_{t+1}(\omega)\max_{x_{t+1}(\omega)\in\mathcal{X}_{t+1}(\omega)}\left((p_{t+1}(\omega)-c_{t+1})x_{t+1}(\omega)\right)\right) \tag{2.26}$$

式(2.24)和式(2.26)中的优化问题相同，但求解式(2.26)的目标只是找到一个决策x_t来执行，之后将继续前行到时间$t+1$、更新不确定的未来$t+2$，然后重复该过程。每个场景ω的决策$x_{t+1}(\omega)$从未真正实施；对它们进行规划只是为了帮助改进现在要实施的决策x_t。这是一种解决优化问题的策略，通常不会明确建模。2.2节将说明如何对目标函数进行建模。

2.1.9 多臂老虎机问题

经典的信息获取问题被称为多臂老虎机问题(multiarmed bandit problem)，这是2.1.1节中介绍的累积回报问题的一个趣称。这个问题自20世纪50年代首次提出以来就受到了极大的关注且该词条每年都被数千篇论文提及！

老虎机故事进展如下。假设赌徒需要选择一台老虎机$x\in\mathcal{X}=\{1,2,...,M\}$。而每台机器的奖金都不同，但赌徒不知道获胜概率。获取信息的唯一方法是先试试看。若要用公式表述这个问题，可先假设：

x^n = 完成第n次试验后所选择的下一台机器，

W_x^n = 第 n 次试验中玩老虎机 $(x=x^{n-1})$ 所赢的钱

在完成第$n-1$次试验后，选择第n次试验要玩哪台机器。设S^n是玩了n次后的信念状态，而：

μ_x = 给出机器x实际预期奖金的随机变量，

$\bar{\mu}_x^n$ = n次试验之后对μ_x预期值的估计，

$\sigma_x^{2,n}$ = n 次试验之后对 μ_x 信念的方差

现在假设对μ的信念呈正态分布(n次试验之后)，平均值为$\bar{\mu}_x^n$、方差为$\sigma_x^{2,n}$。可将信念状态写作：

$$S^n=(\bar{\mu}_x^n,\sigma_x^{2,n})_{x\in\mathcal{X}}$$

我们的挑战是找到策略$X^\pi(S^n)$，决定在第$n+1$次试验时玩哪台机器x^n。必须找到一个策略，更好地了解真正的平均值μ_x，这意味着有时不得不玩一台回报$\bar{\mu}_x^n$有可能并不是最高的机器x^n，但要承认的是这一估计可能不准确。不过，最后玩的机器的平均回报μ_x实际上可能低于最优水平，这意味着可能获得较低的奖金。问题是要找到一个能使奖金随时间最大化的策略。

表示这个问题的一种方法是在无限期内最大化预期的折扣奖金：

$$\max_{\pi} \mathbb{E} \sum_{n=0}^{\infty} \gamma^n W_{x^n}^{n+1}$$

其中，$x^n = X^{\pi}(S^n)$，$\gamma < 1$是折扣因子。当然，也可将其视为一个有限时域问题(有折扣或无折扣)。

一个效果较好的策略示例称为区间估计策略，见下式：

$$X^{IE,n}(S^n|\theta^{IE}) = \arg\max_{x \in \mathcal{X}} \left(\bar{\mu}_x^n + \theta^{IE} \bar{\sigma}_x^{2,n} \right)$$

其中，$\bar{\sigma}_x^{2,n}$是对$\bar{\mu}_x^n$的方差的估计，见下式：

$$\bar{\sigma}_x^{2,n} = \frac{\sigma_x^{2,n}}{N_x^n}$$

其中，N_x^n是前n次实验中测试备选方案x的次数。策略的参数为θ^{IE}，这决定了在估计$\bar{\mu}_x^n$时对不确定性施加多大的权重。如果$\theta^{IE} = 0$，即采用纯利用策略，仅简单地选择看起来最好的备选方案。随着θ^{IE}增大，则需要更加重视估算中的不确定性。如第7章所述，有效的学习策略必须平衡探索(尝试不确定的备选方案)和利用(做看起来最好的事情)。

多臂老虎机问题是在线学习问题的一个例子(也就是说，在该示例中必须边实践边学习)，我们希望最大化累积回报。这些问题的示例如下。

■ 示例2.1

假设有一个刚搬到陌生城市居住的人当下必须找到一条最优的通勤路径。设T_p是一个随机变量，用来给出他从预定义的一组路径$\mathcal{P}$中选择路径p时将经历的时间。他获得行程时间观察值的唯一方法是沿着路径亲自走一趟。当然，他希望选择平均时间最短的路径，但可能有必要尝试较长的路径，因为他可能估计能力不佳。

■ 示例2.2

一位棒球经理想要判断四名球员中的哪一位是最优指定击球手。估计他们命中率的唯一方法是将他们当作指定击球手，按击球的多少顺序排列。

■ 示例2.3

医生正在为患者选择最优的降压药。每个患者对各种药物的反应都不同，因此有必要在一段时间内尝试一种特定药物，若医生觉得其他药物可以获得更好的治疗结果，则进行切换。

多臂老虎机问题由来已久，是应用概率和统计学(可追溯到20世纪50年代)、计算机科学(始于20世纪80年代中期)以及工程和地球科学(始于20世纪90年代)领域的一个利基问题。老虎机领域已经扩展到更广泛的问题(例如，x可以是连续的且/或为向量)，以及越来越多的策略。第7章将进一步讨论这个重要的问题类，届时我们将指出所谓的“多臂老虎机问

题”实际上只是无导数的随机优化问题，可以用4类策略中的任何一类来解决。老虎机问题与早期的无导数随机搜索研究的不同之处在于，随机搜索研究没有明确认识到主动学习的价值：评估x处的函数只是为了更好地学习近似值，以便以后做出更好的决策。

我们注意到，无导数随机搜索通常使用“最终回报”目标函数(见2.1.1节)，而多臂老虎机研究一直以累积回报目标为中心，但这并非普遍正确。有一种多臂老虎机问题称为“最优臂老虎机问题”，它使用最终回报目标。

2.1.10 模拟优化

“模拟优化”领域最初起源于模拟领域，该领域开发了用于模拟制造过程等复杂系统的蒙特卡洛模拟模型。20世纪60年代早期，常用于搜索一系列设计的模拟模型因其搜索的慢速性而成功地激发了人们想高效执行这些搜索的兴趣。

使用噪声评估在有限的备选方案集中进行搜索是一个排序、选择的例子(无导数随机搜索的一种形式)，但这些应用培养的却是模拟领域的研究人员。该领域最早的创新方法之一是一种称为最优算力预算分配(optimal computing budget allocation，OCBA)的算法。

OCBA算法的一般思想是通过获取每个备选方案$x \in \mathcal{X}$的初始样本$N_x^0 = n_0$来实现的，这意味着要基于预算B开展$B^0 = Mn_0$个实验。然后，该算法使用规则来确定如何在不同的备选方案之间分配其算力预算。7.10.2节将详细总结典型的OCBA算法。

多年来，OCBA与“模拟优化”紧密相连，但模拟优化领域仍在持续发展，解决了更多的问题，并创造了很多新的方法来应对新的挑战。不可避免的是，也出现了一些与其他领域的交集。然而，与其他领域类似，“模拟优化”领域的活动范围不断扩大，涵盖随机搜索的其他结果(无导数和基于导数)，以及用于序贯决策问题的工具，如近似动态规划和强化学习。如今，模拟优化领域将所有基于蒙特卡洛采样的搜索方法都归类为“模拟优化”的一种形式。

2.1.11 主动学习

给定数据集(x^n, y^n)，$n = 1, \ldots, N$，经典(批量)机器学习解决了模型$f(x|\theta)$的拟合问题，使误差(或损失)函数$L(x, y)$最小化。在线学习解决了当数据流到达时拟合模型的设置。给出基于前n个数据点的估计值$\bar{\theta}^n$，给定(x^{n+1}, y^{n+1})，查找$\bar{\theta}^{n+1}$。假设无法控制输入x^n。

当部分或完全控制输入x^n时，就会产生主动学习。它可能是我们掌控的价格、规模或专注度；也可能是在为患者选择治疗方案时部分可控的因素，但我们无法控制患者的属性。

主动学习有很多种方法，其中一种流行的方法是在不确定性最大之处做选择。例如，假设存在二元结果(客户是否以价格x购买产品)。设x是客户的属性，$\bar{p}(x)$是该客户购买产

品的概率。可以从客户的登录凭据中了解客户的属性。响应的方差由$\bar{p}^n(x)(1-\bar{p}^n(x))$给出。为了最小化方差，我们希望向具有属性$x$的客户提供报价，其中，属性$x$具有的不确定性最大，由方差$\bar{p}^n(x)(1-\bar{p}^n(x))$给出。这意味着我们将选择求解下式的$x$：

$$\max_x \bar{p}^n(x)(1-\bar{p}^n(x))$$

这是一个非常简单的主动学习示例。

老虎机问题与主动学习之间的关系非常密切。截至本书撰写之时，“主动学习”一词已经越来越多地取代了杜撰的“多臂老虎机问题”。

2.1.12　机会约束规划

对于有些问题，在做决策时必须满足一个依赖于不确定信息的约束。例如，可能希望以“在80%的情况下都能满足需求”为目标分配库存。或者，可能希望安排一个准时率高达90%的航班。可以用下面这个一般形式表示这些问题：

$$\min_x f(x) \tag{2.27}$$

上式服从概率约束(通常称为机会约束)：

$$\mathbb{P}[C(x,W)\geq 0] \quad \leq \quad \alpha \tag{2.28}$$

其中，$0\leq\alpha\leq 1$。约束式(2.28)通常等效地写作：

$$\mathbb{P}[C(x,W)\leq 0] \quad \geq \quad 1-\alpha \tag{2.29}$$

此处，$C(x,W)$是违背约束的数量(如果为正)。在我们的例子中，$C(x,W)$可能是需求减去库存，如果是正数，则未满足需求；如果是负数，则满足需求。或者，它可以是航班实际的到达时间减去预计到达时间，其中正值意味着航班晚点。

机会约束规划(chance-constrained programming)是一种处理涉及不确定性的特定约束类型的方法，通常在以下静态问题中使用：做决策、查看信息、停止。机会约束规划将这些问题转化为确定性的非线性规划，挑战是要在搜索算法中计算概率约束。

2.1.13　模型预测控制

很多情况都需要考虑未来会发生什么，以便现在做出决策。我们最熟悉的一个例子便是导航系统，该系统使用网络每条链路上的估计行程时间来规划到达目的地的路径。随着人类社会不断进步，这些时间可能会改变，路径也会更新。

在最优控制相关文献中，之所以把通过(以某种方式)优化未来为当下做决策的行为称为模型预测控制(model predictive control)，是因为其使用了未来的(通常是近似的)模型为当下做决策。MPC策略的一个例子如下：

$$U^{\pi}(x_t) = \arg\min_{u_t}\left(L(x_t,u_t)+min_{u_{t+1},\ldots,u_{t+H}}\sum_{t'=t}^{t+H}L(x_{t'},u_{t'})\right)$$
$$= \arg\min_{u_t,\ldots,u_{t+H}}\sum_{t'=t}^{t+H}L(x_{t'},u_{t'}) \tag{2.30}$$

式(2.30)中的优化问题需要一个时域为$t,\ldots,t+H$的模型，这意味着需要能够使用$x_{t+1}=f(x_t,u_t)$对损失以及系统动态进行建模。对此，一个更精确的名称可能是“基于模型的预测控制”，但“模型预测控制”(通常被称为MPC)是在控制界发展起来的术语。

模型预测控制是一个广泛使用的概念，通常以“滚动时域法”(rolling horizon procedure)或“后退时域法”(receding horizon procedure)等名称命名。模型预测控制通常使用未来的确定性模型编写，主要是因为大多数控制问题都是确定性的。然而，该术语其实是指用于当下做决策的任何未来模型(甚至是近似模型)。2.1.8节中的两阶段随机规划模型是一种使用未来随机模型的模型预测控制形式。甚至可以解出一个完整的动态规划，这通常是在求解未来的近似随机模型时完成的。所有这些都是“模型预测控制”的形式。本书将这种方法归类为“直接前瞻近似”策略，详见第19章。

2.1.14 鲁棒优化

“鲁棒优化”(robust optimization)一词已应用于经典的随机优化问题(特别是随机规划)，但在20世纪90年代中期，它与需要我们做决策的问题相关联，例如设备或结构的设计，其在不可控参数的最坏设置下发挥作用。可能出现鲁棒优化的示例如下。

■ 示例2.4

结构工程师要设计一座成本最小化的高楼(这可能涉及材料最小化)，并使它能够承受风速和风向等最恶劣的风暴条件。

■ 示例2.5

一位为大型客机设计机翼的工程师希望将机翼的重量降到最低，但在最坏的情况下，机翼仍必须承受压力。

鲁棒优化领域中使用的经典符号是u，u为不确定的参数。本书使用w，并假设w属于不确定性集合$\mathcal{W}$。集合$\mathcal{W}$旨在以某种置信度获得随机结果，可以用θ参数化置信度，因此不确定性集可写作$\mathcal{W}(\theta)$。

鲁棒优化问题表示为：

$$\min_{x\in\mathcal{X}}\max_{w\in\mathcal{W}(\theta)}F(x,w) \tag{2.31}$$

创建不确定性集$\mathcal{W}(\theta)$可能是一个不小的挑战。例如，如果w是含有元素w_i的向量，表示$\mathcal{W}(\theta)$的一种方法是使用盒子：

$$\mathcal{W}(\theta)=\{w|\theta_i^{lower}\le w_i\le\theta_i^{upper},\ \forall i\}$$

其中，$\theta=(\theta^{lower},\theta^{upper})$是可调参数，用于控制不确定性集的创建。

问题是，$\mathcal{W}(\theta)$的最差结果很可能是盒子的一个角落，所有元素w_i处于其上限或下限。这在实践中可能极为罕见。更现实的不确定性集会捕捉向量w发生的可能性。在鲁棒优化中，有大量的研究集中于不确定性集$\mathcal{W}(\theta)$的创建。

我们注意到，之前在式(2.24)中展示了一个两阶段随机规划问题，然后指出这确实是一个前瞻策略(见式(2.26))，类似地，式(2.31)给出的鲁棒优化问题可以写成鲁棒优化策略，如下：

$$X^{RO}(S_t)=\arg\min_{x_t\in\mathcal{X}_t}\max_{w_{t+1}\in\mathcal{W}_{t+1}(\theta)}F(x_t,w_{t+1}) \tag{2.32}$$

关于鲁棒优化的许多论文正是这样做的：用公式表示时间t处的鲁棒优化问题，然后用它做决策x_t，之后，继续前向观察新信息W_{t+1}，接着重复该过程。这意味着其鲁棒优化问题实际上是一种前瞻策略。

2.2　序贯决策问题的通用建模框架

现在，已经讲解了处理不确定性下的序贯决策的主要领域，有必要回顾所有序贯决策问题的要素。第9章将更深入地讨论这个主题，此处仅简单介绍，以便将我们的框架与上面回顾的框架进行比较。

我们的论述侧重于不确定性下的序贯决策问题，这意味着每次决策后都会有新的信息，但总是可以忽略新的信息来创建一个与2.1.4节中的确定性控制问题类似的问题。我们将假设问题会随着时间的推移而演变，但在许多情况下，倾向于使用计数器(第n次实验，第n个客户)。

2.2.1　序贯决策问题的通用模型

序贯决策问题包括以下要素。

(1) 状态变量——S_t。该变量捕获了从t时起对系统进行建模所需的所有信息，这意味着计算成本/贡献函数、决策约束以及对这些信息随时间推移的转变进行建模所需的任何其他变量。状态S_t可能包括物理资源R_t(如库存)，其他确定性信息I_t(产品价格、天气)和信念状态B_t，B_t会捕捉描述不能直接(和完美)观察的参数或量的概率分布的信息。重要的是认识到，无论状态变量描述物理资源、系统属性还是概率分布参数，状态变量始终是一种

信息。

(2) 决策变量——x_t。决策(可称为动作a_t或控制u_t)代表如何控制过程。决策由被称为策略的决策函数决定，在控制理论中也称为控制律。如果决策是x_t，就把策略表示为$X^\pi(S_t)$。同样，如果希望使用a_t或u_t作为决策变量，则使用$A^\pi(S_t)$或$U^\pi(S_t)$作为策略。如果$\mathcal{X}_t$是可行域(取决于S_t的信息)，则假设$X^\pi(S_t) \in \mathcal{X}_t$。

(3) 外生信息——W_{t+1}。这是在$t+1$时首次从外源知道的信息(例如，产品需求、风速、诊治结果、实验结果)。W_{t+1}可以是价格(针对所有不同库存)或产品需求的高维向量。

(4) 转移函数。给出t时做出的决策以及在t时和$t+1$时之间到达的新信息后，该函数会决定系统如何从状态S_t演变到状态S_{t+1}。我们将转移函数(也称为系统模型或状态转移模型)表示为：

$$S_{t+1} = S^M(S_t, x_t, W_{t+1})$$

注意，做出决策x_t时，W_{t+1}是一个随机变量。在整个过程中，我们假设由t(或n)索引的任何变量在t时(或n次观察之后)均已知。

(5) 目标函数。该函数指定最小化的成本、最大化的贡献/回报等表现指标。设$C(S_t, x_t)$是给出决策x_t以及S_t中的信息后的最大化贡献——S_t中的信息可能包含成本、价格和约束信息。目标函数的基本形式如下：

$$F^\pi(S_0) = \mathbb{E}_{S_0}\mathbb{E}_{W_1,\dots,W_T|S_0}\left\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t))\right\} \tag{2.33}$$

我们的目标是找到策略以求解：

$$\max_\pi F^\pi(S_0) \tag{2.34}$$

第7章和第9章将阐述一些其他形式的目标。

如果使用计数器，就可以用S^n表示状态，x^n表示决策，W^{n+1}表示外生信息。有些问题则需要同时按时间(如一周内的小时)和计数器(如第n周)索引化，因此可以使用S_t^n。

下面用一个资产收购问题来说明这个框架。

(1) 叙述。资产收购问题涉及维持一些资源的库存(共同基金中的现金、飞机的备用发动机、疫苗等)，以满足随时间推移的随机需求。假设购买成本和销售价格也会随时间而变化。

(2) 状态变量。状态变量是做决策和计算函数所需的信息，这些函数决定了系统未来的发展。在资产收购问题中，需要3条信息。第一是R_t，表示在做出任何决策(包括满足多少需求)之前手头的资源。第二是需求本身，表示为D_t。第三是价格p_t。我们将状态变量写作$S_t = (R_t, D_t, p_t)$。

(3) 决策变量。有两个决策要做。第一个决策表示为x_t^D，表示在t时间段内应该使用多少可用资产来满足需求D_t，这意味着$x_t^D \le R_t$。第二个决策表示为x_t^O，表示在t时应该收购

多少新资产，该资产可用于满足$t+1$时间段内的需求。

(4) 外生信息——外生信息过程由3类信息组成。第一类信息是出现在t和$t+1$之间的新需求，表示为$\hat{D}_{t+1}$。第二类信息是t至$t+1$之间出售资产的价格变化，表示为$\hat{p}_{t+1}$。最后，假设可用资源可能会发生外源性变化。此类信息可能涉及献血或现金存款(产生积极变化)，或设备故障和现金提取(产生消极变化)。用$\hat{R}_{t+1}$表示这些变化。设W_{t+1}表示在t至$t+1$时之间(即做出决策x_t后)初次了解的所有新信息，在我们的问题中可以写作$W_{t+1}=(\hat{R}_{t+1},\hat{D}_{t+1},\hat{p}_{t+1})$。

除了指定外生信息的类型，还必须为随机模型指定特定结果的可能性。其形式可能是$\hat{R}_{t+1}$、$\hat{D}_{t+1}$及$\hat{p}_{t+1}$的假设概率分布，或者可能依赖于样本实现的外源(股票的实际价格或路径上的实际行程时间)。

(5) 转移函数。用下式描述状态变量S_t的演变：

$$S_{t+1}=S^M(S_t,x_t,W_{t+1})$$

上面各式中：

$$\begin{aligned}R_{t+1} &= R_t-x_t^D+x_t^O+\hat{R}_{t+1},\\ D_{t+1} &= D_t-x_t^D+\hat{D}_{t+1},\\ p_{t+1} &= p_t+\hat{p}_{t+1}\end{aligned}$$

该模型假设未满足的需求会一直保持到下一个时间段。

(6) 目标函数。计算贡献$C_t(S_t,x_t)$，这可能取决于目前的状态和t时采取的动作x_t。资产收购问题(其中状态变量为R_t)的贡献函数为：

$$C_t(S_t,x_t)=p_tx_t^D-c_tx_t^O$$

在该特定模型中，$C_t(S_t,x_t)$是状态和动作的确定性函数。在其他应用中，来自动作x_t的贡献取决于$t+1$时发生的事情。

目标函数由下式给出：

$$\max_{\pi\in\Pi}\mathbb{E}\left\{\sum_{t=0}^{T}C_t(S_t,X^\pi(S_t))|S_0\right\}$$

设计策略将占据本书大部分内容。对于这样的库存问题，可以使用简单的规则或更复杂的前瞻策略，可以通过点预测来展望未来，或者捕捉未来的不确定性。

第9章将专门补充这个基本建模框架的细节。对实际问题建模时，我们鼓励读者按顺序描述上述5个要素。

2.2.2　紧凑型建模

编写序贯决策问题时，如果希望其形式更紧凑，即采用近似于经典确定性数学规划的

形式，则建议将其写作：

$$\max_{\pi} \mathbb{E}_{S_0} \mathbb{E}_{W_1,\ldots,W_T|S_0} \left\{ \sum_{t=0}^{T} C(S_t, X^{\pi}(S_t)) \right\} \tag{2.35}$$

其中，假设策略满足约束：

$$x_t = X^{\pi}(S_t) \in \mathcal{X}_t \tag{2.36}$$

转移函数由下式给出：

$$S_{t+1} = S^M(S_t, X^{\pi}(S_t), W_{t+1}) \tag{2.37}$$

并且会给出一个外生信息过程：

$$(S_0, W_1, W_2, \ldots, W_T) \tag{2.38}$$

当然，这也留下了这样的问题：如何描述外生信息过程的采样方式及设计策略的方式。然而，我们认为不必解释策略，就像不必解释确定性数学规划中的决策x一样。

2.2.3 MDP/RL与最优控制建模框架

停下来思考一个自然而然的问题：在2.1节列出的所有领域中，是否有任何领域符合我们的通用框架？有一个较为接近：最优控制(见2.1.4节)。

在描述最优控制建模框架的优点之前，先介绍截至本书撰写之时所有领域中最流行的强化学习(RL)领域采用的建模框架。从20世纪80年代开始，RL领域采用了长期用于马尔可夫决策过程的建模框架(框架简介参见2.1.3节)。这个框架在数学上实属巧妙，但在建模实际问题时却极难处理。例如，定义“状态空间”$\mathcal{S}$或“动作空间”$\mathcal{A}$后，我们仍对问题一无所知。此外，一步转移矩阵$P(s'|s,a)$几乎不可计算。最后，虽然可以指定单周期回报函数，但真正的问题是对回报进行求和并优化策略。

接下来，将这种形式与最优控制中使用的形式进行对比。在此领域中，我们会指定状态变量和决策/控制变量。最优控制领域引入的转移函数的强大构造看似明显，却往往会被其他领域所忽视。最优控制文献主要关注确定性问题，但也有随机控制问题，通常使用式(2.15)的加性噪声。

最优控制领域没有使用我们的标准格式来优化策略。然而，这个领域积极制定了不同类别的策略。我们观察到，最优控制相关文献首次引入了“线性控制律”(因为它们对于线性二次调节问题是最优的)。该领域率先为价值函数近似赋予多种名称，包括启发式动态规划、神经动态规划和近似/自适应动态规划。最后，引入了(确定性)前瞻策略(称为“模型预测控制”)。这涵盖了4类策略中的3类(PFA、VFA和DLA)。我们猜想有人已将参数化优化模型的思想用于策略(我们称之为CFA)，但由于该策略尚未被公认为正式的方法，因此很难知道它是否已被使用以及何时首次使用。

2.1节中的所有领域都有将建模框架与解决方案联系起来的习惯。最优控制以及动态规划假设起点是贝尔曼方程(在控制界称为哈密顿-雅可比方程)。这是我们对上述所有领域的主要出发点。在我们的通用建模框架中，5个要素都没有指示如何设计策略。相反，我们以一个目标函数(式(2.33)~式(2.34))结束，并声明我们的目标是找到一个最优策略。后续章节将针对1.4.1节中首次介绍的4类策略进行搜索，并将在本书中反复提及。

2.3　应用

接下来通过一系列应用说明我们的建模框架。这些问题阐释了实际应用中可能出现的一些建模问题。我们通常从一个较简单的问题开始，然后展示如何添加细节。在引入复杂性时，请注意状态变量维度的增长。

2.3.1　报童问题

运筹学中一个流行的问题被称为报童问题，它被描述为决定发行多少报纸以满足未知需求。报童问题出现在许多设置中，我们必须选择一个固定参数，然后在随机设置中进行评估。它通常作为一个子问题出现在一系列资源分配问题(管理血液库存、紧急情况成本、分配车队、聘用人员)中。其他情况下也会出现这一问题，例如合同报价(报价过高意味着可能会错失合同)，或者留出额外的旅行时间。

报童问题通常被描述为静态的最终回报公式，但我们会开放性地探讨最终回报和累积回报公式。

1. 基本报童——最终回报

基本报童建模为：

$$F(x,W)=p\min\{x,W\}-cx \tag{2.39}$$

其中，x是在观察随机“需求”W之前必须订购的“报纸”数量。以价格p出售报纸(x和W较小)，但必须以单位成本c购买全部报纸。目标是解决以下问题：

$$\max_x \mathbb{E}_W F(x,W) \tag{2.40}$$

大多数情况下，该报童问题在可以观察到W的环境中发生，但其分布未知(通常被称为“数据驱动”)。这种情况下，假设必须确定在第n天结束时要订购的数量x^n，之后观察需求W^{n+1}，可以得到如下利润(在第$n+1$天结束时)：

$$\hat{F}^{n+1}=F(x^n,W^{n+1})=p\min\{x^n,W^{n+1}\}-cx^n$$

在每次迭代之后，可以假设观察到W^{n+1}，尽管经常只能观察到$\min(x^n,W^{n+1})$(这被称为截尾观察，censored observation)，或可能只能观察实现的利润：

$$\hat{F}^{n+1} = p\min\{x^n, W^{n+1}\} - cx^n$$

可以制定策略，试图了解W的分布，然后尽力以最优方式解决问题(参见练习4.12)。

另一种方法是尝试直接学习函数$\mathbb{E}_W F(x,W)$。不管怎样，假设S^n是关于未知量的信念状态(关于W，或关于$\mathbb{E}_W F(x,W)$)。S^n可能是点估计，但通常是概率分布。例如，可以设$\mu_x = \mathbb{E}F(x,W)$，假设x是离散的(如报纸的数量)。n次迭代之后，可能得到$\mathbb{E}F(x,W)$的估计$\bar{\mu}_x^n$和标准差$\bar{\sigma}_x^n$，然后假设$\mu_x \sim N(\bar{\mu}_x^n, \bar{\sigma}_x^{n,2})$。这种情况下，可得出$S^n = (\bar{\mu}^n, \bar{\sigma}^n)$，其中$\bar{\mu}^n$和$\bar{\sigma}^n$都是$x$所有值的向量。

给定(信念)状态S^n，然后定义一个用$X^\pi(S^n)$表示的策略(也可以称之为规则，或者它可能是一种算法)，其中$x^n = X^\pi(S^n)$是我们将在下一次观察W^{n+1}或$\hat{F}^{n+1}$的试验中使用的决策。虽然希望这一策略能一直适用至$n \to \infty$，但实际上，这会受限于之后给出解决方案$x^{\pi,N}$的N次试验。这个解决方案取决于初始状态S^0、求解$x^{\pi,N}$时出现的观察结果$W^1, \ldots, W^N$，以及随后观察到的用于评估$x^{\pi,N}$的$\widehat{W}$。需要找到求解下式的策略：

$$\max_\pi \mathbb{E}_{S^0}\mathbb{E}_{W^1,\ldots,W^N|S^0}\mathbb{E}_{\widehat{W}|S^0} F(x^{\pi,N}, \widehat{W}) \tag{2.41}$$

2. 基本报童——累积回报

对真实的报童问题的更现实的描述是，在积累利润的同时了解需求W(或函数$\mathbb{E}_W F(x,W)$)。这种情况下，要找到一个策略以求解下式：

$$\max_\pi \mathbb{E}_{S_0}\mathbb{E}_{W_1,\ldots,W_T|S_0} \sum_{t=0}^{T-1} F(X^\pi(S_t), W_{t+1}) \tag{2.42}$$

报童问题的累积回报公式捕捉了主动学习过程，尽管该公式是真实报童问题最自然的模型，但它是全新的。

3. 报童背景

假设在某个报童问题中，产品的价格p是动态的，由p_t给出，这是做出决策之前就已明确的。利润由下式计算：

$$F(x, W|S_t) = p_t \min\{x, W\} - cx \tag{2.43}$$

如前所述，假设不知道W的分布，B_t是对W(或关于$\mathbb{E}F(x,W)$)的信念状态。状态$S_t = (p_t, B_t)$，因为必须同时掌握价格p_t以及信念状态B_t，所以可以把问题改写成：

$$\max_x \mathbb{E}_W F(x, W|S_t)$$

现在，必须找到最优订购量函数$x^*(S_t)$，而不是找到最优订购量x^*。虽然x^*是确定性值，但$x^*(S)$是代表决策x^*“上下文”的状态S的函数。

如上所示，“上下文”(学习领域中的一个常用术语)实际上只是一个状态变量，而$x^*(S)$是一种策略。找到一个最优的策略总是很难，但若要找到一个切实可行的策略，只需要仔细研究4类策略中的每一类，就能得到一个有望成功的策略。

4. 多维报童问题

报童问题可能是多维的。一个版本是加性报童问题，其中有K个产品服务K种需求，但使用的生产流程限制总交付量。这将被表述为：

$$F(x_1,\dots,x_K)=E_{W_1,\dots,W_K}\sum_{k=1}^{K}p_k\min(x_k,W_k)-c_kx_k \tag{2.44}$$

其中：

$$\sum_{k=1}^{K}x_k\le U \tag{2.45}$$

当有多种产品(不同类型/颜色的汽车)试图满足同一需求W时，就会出现第二个版本，见下式：

$$F(x_1,\dots,x_K)=\mathbb{E}_W\left\{\sum_{k=1}^{K}p_k\min\left[x_k,\left(W-\sum_{\ell=1}^{k-1}x_\ell\right)^+\right]-\sum_{k=1}^{K}c_kx_k\right\} \tag{2.46}$$

其中$(Z)^+=\max(0,Z)$。

2.3.2　库存/储存问题

库存(或储存)问题代表的应用类别非常广泛，涵盖了购买/获取(或出售)资源以满足需求的任何问题，其中过剩的库存可以保留到下一个时间段。基本库存问题(具有离散量)似乎是说明某个紧凑状态空间作用的第一个问题，它可以解决当试图将这些问题表述为决策树并求解时出现的指数爆炸。然而，在处理实际应用时，这些基本问题很快会变得复杂。

1. 无滞后库存

最简单的问题就是在t时订购新产品x_t，且立即送达。先定义符号：

$R_t=t$时间段末剩余库存量，

$x_t=t$时间段末订购量，将在t时间段开始时提供，

$\hat{D}_{t+1}=t$至$t+1$之间的产品需求，

$c_t=$在t时订购产品的单位成本，

$p_t=$在$(t,t+1)$期间出售一个单位产品所收取的费用

基本库存流程如下：

$$R_{t+1}=\max\{0,R_t+x_t-\hat{D}_{t+1}\}$$

在每个时间段末将总贡献相加。设y_t是时间段$(t-1,t)$内的销售额。销售受需求$\hat{D}_t$以及现有产品量$R_{t-1}+x_{t-1}$的限制，但可以自行选择卖出多少，y_t可能比这两者都小。由此可得出：

$$\begin{aligned} y_t &\leq R_{t-1} + x_{t-1}, \\ y_t &\leq \hat{D}_t \end{aligned}$$

假设在了解了前一时间段的需求D_t之后要确定t时的y_t，可以通过下式给出t时的收入和成本：

$$C_t(x_t, y_t) = p_t y_t - c_t x_t$$

如果这是一个确定性问题，可将其表示为：

$$\max_{(x_t, y_t), t=0,\ldots,T} \sum_{t=0}^{T} (p_t y_t - c_t x_t)$$

然而，需求$\hat{D}_{t+1}$在t时通常是随机的，我们希望将这一点表示出来。我们可能想允许价格p_t(甚至成本c_t)在可预测(如季节性)且随机(不确定)模式下随时间变化。在这种情况下，需要定义一个状态变量S_t来捕捉在做出决策x_t和y_t之前，t时的已知信息。状态变量的设计是很微妙的，但现在假设它包括R_t、p_t、c_t，以及在区间$(t, t+1)$出现的需求D_{t+1}。

与报童问题不同，库存问题可能更具挑战性，即使需求D_t的分布是已知的，也是如此。然而，如果需求D_t的分布是未知的，那么可能需要保持关于需求分布的信念状态B_t，或下订单x_t时的预期利润。

此问题的特点使得我们能够创建一系列问题。

(1) 静态数据。如果价格p_t和成本c_t是恒定的(也就是说$p_t = p$、$c_t = c$)，在已知需求分布的情况下，有一个随机优化问题，其中状态是$S_t = R_t$。

(2) 动态数据。假设价格p_t随时间随机演变，其中$p_{t+1} = p_t + \varepsilon_{t+1}$，那么状态变量是$S_t = (R_t, p_t)$。

(3) 依赖于历史的流程。现在假设价格流程演变如下：

$$p_{t+1} = \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \varepsilon_{t+1}$$

然后将状态写作$S_t = \big(R_t, (p_t, p_{t-1}, p_{t-2})\big)$。

(4) 学习过程。现在假设需求分布未知。可以建立一个过程来尝试从需求或销售的观察中学习。设B_t捕捉我们对需求分布的信念，这本身可能是一种概率分布。在这种情况下，状态变量将是$S_t = (R_t, p_t, B_t)$。

设$Y^\pi(S_t)$是用来确定y_t的销售策略，而$X^\pi(S_t)$是用来决定x_t的购买策略，其中π带有决定两个策略的参数。可将目标函数写作：

$$\max_\pi \mathbb{E} \sum_{t=0}^{T} (p_t Y^\pi(S_t) - c_t X^\pi(S_t))$$

库存问题非常多。这是一个很容易创建变体的问题，这些变体可以用1.4节中介绍的4类策略来解决。第11章将更深入地描述这4类策略。11.9节讲述了能源储存中出现的库存问题，对此，4类策略中的每一类都可能发挥最优作用。

2. 带预测的库存计划

许多实际应用中都有一个重要扩展，即数据(需求、价格甚至成本)可能遵循可以近似预测的时变模式。设：

$f_{tt'}^{W}=$ 在t时对某些活动(需求、价格、成本)的预测(我们认为活动会发生在t'时)

预测随着时间的推移而变化。它们可能由外源(预测供应商)提供，或者我们可以使用观察数据自行更新预测。假设它们都由外部供应商提供，则可以用下式描述预测的演变：

$$f_{t+1,t'}^{W}=f_{tt'}^{W}+\hat{f}_{t+1,t'}^{W}$$

其中，$\hat{f}_{t+1,t'}^{W}$是在未来所有时间段t'预测的(随机)变化。

当我们有预测时，向量$f_{t}^{W}=(f_{tt'}^{W})_{t'\geq t}$在技术上会成为状态变量的一部分。当预测可用时，标准方法是将预测视为潜在变量，这意味着不必明确地对预测的演变进行建模，而是将预测视为静态向量。此内容详见第9章，第13章将介绍处理滚动预测的策略。

3. 决策滞后

在很多应用中，t时做出的决策(例如订购新货物)会在t'时才完成(由于运输延误)。在全球物流中，这些滞后可能会持续几个月。对于一家订购新飞机的航空公司来说，这种滞后可能会持续数年。

可以用符号表示滞后：

$x_{tt'}=t'$ 时送达的 t 时订购的货物

$R_{tt'}=$ 将在t'时送达的早在t时以前订购的货物

变量$R_{tt'}$表示如何捕捉先前决策的影响。可以将这些变量汇总到向量$x_{t}=(x_{tt'})_{t'\geq t}$和$R_{t}=(R_{tt'})_{t'\geq t}$中。

滞后问题特别难以建模。假设想在t''月签署购买天然气的合同，这可能需要三年的时间来满足不确定的需求。这个决策必须考虑我们在t'时下订单$x_{t't''}$的可能性，t'时位于现在(t时)和t''时之间。在t时，决策$x_{t't''}$是一个随机变量，不仅取决于t'时的天然气价格，还取决于可能在t和t'之间做出的决策，以及不断变化的预测。

2.3.3　最短路径问题

最短路径问题代表一个特别巧妙而强大的问题类别，因为网络中的节点可以表示任何离散状态，而节点外的链路可以表示离散动作。

1. 确定性最短路径问题

经典的序贯决策问题是最短路径问题。设：

$\mathcal{I}=$ 网络中的一组节点(交点)，

$\mathcal{L}=$ 网络中的一组链路(i,j)，

$c_{ij}=$ 从节点i开车到节点j的成本(通常是时间)，$i,j\in\mathcal{I},(i,j)\in\mathcal{L}$，

$\mathcal{I}_i^+ =$ 节点集合j，其中有一个链路$(i,j) \in \mathcal{L}$，

$\mathcal{I}_j^- =$ 节点集合i，其中有一个链路$(i,j) \in \mathcal{L}$

节点i处的行人需要选择链路(i,j)，其中$j \in \mathcal{I}_i^+$是节点i的下游节点。假设行人需要以最小成本从起始节点j到目的节点r。设：

$v_j =$ 从节点j到节点r所需的最小成本

可以将v_j视为处于状态j的价值。在最优条件下，这些值将满足：

$$v_i = \min_{j \in \mathcal{I}_i^+} \left(c_{ij} + v_j\right)$$

这个基本公式是导航系统中使用的所有最短路径算法的基础，不过这些算法经过了大量设计，以实现人们早已习惯的快速响应。图2.4给出了一个基本的最短路径算法，不过这只是一个真正算法的骨架。

步骤0 设：

$$v_j^0 = \begin{cases} M & j \neq r \\ 0 & j = r \end{cases}$$

其中，M被称为大M，代表一个大数。设$n = 1$

步骤1 对于所有$i \in \mathcal{I}$，求解：

$$v_i^n = \min_{j \in \mathcal{I}_i^+} \left(c_{ij} + v_j^{n-1}\right)$$

步骤2 如果对于任何i，$v_i^n < v_i^{n-1}$都成立，则设$n = n + 1$并返回步骤1，否则停止

图2.4 基本最短路径算法

2. 随机最短路径问题

我们通常对最短路径问题感兴趣，其中遍历链路的成本存在不确定性。在交通示例中，可以很自然地将一个链路的行程时间视为随机的，以反映每个链路上交通状况的可变性。

为正确处理这个新维度，在决定是否遍历链路之前或之后，必须指定是否能看到链路上的随机成本结果。如果实际成本仅在遍历链路后才实现，那么节点i处所做的决策x_i将被写作：

$$x_i = \arg\min_{j \in \mathcal{I}_i^+} \mathbb{E}\left(\hat{c}_{ij} + v_j\right)$$

其中，期望值与随机成本$\hat{c}_{ij}$的分布(假设已知)相关。对于这个问题，状态变量S只是所在的节点。

如果在了解$\hat{c}_{ij}$之后做出决策，那么决策可以写作：

$$x_i = \arg\min_{j \in \mathcal{J}_i^+} \left(\hat{c}_{ij} + v_j\right)$$

在此设置中，状态变量S由$S = (i, (\hat{c}_{ij})_j)$给出，既包括当前节点，也包括来自节点$i$的链路成本。

3. 动态最短路径问题

现在假设，任何在线导航系统都能从网络获取实时信息，并定期更新最短路径，从而解决这个问题。假设导航系统在t时拥有遍历链路$(i, j) \in \mathcal{L}$的成本的估计$\bar{c}_{tij}$，其中$\mathcal{L}$是网络中所有链路的集合。该系统使用这些估计来解决确定性最短路径问题，并给出当下的建议。

假设估计成本的向量$\bar{c}_t$每个时间段更新一次(可能每5分钟更新一次)，因此在$t+1$时就得到了估计向量$\bar{c}_{t+1}$。设N_t是行人所在的节点(或前往的目的地)。状态变量当下可以写作：

$$S_t = (N_t, \bar{c}_t)$$

记住，网络中的每个链路都有一个$\bar{c}_t$元素，状态变量S_t具有维度$|\mathcal{L}| + 1$。第19章将描述如何使用简单的最短路径计算来解决如此复杂的问题。

4. 鲁棒最短路径问题

我们知道成本c_{ij}不确定。导航服务可以使用其观察来构建$\bar{c}_{tij}$的概率分布，以根据我们在t时的已知信息来估计行程时间。现在假设使用θ-百分位(用$\bar{c}_{tij}(\theta)$表示)，而不取平均值。因此，如果设$\theta = 0.90$，将使用第90百分位的行程时间，这将阻止我们使用可能变得非常拥堵的链路。

现在假设当处于状态$S_t = (N_t, \bar{c}_t(\theta))$，并通过使用链路成本$\bar{c}_t(\theta)$解决确定性最短路径问题来选择方向时，$\ell_t^\pi(\theta) \in \mathcal{L}$是推荐的链路。设$\hat{c}_{t,\ell_t^\pi(\theta)}$是行人在$t$时遍历链路$\ell_t^\pi(\theta) = (i, j) \in \mathcal{L}$的实际成本。现在的问题是通过求解下式来优化这类策略：

$$\min_\theta \mathbb{E}\left\{\sum_t \hat{c}_{t,\ell_t^\pi(\theta)} | S_0\right\}$$

其中，S_0捕捉了车辆的起点和成本的初始估计。第19章将进一步讨论该策略。

2.3.4　一些车队管理问题

车队管理问题(例如在网约车车队中出现的问题)代表了一类特殊的资源分配问题。本节首先描述一个“漂泊的货车司机”所面临的问题，然后展示如何将该基本理念扩展到货车车队。

1. 漂泊的货车司机

在漂泊的货车司机问题中，一个货车司机会在A处装载一批货物，从A处驶至B处，然后在B处卸载货物，再继续寻找新的货物(在某些地方可以通过打电话获取可运输货物列

表)。司机必须考虑运输货物所得收入，但他也必须认识到，货物会带他驶向另一个城市。他的问题是如何在A处之外的一组货物中进行选择。

司机在每个时间点都有其当前或未来位置ℓ_t(国家的一个地区)、他的设备类型E_t——他所开的货车的类型(根据货物的需要而变化)、他预计到达ℓ_t的时间(表示为τ_t^{eta})和他离开家的时间τ_t^{home}。我们将这些属性转化为属性向量a_t，如下所示：

$$a_t = (\ell_t, E_t, \tau_t^{eta}, \tau_t^{home})$$

当司机到达货物的目的地时，打电话给货运代理并得到一组货物$\mathcal{L}_t$，他可以从中选择接下来要运送的货物。这意味着他的状态变量(做决策之前的信息)如下：

$$S_t = (a_t, \mathcal{L}_t)$$

司机必须在一系列动作$\mathcal{X}_t = (\mathcal{L}_t, \text{"hold"})$中进行选择，其中包括集合$\mathcal{L}_t$中的货物，或者什么都不做。一旦司机做出决策，$\mathcal{L}_t$就不再相关。他做出决策后的状态称为决策后状态$S_t^x = a_t^x$(做出决策后的即时状态)，该状态会被实时更新以反映货物的目的地以及预计到达该位置的时间。

司机在选择采用哪个动作时会自然而然地权衡动作的贡献(可以写作$C(S_t, x_t)$)和他在决策后的状态a_t^x中的值。可称该策略为$X^\pi(S_t)$，并用下式表示：

$$X^\pi(S_t) = \arg\max_{x \in \mathcal{X}_t} \left(C(S_t, x) + \overline{V}_t^x(a_t^x)\right) \tag{2.47}$$

算法上的挑战是创建估计$\overline{V}_t^x(a_t^x)$，这是一个价值函数近似的例子。如果司机属性向量a_t^x的可能值不是太大，那么可以使用与解决2.3.3节中介绍的随机最短路径问题所用的方法相同的方法来解决这个问题。这个问题中隐藏的假设是节点的数量不太多(即使是一百万个节点，也被认为是可管理的)。当“节点”是多维向量a_t时，则可能很难处理所有可能的值(“维数灾难”的另一个例子)。

2. 从一名司机到一个车队

可以通过以下定义对一个车队进行建模：

R_{ta} = 司机数量(带有 t 时的属性向量a)，

$R_t = (R_{ta})_{a \in \mathcal{A}}$

其中，$a \in \mathcal{A}$位于属性空间中，该属性空间跨越每个a_t元素可能取的所有可能值。

同理，可以通过包含始发地、目的地、预定的装/卸货窗口、所需设备类型及货物是否包含危险材料等信息的属性向量b来描述货物。在美国，典型的做法是将全国分为100个区域，从而提供10 000对始发地和目的地。设：

L_{tb} = 货物数量(带有t时的属性向量b)，

$L_t = (L_{tb})_{b \in \mathcal{B}}$

状态变量由下式给出：

$$S_t = (R_t, L_t)$$

读者可以自行尝试估计这个问题的状态空间大小。第18章将阐释如何使用价值函数近似来解决这个问题。

2.3.5　定价

假设正在尝试为产品定价，并且感觉可以使用以下公式给出的逻辑曲线来模拟产品的需求：

$$D(p|\theta) = \theta_0 \frac{e^{\theta_1 - \theta_2 p}}{1 + e^{\theta_1 - \theta_2 p}}$$

价格为p时所得总收入由下式给出：

$$R(p|\theta) = pD(p|\theta)$$

如果知道θ，那么可以轻松找到最优价格。但现在假设不知道θ。图2.5展示了一系列用来表示价格-收入函数的曲线。

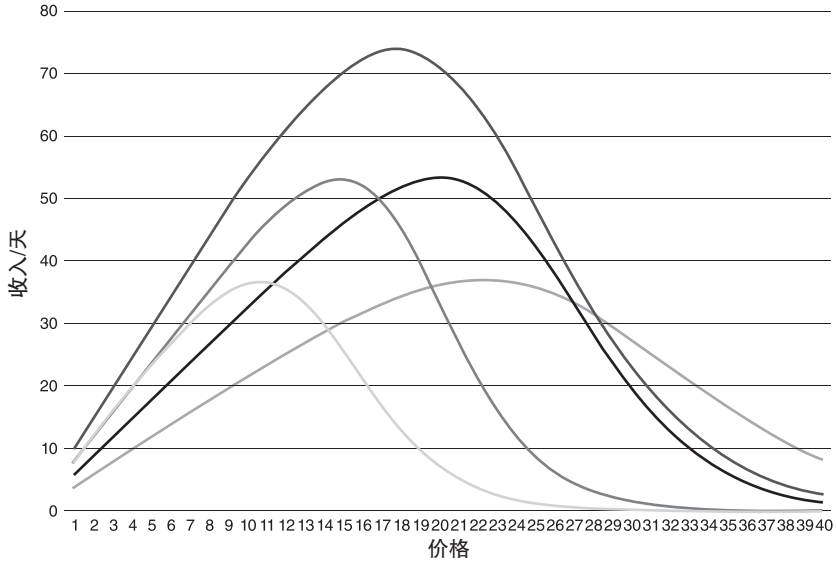

图2.5　一系列可能的收入曲线

可以把这个问题当作了解θ的真实值的一种方式。设$\Theta = (\theta_1, \ldots, \theta_K)$是$\theta$的可能值集合，其中假设$\Theta$的元素之一是真实值。设$P_k^n$是在完成$n$次观察之后$\theta = \theta_k$的概率。那么，该学习系统的状态$S^n = (p_k^n)_{k=1}^K$可捕获关于$\theta$的信念。第7章将进一步讨论这个问题。

2.3.6　医疗决策

医生必须对带着某种抱怨前来就医的患者做出决策。这一过程始于记录病历，其中包括一系列关于患者病史和生活习惯的问题。设h^n是病史，h^n可能由数千种不同的症状组成

(人类的疾病很复杂)。然后，医生可能会要求额外的测试，以得到额外信息，或者可能会开药或要求手术。设d^n捕捉这些决策。可以将患者病史h^n和医疗决策d^n的组合包含在指定的一组解释变量$x^n=(h^n,d^n)$中，同时设θ是维度与x^n相同的参数向量。

现在假设观察到一个结果y^n，为简单起见，将其表示为二元，其中$y^n=1$可以解释为“成功”，$y^n=0$则表示“失败”。我们将假设可以使用逻辑回归模型对随机变量y^n进行建模(所谓随机，即指在观察治疗结果之前)，如下式所示：

$$\mathbb{P}[y^n=1|x^n=(h^n,d^n),\theta]=\frac{e^{\theta^T x^n}}{1+e^{\theta^T x^n}} \tag{2.48}$$

这个问题展示了两种类型的不确定性。首先是患者病史h^n，我们通常没有描述这些属性的概率分布。很难(实际上，不可能)对h^n获得的复杂特征建立一个概率模型，因为历史会呈现复杂的相关性。相比之下，随机变量y^n具有明确定义的数学模型，其特征是未知(和高维)参数向量θ。

可以使用两种不同的方法来处理这些不同类型的不确定性。对于患者属性，可使用通常称为数据驱动的方法。可以访问包含先验属性、决策和结果的大型数据集，表示为$(x^n=(h^n,d^n),y^n)_{n=1}^N$，或者，可以假设只观察患者$h^n$(这是数据驱动的部分)，然后使用取决于状态变量$S^n$的决策函数$D^\pi(S^n)$做出决策$d^n=D^\pi(S^n)$，最后观察可以使用概率模型来描述的结果$y^n$。

2.3.7 科学探索

科学家们在发明新药、新材料或开发新型机翼或火箭发动机时，往往需要进行艰难的实验，寻找产生最优结果的输入和过程。输入可能是催化剂的选用、纳米粒子的形状或分子化合物的选用。制造过程中可能涉及不同的步骤，或者涉及抛光镜片机器的选用。

然后是连续的决策。温度、压力、浓度、比率、位置、直径、长度和时间都是连续参数的示例。在某些设置中，这些参数是自然离散的，不过如果同时调整3个或3个以上的连续参数，则可能出现问题。

可将离散决策表示为选择元素$x\in\mathcal{X}=\{x_1,\ldots,x_M\}$。或者，可以有一个连续向量$x=(x_1,x_2,\ldots,x_K)$。设$x^n$是$x$(离散或连续)的选择。假设$x^n$是在运行指导第$n+1$个实验的第$n$个实验后所做的决策，从中观察$W^{n+1}$。结果$W^{n+1}$可能是材料的强度、表面的反射性或杀死的癌细胞的数量。

使用实验结果来更新信念模型。如果x是离散的，假设有一个估计$\bar{\mu}_x^n$，这是在选择x进行实验时对其表现的估计。如果选择$x=x^n$并观察W^{n+1}，之后就可以使用统计方法(参见第3章)来获得更新的估计$\bar{\mu}_x^{n+1}$。事实上，可以使用一种被称为相关信念(correlated belief)的特性对值x'(而非x)进行实验$x=x^n$并更新估计$\bar{\mu}_{x'}^{n+1}$。

通常会使用一些参数模型来预测响应。例如，可以创建如下所示的线性模型：

$$f(x^n|\theta) = \theta_0 + \theta_1\phi_1(x^n) + \theta_2\phi_2(x^n) + ... \tag{2.49}$$

其中，$\phi_f(x^n)$是从实验的输入x^n中提取相关信息的函数。例如，如果元素x_i是温度，则可能有$\phi_1(x^n) = x_i^n$和$\phi_2(x^n) = (x_i^n)^2$。如果x_{i+1}是压力，也可能有$\phi_3(x^n) = x_i^n x_{i+1}^n$和$\phi_4(x^n) = x_i^n(x_{i+1}^n)^2$。

式(2.49)被称为线性模型，因为它在参数向量θ中是线性的。式(2.48)中的逻辑回归模型是非线性模型的一个例子(因为它在θ中是非线性的)。无论是线性还是非线性的，参数信念模型都能捕捉问题的结构，从而减少不确定性，该不确定性可以来自每个x的未知$\bar{\mu}_x$(其中不同x值的数量在数千到数百万甚至更多)或者一组参数θ(这个数字可能在几十到几百之间)。

2.3.8　机器学习与序贯决策问题

序贯决策问题的策略设计和机器学习密切相关。设：

x^n = 希望用来预测结果y^n的与第n个问题实例对应的数据(患者的特征、文档的属性、图像的数据)，

y^n = 该响应可能是患者对治疗的响应、文档的分类或图像的分类，

$f(x^n|\theta)$ = 给定x^n时用来预测y^n的模型，

θ = 用于确定模型的未知参数向量

假设有一些可以用于标识模型$f(x|\theta)$性能的指标。例如，可以使用：

$$L(x^n, y^n|\theta) \quad = \quad (y^n - f(x^n|\theta))^2$$

函数$f(x|\theta)$可以有多种形式。最简单的是基本线性模型：

$$f(x|\theta) = \sum_{f\in\mathcal{F}} \theta_f \phi_f(x)$$

其中，$\phi_f(x)$被称为特征，并且$\mathcal{F}$是一组特征。可能只有少量特征，也可能有数千个特征。统计和机器学习领域开发了广泛的一系列函数，每个函数中都有一些向量θ作参数(有时指定为权重w)。第3章将深入介绍这些函数。

机器学习问题首先会选择一类统计模型$f \in \mathcal{F}$，然后调整与该类函数相关的参数$\theta \in \Theta^f$。可将其写作：

$$\min_{f\in\mathcal{F},\theta\in\Theta^f} \frac{1}{N}\sum_{n=1}^{N}(y^n - f(x^n|\theta))^2 \tag{2.50}$$

在解决序贯决策问题时，需要找到最优策略。我们可以想出一个策略π，包括选择函数$f \in \mathcal{F}$和可调参数$\theta \in \Theta^f$。当编写优化策略的问题公式时，通常使用：

$$\max_{\pi = (f\in\mathcal{F},\theta\in\Theta^f)} \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^\pi(S_t|\theta))|S_0\right\} \tag{2.51}$$

比较机器学习问题(式(2.50))和序贯决策问题(式(2.51))时，不难发现两者都在搜索函数类。第3章会指出，有3类(重叠的)函数用于机器学习：查找表、参数函数和非参数函数。第11章则会指出，有4类策略(即设计策略时$\mathcal{F}$中的4组函数)，其中策略函数近似包括可能在机器学习中使用的所有函数，其他3种都是优化问题的形式。

2.4 参考文献注释

2.1.1节——随机搜索领域的起源可追溯到两篇论文：Robbins和Monro于1951年发表的《论基于导数的随机搜索》、Box和Wilson于1951年发表的《论无导数方法》。一些早期的论文包括Wolfowitz(1952)(使用数值导数)、Blum(1954)(扩展到多维问题)和Dvoretzky(1956)(关于随机近似)的论文，这些论文对理论研究作出了贡献。另一个研究方向集中探讨“随机准梯度”方法所涵盖的约束问题，Ermoliev(1988)、Shor(1979)、Pflug(1988)、Kushner和Clark(1978)、Shapiro和Wardi(1996)，以及Kushner和Yin(2003)为此作出了重要贡献。与其他领域一样，这一领域多年来一直在不断发展和扩大。对随机搜索领域(冠以此名)最好的现代评述是Spall(2003)的评述，这是第一本将当时所流行的随机搜索领域整合在一起的书籍。Bartlett等人(2007)从在线算法的角度探讨了这一主题，在线算法指的是由外源提供样本的随机梯度方法。

具有离散备选方案的无导数随机搜索被当作排序和选择问题广泛研究。排序和选择有着悠久的历史，可以追溯到20世纪50年代，最有代表性的早期研究来自DeGroot(1970)，而Kim和Nelson(2007)曾对此进行过更新的评述。最近的研究重点是并行计算(Luo等人(2015)、Ni等人(2016))和未知相关结构的处理(Qu等人，2012)。然而，排序和选择只是无导数随机搜索的另一个名称，并在此范围内被广泛研究(Spall，2003)。该领域已经引起了模拟优化界的极大关注，下面将对其进行回顾。

2.1.2节——决策树是建模或者简单设置下解决序贯决策问题的最简单方法。决策树可以处理健康(患者是否应该接受MRI检查)、商业(企业是否应该进入新市场)和策略(军队是否应该推行新战略)等方面的复杂决策问题。《决策分析概论》(Skinner，1999)是关于决策树的众多书目之一，其中有几十篇调查文章讨论了决策树在不同应用领域的使用。

2.1.3节——马尔可夫决策过程领域最初由Bellman(1952)以确定性动态规划的形式引入，成就了他的经典参考文献(Bellman，1957)，另见Bellman(1954)和Bellman等人(1955)的研究，但这项研究又持续地引出了一众著作，包括Howard(1960)的著作(另一经典图书)，以及Nemhauser(1966)、Denardo(1982)、Heyman和Sobel(1984)的著作，直到Puterman(2005)(1994年首次提出)的著作出版才告一段落。其中，Puterman的书是最新的一本关于马尔可夫决策过程的书，也是最好的一本，现在算是一个大型理论领域的主要参考文献，因为该领域的核心依赖于一步转移矩阵，而这种矩阵几乎都是不可计算的，而且只适用于极小的问题。最近，Bertsekas(2017)的论著深入总结了动态规划和马尔可夫决策过程领域，使用

的形式混合了最优控制符号和马尔可夫决策过程原理，同时涵盖了近似动态规划和强化学习的许多概念(如下所述)。

2.1.4节——优化控制的发展历史悠久，可追溯到20世纪50年代，许多书对其进行了综述，包括Kirk(2012)、Stengel(1986)、Sontag(1998)、Sethi(2019)及Lewis和Vrabie(2012)的论著。典型的控制问题是连续的、低维的、无约束的，引出一个解析解。当然，应用程序经历了这一典型问题的演变阶段后，才开始使用数值方法。确定性最优控制广泛应用于工程中，而随机最优控制往往涉及更复杂的数学。一些最著名的书包括Astrom(1970)、Kushner和Kleinman(1971)、Bertsekas和Shreve(1978)、Yong和Zhou(1999)，以及Nisio(2014)和Bertsekas(2017)的著作(注意，一些关于确定性控制的书提及了随机情况)。

作为一个一般问题，随机控制问题涵盖了任何序贯决策问题，因此随机控制和其他形式的序贯随机优化之间的区别更像是词汇和符号的区别(Bertsekas(2017)的著作就是一本整合了这些词汇的书)。控制理论思维已广泛应用于库存理论和供应链管理(例如Ivanov和Sokolov(2013)以及Protopappa Sieke和Seifert(2010)的论著)、金融(Yu等人，2010)和医疗服务(Ramirez-Nafarrate等人，2014)等领域。

虽然动态规划(包括马尔可夫决策过程)和最优控制(包括随机控制)领域之间存在相当大的重叠，但这两个领域在很大程度上是独立发展的，使用了不同的符号，并有非常不同的应用。然而，解决这两个领域问题的数值方法的发展过程却有许多相似之处。这两个领域是从同一个基础开始的，这个基础在动态规划中称为贝尔曼方程，在最优控制中则称为哈密顿-雅可比方程(因此一些人将其称为哈密顿-雅可比-贝尔曼(或HJB)方程)。

2.1.5节——自贝尔曼首次认识到离散动态规划遭受“维数灾难”(参见Bellman和Dreyfus(1959)和Bellman等人(1963)的著作)以来，人们便开展了对近似动态规划(也称为自适应动态规划，在一段时间内曾被称为神经动态规划)的研究，但到了20世纪80年代，运筹学界似乎停止了对近似方法的进一步研究。随着计算机的改进，研究人员开始使用数值近似方法处理贝尔曼方程，Judd(1998)在他的著作中最全面地总结了近10年的研究(另见Chen等人(1999)的论著)。

Paul Werbos(1974)认识到可以使用各种技术近似“代价函数”(与动态规划中的价值函数相同)，控制理论界随之发展出了一条完全独立的近似研究路线。Werbos的一系列论文(例如Werbos(1989)、Werbos(1990)、Worbos(1992)和Werbos(1994)的论文)帮助开发了这一领域。重要的参考文献是已经编辑并出版成书的内容，如White和Sofge(1992)和Si等人(2004)的著作，其中突出了使用神经网络来近似策略(“行动者网络”)和价值函数(“评论家网络”)的流行方法。Si等人(2004)对2002年的该领域进行了较为全面的综述。Tsitsiklis(1994)和Jaakkola等人(1994)最先认识到在强化学习范围内开发的基本算法代表了Robbins和Monro(1951)提出的早期随机梯度算法的泛化。Bertsekas和Tsitsiklis(1996)使用“神经动态规划”这一名称为动态规划中的自适应学习算法奠定了基础。Werbos(例如Werbos(1992)的论著)一直在使用“近似动态规划”这一术语，后来该术语还成了

Powell(2007)的著作的书名(之后Powell(2011)对该书进行了大幅更新)。这本书结合了数学规划和价值函数近似来解决高维凸随机优化问题(不过，还请参阅下文以了解随机规划的发展)。后来，随着运筹学界采用“近似动态规划”，工程控制界又回到了“自适应动态规划”。

2.1.6节——第三个近似方法研究方向出现于20世纪80年代，在计算机科学界被称为“强化学习”，标志是Richard Sutton和Andy Barto对Q学习的研究。随着他们的书(Sutton和Barto，2018)面世(现在被广泛引用)，该领域开始蓬勃发展，尽管在此之前该领域也相当活跃(参见Kaelbling等人(1996)的评述)。在“强化学习”范围内进行的研究已经发展到包括其他算法策略，如策略搜索和蒙特卡洛树搜索。强化学习领域的其他参考文献包括Busoniu等人(2010)的著作和Szepsvári(2010)的论文。2017年，Bertsekas出版了探讨最优控制的图书(Bertsekas(2017))的第4版，该书涵盖了一系列主题，包括经典马尔可夫决策过程以及与近似动态规划和最优控制相关的近似算法，但使用最优控制的符号和马尔可夫决策过程的构造(例如一步转移矩阵)。Bertsekas的书对ADP/RL文献进行了最全面的评述，我们建议读者阅读这本书，以获得这些领域的详尽参考书目(截至2017年)。2018年，Sutton和Barto出版了他们经典的*Reinforcement Learning*(《强化学习》)一书的第2版，其篇幅得到了极大的扩充，但方法远远落后于第1版的基础Q学习算法。通过对比《强化学习》的第1版和第2版的语言，读者便可以察觉到从仅基于价值函数的策略(RL领域中的Q学习)到所有4类策略示例的转变。

RL领域的领导者Benjamin van Roy教授在一次研讨会上介绍了“强化学习”的3个特征(例如“对环境采取动作的智能体获得回报”)。

2.1.7节——最优停止是一个古老而经典的话题。Cinlar(1975)曾给出一个巧妙的演示，Cinlar(2011)则给出了更新的讨论，其中，最优停止用于说明筛选。DeGroot(1970)很好地总结了早期文献。Shiryaev(1978)的著作(原版是俄文)是最早关注这一主题的书籍之一。Moustakides(1986)描述了一种用于识别随机过程何时发生变化的应用，例如疾病发病率增加或生产线质量下降。Feng和Gallego(1995)使用最优停止来确定何时开始季节性商品的季末销售。最优停止在金融(Azevedo和Paxson，2014)、能源(Boomsma等人，2012)和技术采用(Hagspiel等人，2015)等领域有很多用途。

2.1.8节——有大量文献利用了x_0中$Q(x_0, W_1)$的自然凸性，从Van Slyke和Wets(1969)的论文开始，随后是关于随机分解的开创性论文(Higle和Sen，1991)和探讨随机双动态规划(stochastic dual dynamic programming，SDDP)的论文(Pereira和Pinto，1991)。学者们围绕这项研究展开了大量文献创作，包括Shapiro(2011)，他对SDDP进行了仔细分析，并将其扩展到风险处理措施(Shapiro等人(2013)，Philpott等人(2013))。基于Benders的解方法的收敛性证明的论文非常多，但最好的是Girardeau等人(2014)的。Kall和Wallace(2009)以及Birge和Louveaux(2011)的著作是随机规划领域的上佳入门书籍。King和Wallace(2012)很好地介绍了将问题建模为随机规划的过程。Shapiro等人(2014)对该领域进行了现代化的概述。

2.1.9节——自1960年以来，应用频率领域一直将主动学习问题作为“多臂老虎机问题”研究。DeGroot(1970)首个证明可以使用贝尔曼方程来制定解决多臂老虎机问题的最优策略(适用于任何学习问题，无论是最大化最终回报还是累积回报)。第一次真正的突破是Gittins和Jones在1974年发表的论文(该领域的第一篇也是最著名的论文)，其次是Gittins(1979)发表的论文。Gittins在他的第一本书(Gittins，1989)中对Gittins指数理论进行了详尽的描述，然而，几乎摒弃了第1版内容的“第2版”(Gittins等人，2011)是对Gittins指数领域的最好介绍，该领域目前已有数百篇论文。然而，该领域对数学要求很高，指数策略很难计算。

Lai和Robbins(1985)的著作在计算机科学界同样掀起了研究浪潮，他们发现，一个被称为上置信区间的简单策略具有一种性质，即测试错误老虎臂的次数可以被限制(尽管它会随n的增大而持续增大)。计算的简便性，加上这些理论性质，使得这一研究领域极具吸引力，引发了热烈关注。虽然目前还没有关于这一主题的书籍，但Bubeck和Cesa Bianchi(2012)曾经发表过一篇专题论文。

与此相同的理念已经应用于通过“最优臂”老虎机问题标签使用终端回报目标的老虎机问题(见Audibert和Bubeck(2010)、Kaufmann等人(2016)、Gabillon等人(2012)的论著)。

2.1.10节——Chun-Hung Chen在其1995年发表的论文中开创了关于最优算力预算分配的研究，随后发表了一系列文章(Chen，1996；Chen等人，1997；Chen等人，1998；Chen等人，2003；Chen等人，2008)，最后Chen和Li(2011)出版了一本书，对该领域进行了全面的概述。该领域主要关注离散备选方案(例如，制造系统的不同设计)，但也包括探讨连续备选方案的论著(例如，Hong和Nelson(2006))。Ryzhov(2016)最近的一个重要研究结果表明OCBA和最大化信息价值的期望改进策略具有渐近等价性。当备选方案的数量很大(例如，10 000个)时，模拟退火、遗传算法和禁忌搜索(适用于随机环境)等技术就应运而生了。Swisher等人(2000)评述了相关文献。其他评述包括Andradóttir(1998a)、Andradóttir(1998b)、Azadivar(1999)、Fu(2002)以及Kim和Nelson(2007)的评述。最近Chau等人(2014)的评述则侧重于基于梯度的方法。

在“模拟优化”范围内研究的问题和方法的范围已经得到稳步增长(这一模式与随机优化中的其他领域相似)。最好的证据是Michael Fu的*Handbook of Simulation Optimization*一书(2014)，该书为该领域的许多工具提供了参考。

2.1.11节——主动学习是机器学习领域中的一个领域；与老虎机问题领域相似，智能体可以控制(或影响)从输入x^n到产生观察结果y^n的学习过程。该领域主要出现在20世纪90年代(特别参见Cohn等人(1996)和Cohn等人(1994)的论文)。Settles(2010)的书对这一领域作了很好的介绍，表明人们强烈意识到主动学习和多臂老虎机问题之间的相似之处。最近，Krempl等人(2016)提供了教程。

2.1.12节——机会约束优化用于处理涉及不确定性的约束，最早由Charnes等人(1959)提出，后由Charnes和Cooper(1963)跟进。它也作为“概率约束规划”被研究

(Prekopa(1971)，Prekopa(2010))，每年都有数百篇论文涉及此主题。机会约束规划是许多随机优化相关书籍中的标准(例如，参见Shapiro等人(2014)发表的论著)。

2.1.13节——模型预测控制是优化控制的一个子领域，但已演变成一个独立的领域，拥有Camacho和Bordons(2003)的著作等热门书籍和数千篇文章(见Lee(2011)的30年评述)。截至本书撰写之时，自2010年以来，已有超过50篇文章对模型预测控制进行了评述。

2.1.14节——Ben Tal等人(2009)和Bertsimas等人(2011)全面评述过鲁棒优化领域，最近的评述参见Gabrel等人(2014)的成果。Bertsimas和Sim(2004)研究了鲁棒性的代价，并描述了一些重要属性。鲁棒优化引发了多个应用领域的研究人员的兴趣，如供应链管理(Bertsimas和Thiele(2006)，Keyvanshokooh等人(2016))、能源(Zugno和Conejo，2015)和金融(Fliege和Werner，2014)。

练习

复习问题

2.1 期望算子的简化式和扩展式的定义是什么？请分别举例说明。

2.2 请写出最大化累积回报或最大化最终回报时使用的目标函数。

2.3 通过创建表来比较2.1.3节中的马尔可夫决策过程模型与2.1.4节中的最优控制模型，此表要显示每种方案如何对以下内容进行建模：

- 状态变量；
- 决策/控制变量；
- 转移函数(使用包含随机性w_t的最优控制公式中的版本)；
- 在t时处于某种状态的价值；
- 给定状态x_t，如何使用该值来查找最优决策(也称为策略)。

2.4 根据本章的简短介绍，讲述近似动态规划和强化学习(使用Q学习)的区别。

2.5 写出一个最优停止问题(作为最优控制问题)。最优策略是否采用式(2.13)中的形式？说明理由。

2.6 求解式(2.23)中的优化问题时是否产生最优策略？请说明原因。

2.7 在式(2.24)的随机规划模型中，“ω”表示什么？使用在0时将库存分配给仓库的设置(此决策由x_0给出)，待了解需求之后再确定哪个仓库应该满足每个需求。

2.8 为多臂老虎机问题编写目标函数，以寻找最优区间估计策略。

2.9 用文字描述在模拟优化中使用OCBA算法优化的决策。(笼统地)对比OCBA的操作与多臂老虎机问题的区间估计。

2.10 主动学习中被优化的目标是什么？你能用区间估计来解决这个问题吗？

2.11　机会约束规划中的核心计算挑战是什么？

2.12　试比较模型预测控制与用作策略的随机规划。

2.13　用文字描述鲁棒优化的核心思想，并举例说明。参照将式(2.24)中的两阶段随机规划写成策略(如式(2.26))的形式，将鲁棒优化也写成策略。

2.14　根据2.3.8节的内容，总结机器学习问题和序贯决策问题之间的区别。

建模问题

2.15　为以下每个问题分别提供3个示例：

(1) 最大化累积回报(或最小化累积成本)；

(2) 最大化最终回报(或最小化最终成本)。

2.16　展示如何使用贝尔曼方程(式(2.7))将决策树(见2.1.2节)作为马尔可夫决策过程(见2.1.3节)进行求解。

2.17　将2.3.1节中的情境性报童问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

2.18　将2.3.2节中带预测的库存计划问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

2.19　将2.3.3节中的动态最短路径问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

2.20　将2.3.3节中的鲁棒最短路径问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

2.21　将2.3.4节中的漂泊的货车司机问题转化为2.2节中通用建模框架的形式。本节中给出的状态变量$S_t = (a_t, \mathcal{L}_t)$不完整。缺少什么？介绍并定义可能需要的任何其他符号。提示：仔细查看2.2节中给出的状态变量的定义。查看式(2.47)中的策略，判断是否有任何用于做出决策的统计数据会随着时间的推移而改变(这意味着它必须进入状态变量)。

2.22　将2.3.5节中的定价问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

2.23　将2.3.6节中的医疗决策问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

2.24　将2.3.7节中的科学探索问题转化为2.2节中通用建模框架的形式。介绍并定义可能需要的任何其他符号。

每日一问

“每日一问”是你选择的一个问题(参见第1章中的指南)。针对你的每日一问，回答以下问题。

2.25 哪些典型问题(可以列举多个)看起来使用了最适合你的每日一问的语言？从你的每日一问中举例，说明其看上去符合一个特定的典型问题。

参考文献

第3章

在线学习

有一个庞大的领域是从统计学、统计学习、机器学习和数据科学等名称演变而来的，该领域的绝大部分研究被称为监督学习，涉及获取数据集(x^n, y^n)、输入数据x^n $(n=1,\ldots,N)$以及相应的观察结果(有时称为“标签”)y^n，并以此设计统计模型$f(x|\theta)$，从而在$f(x^n|\theta)$以及相关观察结果(或标签)y^n之间产生最优匹配。这便是大数据领域。

本书的主题是做决策(x)。那么，为什么需要一个关于学习的章节？简单来说，机器学习是在帮计算机做决策的整个过程中产生的。经典的机器学习专注于学习有关外生过程(exogenous process)的知识：预测天气、预测需求、估计药物或材料的表现。本书关注外生学习(exogenous learning)的原因同上，但大多数时候将关注内生学习(endogenous learning)，即学习价值函数、策略和响应面，这些都是在决策方法的背景下出现的学习问题。

本章开头将概述机器学习在序贯决策中的作用。其余部分则介绍机器学习，重点是随着时间的推移展开学习，这一主题被称为在线学习，因为这将主导机器学习在序贯决策中的应用。

与其他章节一样，本章中标有*的部分在初读时可以跳过。读者应理解本章内容，不然，则应将其当作参考以便需要时查阅(本书其他章节的内容多参考本章)。

3.1 序贯决策的机器学习

有必要通过描述序贯决策背景下出现的学习问题，开始对统计学习的讨论。本节概述了学习问题的以下方面。

- 序贯决策中的观察和数据。经典统计学习问题包含由输入变量(或自变量)x和输出变量(或因变量)y组成的数据集，序贯决策中的因变量x^n是我们控制(至少部分控制)的决策。

- 索引数据。进行批量学习时，使用数据集(x^n, y^n), $n=1,\ldots,N$，其中y^n是与输入数据x^n相关联的响应。在序贯决策的背景下，先选择x^n，然后观察y^{n+1}。
- 正在学习的函数。在不同的随机优化背景下，可能需要对6类不同的函数进行近似。
- 序贯学习。大多数应用都涉及从很少的数据(甚至从零)开始逐步获取更多数据。这通常意味着必须从低维模型(可以用很少的数据拟合)过渡到高维模型。
- 近似策略。这里总结了统计学习文献中的三大类近似策略。本章的其余部分总结了这些策略。
- 目标。有时试图将一个函数与数据相匹配，以最小化误差；有时需要找到一个函数来最大化贡献或最小化成本。无论怎样，学习函数总是涉及其自身的优化问题，有时会隐藏在更大的随机优化问题中。
- 批量学习与递归学习。大多数统计学习文献都侧重于使用给定的数据集(最近，这些数据集非常大)来拟合复杂的统计模型。在序贯决策问题的背景下，我们主要依赖于自适应(或在线)学习，因此本章将讲述递归学习算法。

3.1.1 随机优化中的观察和数据

在介绍统计技术之前，需要先介绍一下用于估计函数的数据。在统计学习中，通常假设给定输入数据x，之后观察响应y。一些示例如下。

- 观察某患者的特征x以预测可能性y(即患者对治疗方案的反应)。
- 根据当下观察到的气象条件x预测天气y。
- 观察附近酒店的定价以及自己酒店房间的价格(表示为x)，预测反应y(即客户是否预订房间)。

在这些设置中，可获得一个数据集，在该数据集中，将响应y^n与观察结果x^n相连，便可以得到一个数据集$(x^n, y^n)_{n=1}^N$。

在序贯决策问题的背景下，x可能是一个决策，例如药物治疗的选择、产品的定价、疫苗库存的确定或在用户的互联网账户上显示的电影的选择。在许多设置中，x可能由可控因素(如药物剂量)和不可控因素(如患者特征)共同组成。但总是可以将机器学习视为通过获取已知信息x来预测或估计未知信息y的过程。

3.1.2 索引输入x^n和响应y^{n+1}

机器学习中的大多数研究都使用可以表示为(x^n, y^n), $n=1,\ldots,N$的批量数据集，其中，x^n是因变量或自变量，y^n是相关的响应(有时称为标签)。

在序贯决策的情况下，可发现基于S^n给出的已知信息以及一些规则或策略$X^\pi(S^n)$来选择决策$x^n = X^\pi(S^n)$更方便。决策x^n基于用于创建状态变量S^n的历史观察结果$y^1,\ldots,y^n$得出。然后观察提供更新状态S^{n+1}的y^{n+1}。注意，从$n=0$开始，其中x^0是必须在看到任

何观察结果之前做出的第一个决策。

这种索引方式与我们对时间进行索引的方式一致，其中$x_t = S^\pi(S_t)$，之后观察W_{t+1}，这是在t和$t+1$之间到达的信息。然而，它会产生不自然的标签。假设在某个医疗环境中，治疗了n名患者。使用从前n名患者中获取的信息S^n，为第$n+1$个患者决定治疗方式，之后观察第$n+1$个患者的响应y^{n+1}(如果使用W符号，则观察W^{n+1})。这看起来很不自然。然而，重要的是遵循以下原则：如果变量由n索引，那么它只取决于前n个观察的信息。

3.1.3 正在学习的函数

在随机优化的许多设置中，都需要近似函数。其中最重要的包括以下设置。

(1) 近似函数的期望值$\mathbb{E}F(x,W)$，使其最大化，假设对于给定的决策x，可以获得无偏观察$\hat{F} = F(x,W)$，这基于统计学习的一个主要分支——监督学习。

(2) 创建近似策略$X^\pi(S|\theta)$。可以从两种方法中选择一种来拟合这些函数。假设可以用决策x的外源来拟合策略$X^\pi(S|\theta)$(这将是监督学习)；更常见的是，调整策略以最大化贡献(或最小化成本)，这有时被称为一种强化学习。

(3) 近似处于状态S的价值$V_t(S_t)$。即使一个或多个S_t的元素是连续的，且/或S_t是多维的，我们也希望找到一个近似值$\overline{V}_t(S_t)$来获得估计值。近似$\mathbb{E}F(x,W)$与$V_t(S_t)$之间的差异就是：$\mathbb{E}F(x,W)$的观察是无偏的，而$V_t(S_t)$的观察依靠“使用次优策略来对引入了偏差的$t+1,t+2,...$做决策”的模拟。

(4) 学习动态系统中的任何基础模型，如下所示。

① 描述系统如何随时间演变的转移函数。可写作$S^M(S_t,x_t,W_{t+1})$，用于计算下一个状态S_{t+1}。这发生在动态未知的复杂环境中，例如建模水库中保留的水量时，需要考虑降雨量和温度的综合结果。可以使用必须估计的参数模型来近似损失。

② 成本或贡献函数(也称为回报、收益、损失函数)。人类是否决定最大化未知效用这件事就可能是未知的，可以将其表示为一个线性模型，其参数由观察到的行为确定。

③ 外部量(如风或价格)的演变，可以在其中将观察结果W_{t+1}建模为历史$W_t,W_{t-1},W_{t-2},...$的函数，并从过去的观察中拟合模型。

可以使用3种策略来解决这类学习问题。

外生学习——一个转移函数的例子是风速w_t的时间序列模型，可以写作：

$$w_{t+1} = \bar{\theta}_{t0}w_t + \bar{\theta}_{t1}w_{t-1} + \bar{\theta}_{t2}w_{t-2} + \varepsilon_{t+1}$$

其中，输入$x_t = (w_t, w_{t-1}, w_{t-2})$以及响应$y_{t+1} = w_{t+1}$支持更新参数向量$\bar{\theta}_t$的估计。响应$y_{t+1}$来自系统外部。

内生学习——可能对以下价值函数有一个估计：

$$\overline{V}_t^n(S_t|\bar{\theta}_t) = \sum_{f\in\mathcal{F}} \bar{\theta}_{tf}^n \phi_f(S_t)$$

然后使用下式生成样本观察$\hat{v}_t^n$：

$$\hat{v}_t^n = \max_{a_t} \left(C(S_t^n, a_t) + \mathbb{E}_{W_{t+1}}\{\overline{V}_{t+1}(S_{t+1}^n|\bar{\theta}^{n-1})|S_t^n\}\right)$$

从而更新参数$\bar{\theta}_t^n$。采样估计值$\hat{v}_t^n$是内生的。

反向优化——假设正在观察某人做决策(玩游戏、管理机器人、调度货车、决定医疗方式)，而没有明确定义的贡献函数$C(S_t, x_t)$。假设可以得到一个参数化的贡献函数$C(S_t, x_t|\theta^{cont})$，没有对贡献的外部观察，也没有内生计算(例如提供贡献的噪声估计的$\hat{v}_t$)，但是给出了实际决策x_t的历史。假设正在使用取决于$C(S_t, x_t|\theta^{cont})$(即取决于$\theta^{cont}$)的策略$X^\pi(S_t|\theta^{cont})$，在这种情况下，策略$X^\pi(S_t|\theta^{cont})$的作用与统计模型完全相似，也就是说，使$\theta^{cont}$在策略$X^\pi(S_t|\theta^{cont})$以及观察到的决策之间得到最优拟合。当然，这是一种外生学习，但决策只暗示了贡献函数应该是什么。

(5) 稍后将介绍的一类策略称为参数化成本函数近似(parametric cost function approximations)，为此，必须学习两类函数：

① 成本函数的参数修改(例如，对当下不满足需求而保留到将来的惩罚)。这与从观察到的决策来估计回报函数(见第(4)项)不同。

② 约束的参数修改(例如，在航空公司时刻表中插入冗余时间以处理行程时间的不确定性)。

必须调整上述每类参数修改(这是函数估计的一种形式)，以随时间产生最优结果。

3.1.4 序贯学习：从很少的数据到更多的数据

在序贯决策问题的背景下，学习问题的一个共同主题是必须自适应地进行学习。这通常意味着，不能只拟合一个模型，而是必须从参数相对较少的模型(可以称之为低维架构)过渡到高维架构。

参数估计的在线更新受到了相当大的关注。它在线性模型的情况下尤其简单，不过在神经网络等非线性模型中较有挑战性。然而，在在线环境中，人们很少关注模型本身的结构更新。

3.1.5 近似策略

统计学习涉及以下几类近似策略。

(1) 查找表。在查找表中，估计函数$f(x)$的x落在离散区域$\mathcal{X}$，由一组点——$x_1, x_2, \ldots, x_M$给出。点x_m可能是一个人、一种材料或一部电影的特征，也可能是离散连续区域中的一个点。只要x是离散元素，$f(x)$就是用于选择x，然后“查找”其值$f(x)$的函数。一些作者称其为“表格”表示。

在大多数应用中，查找表在一个或两个维度上运行良好，然后在三个或四个维度上

变得困难(但可行)，接着很快从五个或六个维度开始变得不切实际。这就是经典的“维数灾难”。我们的演示重点是聚合的使用，尤其是分层聚合的使用，既可以处理“维数灾难”，也可以管理递归估计的过渡：从用很少数据的初始估计，到在更多数据可用时产生更好的估计。

(2) 参数模型。有许多问题都可以根据一些未知参数，使用分析模型来近似函数。

它们分为以下两大类。

线性模型——最简单的参数模型的参数是线性的，可以写作：

$$f(x|\theta) = \theta_0 + \theta_1\phi_1(x) + \theta_2\phi_2(x) + ... \tag{3.1}$$

其中，$(\phi_f(x))_{f\in\mathcal{F}}$是从$x$中提取可能有用信息的特征，它可以是向量，也可以是描述电影或广告的数据。式(3.1)被称为线性模型，因为它相对于θ呈线性(相对于x则可能呈高度非线性)。或者，可以使用非线性模型，例如：

$$f(x|\theta) = e^{\sum_{f\in\mathcal{F}}\theta_f\phi_f(x)}$$

参数模型可以是低维(1~100个参数)或高维(例如几百到几千个参数)。

非线性模型——通常，在选择非线性参数模型的同时选择问题驱动的特定形式。例如阶跃函数(在资产买卖或库存问题中有用)：

$$f(x|\theta) = \begin{cases} -1 & x \le \theta^{low}, \\ 0 & \theta^{low} < x < \theta^{high}, \\ +1 & x \ge \theta^{high} \end{cases} \tag{3.2}$$

或逻辑回归(适用于定价和推荐问题)：

$$f(x|\theta) = \frac{1}{1 + e^{\theta_0+\theta_1 x_1+...}} \tag{3.3}$$

有些模型(例如神经网络)的主要优点是不强加任何结构，这意味着它们可以近似任何东西(特别是深度神经网络)。这些模型可以有数万到数亿个参数。毫不奇怪，它们需要非常大的数据集来确定这些参数。

(3) 非参数模型。非参数模型通过直接从数据构建结构来创建估计。例如，根据(f^n, x^n), $n = 1, ..., N$的附近观察值的加权组合估计$f(x)$。也可以通过局部线性近似来构造近似。

三类统计模型——查找表、参数和非参数，应被看作重叠的集合，如图3.1所示。例如，下面描述的神经网络可以分为参数模型(较简单的神经网络)或非参数模型(深度神经网络)。其他方法是有效的混合方法，例如基于树回归的方法，它可以围绕输入数据的特定区域(区域的定义是查找表)创建线性近似(参数化)。

本章缺少凸函数的近似方法。在许多应用中，$F(x, W)$在x中呈凸性。这个函数非常特殊，因此留到第5章(尤其是第18章)详细介绍，届时将处理随机凸(或凹)随机优化问题，例如具有随机数据的线性规划问题。

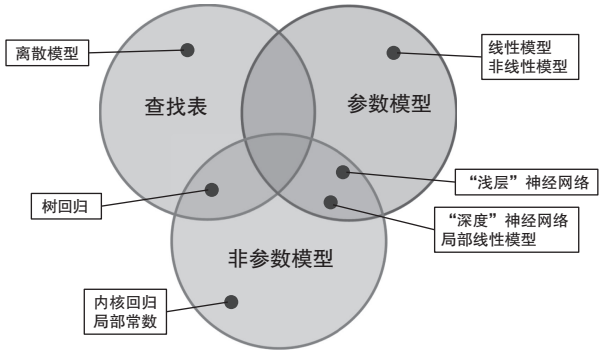

图3.1 查找表、参数模型和非参数模型之间的重叠

我们从查找表开始阐述，这是表示函数而不假定任何结构的最简单方法。首先从频率论和贝叶斯角度讲解查找表。序贯决策问题需要两个信念模型。一般而言，贝叶斯模型最适合用于能够获得一些先验信息并且函数评估成本高的情况。

3.1.6 从数据分析到决策分析

可以从两个广泛目标的角度处理序贯决策问题中的学习。

- 学习函数——我们可能要学习函数的近似值，例如目标函数$\mathbb{E}F(x,W)$或价值函数$V(s)$甚至转移函数$S^M(s,x,W)$。在这些设置中，假设有一个观察函数的源，它可能是有噪声的，甚至是有偏差的。例如，可以获得$\mathbb{E}F(x^n,W^{n+1})$的噪声观察值y^{n+1}并用函数$f(x|\theta)$来近似它。如果收集数据集$(x^0,y^1,x^1,y^2,\ldots,x^{n-1},y^n)$，我们会使用下式来寻找将观察值$y$和$f(x|\theta)$之间的误差最小化的$\theta$：

$$\min_{\theta}\frac{1}{N}\sum_{n=0}^{N-1}\left(y^{n+1}-f(x^n|\theta)\right)^2 \tag{3.4}$$

- 最大化回报(或最小化成本)——可以使用下式搜索最大化贡献函数$C(S,x)$的策略$X^\pi(S|\theta)$：

$$\max_{\theta}\mathbb{E}C(S,X^\pi(S))\approx\frac{1}{N}\sum_{n=0}^{N-1}C(S^n,X^\pi(S^n|\theta)) \tag{3.5}$$

其中，状态根据已知的转移函数$S^{n+1}=S^M(S^n,x^n,W^{n+1})$演化。

式(3.4)中的目标函数常见于经典机器学习，我们将其归入“数据分析”范围。表示目标的方法有很多种，例如，可能想使用$|y^{n+1}-f(x^n|\theta)|$，但它们总是涉及来自模型的预测$f(x|\theta)$和观察结果y。

式(3.5)中的目标函数常见于优化问题，我们将其归入“决策分析”范围。它假定了某种形式的预定义表现指标(成本、贡献、回报、效用)，并且不需要外生数据集$(y^n)_{n=1}^N$。

3.1.7 批量学习与在线学习

式(3.4)(或式(3.5))是批量学习问题中的标准问题，其中使用一个固定的数据集(在当下的“大数据”时代可能是一个非常大的数据集)来拟合一个模型(维度越来越高的模型，如下面介绍的神经网络)。

虽然批量学习可能出现在随机优化中，但最常见的学习问题是自适应的，这意味着在新数据到达时更新估计，就像在线应用程序中发生的那样。假设n次迭代(或样本)后，有以下序列：

$$(x^0, W^1, y^1, x^1, W^2, y^2, x^2, \dots, W^n, y^n)$$

假设使用这些数据来获得称为$\bar{F}^n(x)$的函数估计。现在假设使用这个估计来做决策x^n，之后得到外生信息W^{n+1}，然后是响应y^{n+1}。需要使用先验估计$\bar{F}^n(x)$及新信息(W^{n+1}, y^{n+1})以产生新的估计$\bar{F}^{n+1}(x)$。

当然，只需要再观察一次就可以解决一个新的批量问题。这在计算上可能要求很高，而且会对整个历史产生同等的影响。某些情况下，最近的观察更重要。

3.2 使用指数平滑的自适应学习

用于自适应学习的最常见方法有各种各样的名称，但通常被称为指数平滑(exponential smoothing)。假设有一个对某个量的观察结果的序列，例如预订房间的人数、患者对特定药物的反应或路径上的行程时间。设μ是未知的事实，可能是以特定价格预订房间的平均人数，或者患者对药物的反应概率，或者路径的平均行程时间，想从观察序列中估计平均值。

设W^n是试图估计的量的第n次观察，$\bar{\mu}^n$是n次观察后的真实平均值μ的估计。在给定$\bar{\mu}^n$和一次新的观察W^{n+1}的情况下，应用最广泛的计算$\bar{\mu}^{n+1}$的方法见下式：

$$\bar{\mu}^{n+1} = (1-\alpha_n)\bar{\mu}^n + \alpha_n W^{n+1} \tag{3.6}$$

第5章将对式(3.6)使用随机梯度算法策略，以解决特定的优化问题。目前，可以说，这一基本公式将在各种在线学习问题中频繁出现。

毫不奇怪，这种方法的最大挑战是α_n的选择。变量α_n被称为学习率、平滑因子或(本书中的)步长(第5章将分析使用“步长”一词的原因)。这个主题非常丰富，因此第6章将专门介绍这个主题。

现在，可以概括出一些简单策略。

- 恒定步长——最简单的策略是实际广泛使用的策略，即简单设置$\alpha_n = \bar{\alpha}$，其中$\bar{\alpha}$是预先选择的常数。

- 谐波步长——这是一个算术递减序列：

$$\alpha_n = \frac{\theta^{step}}{\theta^{step} + n - 1}$$

如果$\theta^{step} = 1$，则$\alpha_n = 1/n$(第6章将表明，这产生了一个简单的平均值)。通常这种步长下降得太快。可增大θ^{step}以减缓步长的下降，从而加速学习。也可能有接近极限点的下降序列。

- 第6章还会介绍一系列响应数据的自适应步长。

3.3 使用频率更新的查找表

频率论观点可以说是具有统计学入门知识的人最熟悉的方法。假设试图估计随机变量W的平均值μ，随机变量可能是设备或策略的表现。设W^n是第n次样本观察，例如产品的销售或特定药物实现的血糖降低。设$\bar{\mu}^n$是对μ的估计，$\hat{\sigma}^{2,n}$是对W的方差的估计。基于基础统计学，可以将$\bar{\mu}^n$和$\hat{\sigma}^{2,n}$写作：

$$\bar{\mu}^n = \frac{1}{n}\sum_{m=1}^{n} W^m \tag{3.7}$$

$$\hat{\sigma}^{2,n} = \frac{1}{n-1}\sum_{m=1}^{n}(W^m - \bar{\mu}^n)^2 \tag{3.8}$$

估计量$\bar{\mu}^n$是一个随机变量(从频率论视角看)，因为它是根据其他随机变量(即$W^1, W^2, \ldots, W^n$)计算的。假设我们让100个人每人选择一个样本，样本包含对W的n次观察。我们将获得$\bar{\mu}^n$的100个不同的估计，反映了我们对W的观察的不同。估计量$\bar{\mu}^n$的方差的最优估计如下：

$$\bar{\sigma}^{2,n} = \frac{1}{n}\hat{\sigma}^{2,n}$$

注意，随着$n \to \infty$，$\bar{\sigma}^{2,n} \to 0$，但$\hat{\sigma}^{2,n} \to \sigma^2$，其中$\sigma^2$是$W$的真实方差。如果$\sigma^2$是已知的，则不必计算$\hat{\sigma}^{2,n}$，$\bar{\sigma}^{2,n}$将由上式与$\hat{\sigma}^{2,n} = \sigma^2$一起求得。

可以递归地写出如下表达式：

$$\bar{\mu}^n = \left(1 - \frac{1}{n}\right)\bar{\mu}^{n-1} + \frac{1}{n}W^n \tag{3.9}$$

$$\hat{\sigma}^{2,n} = \frac{n-2}{n-1}\hat{\sigma}^{2,n-1} + \frac{1}{n}(W^n - \bar{\mu}^{n-1})^2, \quad n \geq 2 \tag{3.10}$$

我们经常谈及信念状态，该状态捕捉了我们试图估计的参数的已知信息。根据观察，信念状态可以用下式表示：

$$B^n = \left(\bar{\mu}^n, \hat{\sigma}^{2,n}\right)$$

式(3.9)和式(3.10)描述了信念状态如何随时间演变。

3.4 使用贝叶斯更新的查找表

贝叶斯视角对我们计算的统计数据给出了不同的解释，这在观察成本较高时(假设必须运行成本较高的模拟或现场实验)，在学习的背景下特别有用。从频率论的角度来看，在收集任何数据之前，我们不会先了解系统。很容易从式(3.9)和式(3.10)中证实我们从未使用过$\bar{\mu}^0$或$\hat{\sigma}^{2,0}$。

相比之下，从贝叶斯视角看，假设我们从关于未知参数μ的信念的先验分布开始。换句话说，任何不知道其值的数字都被解释为一个随机变量，而这个随机变量的分布代表了我们对μ取某些值的可能性的信念。因此，如果μ是W的真实但未知的平均值，我们可能会说，虽然不知道平均值是什么，但我们认为这是围绕θ^0的具有标准差σ^0的正态分布。

因此，真正的平均值μ被视为具有已知平均值和方差的随机变量，但我们愿意在收集额外信息时调整平均值和方差的估计。如果添加一个分布假设，如正态分布，我们会说这是初始信念分布，通常称为贝叶斯先验。

贝叶斯视角非常适合收集有关观察成本较高的过程的信息的问题。当试图在互联网上为一本书定价或计划一项成本高昂的实验时，可能会出现这种情况。在这两种情况下，都可以获得先验信息：关于一本书的适当价格，或者使用物理和化学知识进行实验的行为。

我们注意到，从概率论角度看，符号有微妙的变化，其中$\bar{\mu}^n$给出了我们对μ的估计。在贝叶斯视角中，我们设$\bar{\mu}^n$是在完成n次观察后，对随机变量μ的平均值的估计。重要的是记住μ是一个随机变量，其分布反映了我们对μ的先验信念。参数$\bar{\mu}^0$不是随机变量。这是我们对先验分布平均值的初步估计。n次观察后，$\bar{\mu}^n$是我们对随机变量μ的平均值(真正的平均值)的更新估计。

下面首先使用一些简单的概率表达式来说明收集信息的效果。然后，对于独立信念的情况，给出式(3.9)和式(3.10)的贝叶斯版本，其中对一个选择的观察不会影响对其他选择的信念。我们通过给出相关信念的更新公式来继续这一讨论，其中对备选方案x的观察μ_x会帮助了解$\mu_{x'}$。最后通过讨论其他重要的分布类型来完善演示。

3.4.1 独立信念的更新公式

先假设(正如在大部分演示中所做的那样)随机变量W呈正态分布。设σ_W^2是W的方差，它捕获了观察真实值的能力中的噪声。为了简化代数，可以按如下方式定义W的精度：

$$\beta^W = \frac{1}{\sigma_W^2}$$

精度有一个直观的含义：方差较小意味着观察值将更接近未知平均值，也就是说，它们将更精确。

现在设$\bar{\mu}^n$是n次观察后对μ的真实平均值的估计，β^n是这个估计的精度。如果观察

W^{n+1}，则$\bar{\mu}^n$和β^n将根据以下两式更新：

$$\bar{\mu}^{n+1} = \frac{\beta^n \bar{\mu}^n + \beta^W W^{n+1}}{\beta^n + \beta^W} \tag{3.11}$$

$$\beta^{n+1} = \beta^n + \beta^W \tag{3.12}$$

第7章中的式(7.26)~式(7.27)是式(3.9)~式(3.10)的贝叶斯对应式，不过，通过假设W的方差已知，我们已稍稍简化了问题。贝叶斯视角中的信念(具有正态分布的信念)状态由如下信念状态式给出：

$$B^n = (\bar{\mu}^n, \beta^n)$$

如果有关μ的信念的先验分布呈正态，且观察W也呈正态分布，则后验分布也呈正态。事实证明，经过几次观察(也许5到10次)，由于大数定律，对于W的几乎任何分布，关于μ的信念分布将近似呈正态分布。出于同样的原因，无论W的分布如何，后验分布都将近似呈正态分布！因此，更新的式(7.26)和式(7.27)将产生几乎所有问题的正态分布的平均值和精度！

3.4.2 相关信念的更新

接下来进行转移，现在用一个从集合$\mathcal{X} = \{x_1, \ldots, x_M\}$中选择的向量$\mu_{x_1}, \mu_{x_2}, \ldots, \mu_{x_M}$来代替数字$\mu$。可以将$\mu$的一个元素表示为$\mu_x$，这可能是在$x$处对函数$\mathbb{E}F(x, W)$的估计。通常，$\mu_x$和$\mu_{x'}$是相关的，当$x$是连续的，并且$x$和$x'$彼此相近时，也是如此。有许多例子说明了所谓的相关信念(correlated belief)是什么。

■ 示例3.1

我们有兴趣找到使总收入最大化的产品价格。认为将收入与价格联系起来的函数$R(p)$是连续的。假设设定了一个价格p^n并观察高于预期的收入R^{n+1}。当价格为p^n时，如果提高对函数$R(p)$的估计，则对邻近价格收入的信念应该更高。

■ 示例3.2

我们选择五个人组成篮球队的首发阵容，并观察其一段时间的总得分。试图判断这个五人组是否比另一个由同一组的三个人与另外两个人组成的阵容更好。如果这五个人的得分高于预期，则可能会提高我们对另一组的信心，因为有三个人是相同的。

■ 示例3.3

一位医生正在尝试用三种药物治疗糖尿病，她观察到特定治疗过程中病人血糖值的下降。若一种治疗产生了比预期更好的反应，那么对于其他有一两种相同药物的治疗，我们将更有信心获得好的反馈。

■ 示例3.4

我们努力寻找病毒浓度最高的群体。如果一组人身体中病毒的浓度高于预期，我们会预期其他亲密群体(无论是地理位置邻近还是有其他亲密关系)的身体中的病毒浓度也将较高。

相关信念是学习函数的一个特别强大的工具，支持将单次观察的结果推广到未直接测量的其他备选方案。

设$\bar{\mu}_x^n$是我们在n次测量后对备选方案x的信念。现在有：

$Cov^n(\mu_x, \mu_y) =$ 给定前n次观察的情况下，有关μ_x和μ_y信念的协方差

设Σ^n是协方差矩阵，带有元素$\Sigma_{xy}^n = Cov^n(\mu_x, \mu_y)$。正如前面将精度$\beta_x^n$定义为方差的逆矩阵，此处可以将精度矩阵$M^n$定义为：

$$M^n = (\Sigma^n)^{-1}$$

设e_x是零的列向量，元素x为1，和之前一样，设W^{n+1}是当我们决定测量备选方案x时的(标量)观察值。可以将W^{n+1}标记为W_x^{n+1}，以使对备选方案的依赖更加明确。本次讨论中将使用我们选择的用来测量x^n的符号，得到的观察值是W^{n+1}。

如果选择测量x^n，还可以将观察结果解释为$W^{n+1}e_{x^n}$给出的列向量。记住，$\bar{\mu}^n$是我们对μ的期望的信念的列向量，在存在相关信念的情况下，更新该向量的贝叶斯公式如下：

$$\bar{\mu}^{n+1} = (M^{n+1})^{-1}\left(M^n\bar{\mu}^n + \beta^W W^{n+1}e_{x^n}\right) \tag{3.13}$$

其中，M^{n+1}由下式给出：

$$M^{n+1} = (M^n + \beta^W e_{x^n}(e_{x^n})^T) \tag{3.14}$$

注意，$e_x(e_x)^T$是一个零矩阵，在行x、列x有一个1，而β^W是给出测量W的精度的标量。

可以执行这些更新而不必处理协方差的逆矩阵。这通过谢尔曼-莫里森(Sherman-Morrison)公式来完成。如果A是可逆矩阵(如Σ^n)，u是列向量(例如e_x)，那么谢尔曼-莫里森公式为：

$$[A + uu^T]^{-1} = A^{-1} - \frac{A^{-1}uu^TA^{-1}}{1 + u^TA^{-1}u} \tag{3.15}$$

该公式的推导参见3.14.2节。

使用谢尔曼-莫里森公式，且设$x = x^n$，可以将更新公式改写为：

$$\bar{\mu}^{n+1}(x) = \bar{\mu}^n + \frac{W^{n+1} - \bar{\mu}_x^n}{\sigma_W^2 + \Sigma_{xx}^n}\Sigma^n e_x \tag{3.16}$$

$$\Sigma^{n+1}(x) = \Sigma^n - \frac{\Sigma^n e_x(e_x)^T\Sigma^n}{\sigma_W^2 + \Sigma_{xx}^n} \tag{3.17}$$

其中，表达了$\bar{\mu}^{n+1}(x)$和$\Sigma^{n+1}(x)$与我们选择测量的备选方案x的相关性。

为了进行说明，可假设有3个备选方案，其平均向量为：

$$\bar{\mu}^n = \begin{bmatrix} 20 \\ 16 \\ 22 \end{bmatrix}$$

假设$\sigma_W^2 = 9$，协方差矩阵Σ^n如下：

$$\Sigma^n = \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix}$$

假设选择测量$x = 3$并观察到$W^{n+1} = W_3^{n+1} = 19$。

代入式(3.16)中，可使用下式更新信念的平均值：

$$\begin{aligned} \bar{\mu}^{n+1}(3) &= \begin{bmatrix} 20 \\ 16 \\ 22 \end{bmatrix} + \frac{19-22}{9+15} \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 20 \\ 16 \\ 22 \end{bmatrix} + \frac{-3}{24} \begin{bmatrix} 3 \\ 4 \\ 15 \end{bmatrix} \\ &= \begin{bmatrix} 19.625 \\ 15.500 \\ 20.125 \end{bmatrix} \end{aligned}$$

使用下式计算协方差矩阵的更新：

$$\begin{aligned} \Sigma^{n+1}(3) &= \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix} - \frac{\begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [0\ 0\ 1] \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix}}{9+15} \\ &= \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix} - \frac{1}{24} \begin{bmatrix} 3 \\ 4 \\ 15 \end{bmatrix} [3\ 4\ 15] \\ &= \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix} - \frac{1}{24} \begin{bmatrix} 9 & 12 & 45 \\ 12 & 16 & 60 \\ 45 & 60 & 225 \end{bmatrix} \\ &= \begin{bmatrix} 12 & 6 & 3 \\ 6 & 7 & 4 \\ 3 & 4 & 15 \end{bmatrix} - \begin{bmatrix} 0.375 & 0.500 & 1.875 \\ 0.500 & 0.667 & 2.500 \\ 1.875 & 2.500 & 9.375 \end{bmatrix} \\ &= \begin{bmatrix} 11.625 & 5.500 & 1.125 \\ 5.500 & 6.333 & 1.500 \\ 1.125 & 1.500 & 5.625 \end{bmatrix} \end{aligned}$$

这些计算相当简单。这意味着即使有几千个备选方案，也可以执行。然而，如果备选方案的数量为10^5或更多个时，则该方法也将不可行。在考虑的问题中，备选方案x本身是一个多维向量时会发生这种情况。

3.4.3 高斯过程回归

近似连续函数的一种常见策略是将其离散化，然后通过注释来表明附近点的值是相关的，从而捕捉连续性，这得益于连续性，被称为高斯过程回归(Gaussian process regression，GPR)。

假设有一个未知函数$f(x)$在x中是连续的，当前假设x是一个标量，离散化为值$(x_1, x_2, \ldots, x_M)$。设$\bar{\mu}^n(x)$是对$f(x)$在离散集合上的估计。设$\mu(x)$是$f(x)$的真实值，在贝叶斯视角下将其解释为一个平均值为$\bar{\mu}_x^0$和方差为$(\sigma_x^0)^2$(这是我们的先验知识)的正态分布随机变量。接下来进一步假设μ_x和$\mu_{x'}$是相关的，具有以下协方差：

$$Cov(\mu_x, \mu_{x'}) = (\sigma^0)^2 e^{\alpha\|x-x'\|} \tag{3.18}$$

其中，$\|x-x'\|$是诸如$|x-x'|$或$(x-x')^2$(如果x是标量)或$\sqrt{\sum_{i=1}^{I}(x_i-x_i')^2}$(如果$x$是向量)的距离指标。如果$x = x'$，那么可以在关于$\mu_x$的信念中提取方差。参数$\alpha$捕捉$x$和$x'$相距越来越远时的相关程度。

图3.2展示的是使用式(3.18)中给出的不同α值的协方差函数，从信念模型中随机生成的一系列曲线。因为较小的α代表相距较远的x和x'值之间的较高协方差，所以较小的α值生成的曲线更少起伏，较为平滑。随着α的增大，协方差下降，曲线上的两个不同点变得更加独立。

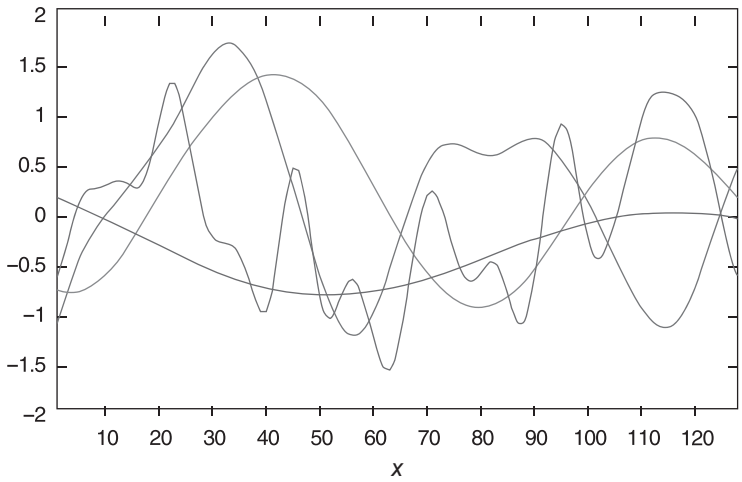

图3.2 使用不同α值的高斯过程回归(相关信念)生成的一系列函数

高斯过程回归(通常简称为GPR)是近似连续但没有特定结构的平滑函数的一种强大方法。这里将GPR作为查找表信念模型的一种泛化，但它也可以被描述为一种非参数统计数据，参见后面的讲解。第7章将展示如何使用GPR信念模型显著加速如下连续参数的优化函数：医疗应用中药物剂量的选择或实验室科学应用中温度、压力和浓度的选择。

3.5 计算偏差和方差*

使用查找表估计多维向量函数的一种强大策略是分层聚合，即在不同的聚合级别上估计函数。这种方法的基础是统计估计中的偏差和方差的基本结果。

假设想估计一个可观察到的真实但未知的参数μ，但必须处理偏差β和噪声ε，则有：

$$\hat{\mu}^n = \mu + \beta + \varepsilon^n \tag{3.19}$$

其中，μ和β都未知，但可以假设有某种方法对将要调用$\hat{\beta}^n$的偏差进行噪声估计。稍后，将提供示例说明如何获取β的估计。

现在假设$\bar{\mu}^n$是在n次观察后对μ的估计。可对$\bar{\mu}^n$使用以下递归公式：

$$\bar{\mu}^n = (1-\alpha_{n-1})\bar{\mu}^{n-1} + \alpha_{n-1}\hat{\mu}^n$$

我们有意估计$\bar{\mu}^n$的方差及其偏差$\bar{\beta}^n$。先计算$\bar{\mu}^n$的方差。假设可以使用式(3.19)表示对μ的观察，其中$\mathbb{E}\varepsilon^n = 0$，$Var[\varepsilon^n] = \sigma^2$。利用该模型，可以参照下式计算$\bar{\mu}^n$的方差：

$$Var[\bar{\mu}^n] = \lambda^n\sigma^2 \tag{3.20}$$

其中，λ^n(这表示第n次迭代的λ，而不是λ的n次幂)可以通过简单的递归计算：

$$\lambda^n = \begin{cases} \alpha_{n-1}^2 & n = 1, \\ (1-\alpha_{n-1})^2\lambda^{n-1} + \alpha_{n-1}^2 & n > 1 \end{cases} \tag{3.21}$$

为此，从$n = 1$开始。对于给定的(确定的)初始估计$\bar{\mu}^0$，首先观察到$\bar{\mu}^1$的方差由下式给出：

$$\begin{aligned} Var[\bar{\mu}^1] &= Var[(1-\alpha_0)\bar{\mu}^0 + \alpha_0\hat{\mu}^1] \\ &= \alpha_0^2 Var[\hat{\mu}^1] \\ &= \alpha_0^2\sigma^2 \end{aligned}$$

对于$\bar{\mu}^n(n > 1)$，使用归纳法证明。假设$Var[\bar{\mu}^{n-1}] = \lambda^{n-1}\sigma^2$。然后，由于$\bar{\mu}^{n-1}$和$\hat{\mu}^n$无关，因此有：

$$\begin{aligned} Var[\bar{\mu}^n] &= Var\left[(1-\alpha_{n-1})\bar{\mu}^{n-1} + \alpha_{n-1}\hat{\mu}^n\right] \\ &= (1-\alpha_{n-1})^2 Var\left[\bar{\mu}^{n-1}\right] + \alpha_{n-1}^2 Var[\hat{\mu}^n] \\ &= (1-\alpha_{n-1})^2\lambda^{n-1}\sigma^2 + \alpha_{n-1}^2\sigma^2 \qquad (3.22) \\ &= \lambda^n\sigma^2 \qquad (3.23) \end{aligned}$$

假设(在归纳证明中)式(3.22)正确，而式(3.23)建立了式(3.21)中的递归。这便可得到方差，当然要假设σ^2是已知的。

假设可以获得偏差的有噪声估计β^n，那么可以采用下式计算均方误差：

$$\mathbb{E}\left[\left(\bar{\mu}^{n-1} - \bar{\mu}^n\right)^2\right] = \lambda^{n-1}\sigma^2 + \beta^{2,n} \tag{3.24}$$

参见练习3.11以证明这一点。该式给出了已知的平均值$\bar{\mu}^n$周围的方差。也可在观察值$\hat{\mu}^n$周围设置方差。设：

$$\nu^n = \mathbb{E}\left[\left(\bar{\mu}^{n-1} - \hat{\mu}^n\right)^2\right]$$

是当前估计值$\bar{\mu}^{n-1}$和观察值$\hat{\mu}^n$之间的均方误差(包括噪声和偏差)。可以证明(见练习3.12)：

$$\nu^n = (1 + \lambda^{n-1})\sigma^2 + \beta^{2,n} \tag{3.25}$$

其中，使用式(3.21)计算λ^n。

实际上，我们不知道σ^2，当然也不知道偏差β。因此，必须从数据中估计这两个参数。先提供偏差的估计值：

$$\bar{\beta}^n = (1 - \eta_{n-1})\bar{\beta}^{n-1} + \eta_{n-1}\beta^n$$

其中，η_{n-1}是用于估计偏差和方差的(通常简单的)步长规则。一般来说，η_{n-1}得出的步长应大于α_{n-1}，这是因为我们更感兴趣的是跟踪真实信号，而非生成具有低方差的估计。我们发现，恒定步长(如0.10)对大多数问题都非常有效，但如果需要精确收敛，则有必要使用步长为零的规则，比如谐波步长规则(式(6.15))。

要估计方差，首先需要找到总方差ν^n的估计值。设$\bar{\nu}^n$是总方差的估计值，可以使用下式计算：

$$\bar{\nu}^n = (1 - \eta_{n-1})\bar{\nu}^{n-1} + \eta_{n-1}(\bar{\mu}^{n-1} - \hat{\mu}^n)^2$$

使用$\bar{\nu}^n$作为对总方差的估计，可以使用下式计算σ^2的估计值：

$$\bar{\sigma}^{2,n} = \frac{\bar{\nu}^n - \bar{\beta}^{2,n}}{1 + \lambda^{n-1}}$$

可以使用式(3.20)获取$\bar{\mu}^n$方差的估计值。

如果求真正的平均(使用步长$1/n$)，通过使用小样本方差公式的递归形式，就可以得到小样本方差的更精确估计：

$$\hat{\sigma}^{2,n} = \frac{n-2}{n-1}\hat{\sigma}^{2,n-1} + \frac{1}{n}(\bar{\mu}^{n-1} - \hat{\mu}^n)^2 \tag{3.26}$$

$\hat{\sigma}^{2,n}$是$\hat{\mu}^n$的方差的估计值。可以使用下式计算估计值$\bar{\mu}^n$的方差：

$$\bar{\sigma}^{2,n} = \frac{1}{n}\hat{\sigma}^{2,n}$$

可在以下两种情况下利用这些结果，这两种情况的区别在于对偏差β^n的估计的计算方式。

- 分层聚合——在不同的聚合级别上估计函数。可以假设在最解聚(disaggregate)水平的函数的估计是有噪声但无偏的，然后设某个聚合水平的函数和最解聚水平的函数之间的差为偏差的估计。
- 瞬时函数——稍后将使用这些结果来近似价值函数。这是估计基础过程随时间变化

的价值函数算法的优先算法(详见第14章)。在这种情况下，我们根据一个随时间变化的事实进行观察，而这会引入偏差。

3.6 查找表和聚合*

查找表是表示函数的最简单和最通用的方法。如果想建模函数$f(x) = \mathbb{E}F(x,W)$或者价值函数$V_t(S_t)$，就可以假设函数是在一组离散的值$x_1,\ldots,x_M$(或离散状态$\mathcal{S} = \{1,2,\ldots,|\mathcal{S}|\}$)上定义的。我们希望使用对函数的观察(可能是$f^n = F(x^n,W^{n+1})$或源自对处于状态$S_t$的价值的模拟$\hat{v}_t^n$)，以创建估计$\bar{F}^{n+1}$(或$\overline{V}_t^{n+1}(S_t)$)。

查找表表示的问题是，如果变量x(或状态S)是一个向量，则可能值的数量会随维数增加呈指数增长。这是经典的“维数灾难”。克服“维数灾难”的一种策略是使用聚合，但选择单一级别的聚合通常不会令人满意。特别是，通常必须从无数据开始，并稳定地建立函数的估计。

通过使用分层聚合，可以实现从少数据到无数据，再到不断增加的观察数量的转变。我们使用一系列具有分层结构的聚合，而非选择单一级别的聚合。

3.6.1 分层聚合

函数的查找表表示通常是优先考虑的策略，因为它不要求采取任何结构化形式。但问题是查找表会遭受维数灾难。一种可以扩展查找表的强大策略是使用分层聚合。不是简单地将一个状态空间聚合到一个更小的空间中，而是提出一个聚合族，然后根据在每个聚合级别上的估计将它们组合起来。这不是万能的，也不应该被视为“解决维数灾难”的方法，但它确实是对近似策略的有力补充。正如你将看到的那样，这在应用于序贯决策问题时特别有用。

可以使用2.3.4节中首次介绍的漂泊的货车司机示例来说明分层聚合。在这个示例中，管理的是一个装卸货物的货车司机(想象一下货运出租车)，司机必须根据运送货物的收入和到达货物目的地的价值来选择货物。使问题复杂化的是，司机由多维属性向量$a = (a_1,a_2,\ldots,a_d)$描述，包括诸如货车的位置(这意味着在某个地区中的位置)、司机的设备类型和家庭住所(同样是一个地区)等属性。

如果用状态向量$S_t = a_t$来描述漂泊的货车司机，对状态向量采取动作x_t(移动一批可用货物)，那么转移函数$S_{t+1} = S^M(S_t,x_t,W_{t+1})$就可以表示高细节级别的状态向量(一些值可以是连续的)。但决策问题

$$\max_{x_t\in\mathcal{X}}\left(C(S_t,x_t) + \mathbb{E}\{\overline{V}_{t+1}(G(S_{t+1}))|S_t\}\right) \tag{3.27}$$

使用价值函数$\overline{V}_{t+1}(G(S_{t+1}))$，其中，$G(\cdot)$是将原始(非常详细的)状态$S$映射为更简单内容的聚合函数。聚合函数$G$可以忽略维度、对其进行离散化，或者使用某种方法来减少状

态向量的可能值的数量。这也减少了必须估计的参数数量。在下面的内容中，我们删除了聚合函数G的显式引用，只使用$\overline{V}_{t+1}(S_{t+1})$。聚合在价值函数近似中是隐式的。

可用于聚合的一些主要特征列举如下。

- 空间。运输公司有兴趣评估货车司机在特定地点的价值。地点可以按5位数的邮政编码(美国约有55 000个)、3位数的邮政编码(美国约有1000个)或州(美国本土48个州)等不同级别计算。
- 时间。银行可能有兴趣估计某一时间点持有资产的价值。时间可以用天、周、月或季度来衡量。
- 连续参数。飞机的状态可能是其燃油油位；旅行推销员的状态可能是他离家的时长；蓄水池的状态可以是水的深度；共同基金的现金储备状态是一天结束时持有的现金量。以上皆是具有至少近似连续状态的至少一个维度的系统示例。变量可以全部离散为不同长度的区间。
- 分级分类。投资组合问题可能需要估计投资特定公司股票的价值。按行业对公司进行汇总的做法可能很有用(例如，某公司可能在化工行业，可能会根据其被视为国内公司还是跨国公司进行进一步汇总)。同理，对于管理大量零件库存(例如汽车)的问题，最好将零件组织到零件族(变速器零件、发动机零件、仪表板零件)。

下面的示例提供了补充说明。

■ 示例3.5

喷气式飞机的状态可以由多个属性表征，这些属性包括空间和时间维度(位置，或者自上次维护检查以来的飞行时间)等属性。一个连续的参数可以是燃油油位——一个有助于分级聚合的属性可以是特定类型的飞机。可以通过将每个维度聚合为较少数量的潜在结果，来减少该资源的状态(属性)数量。

■ 示例3.6

投资组合的状态可能包括债券的数量，其特征在于债券的来源(公司、自治市或联邦政府)、到期日(6个月、12个月、24个月)、购买时间以及债券机构的评级。公司可以按行业分类聚合。债券可以通过其债券评级进一步聚合。

■ 示例3.7

血库中储存的血液可以按血型、来源(可能表明疾病风险)、储存时长 (最多可储存42天)和当前储存地点进行分类。国家血液管理机构可能希望通过忽略来源(忽略维度是一种聚合形式)、将储存时长从几天离散到几周，并将地点聚合到聚合区域，从而聚合状态空间。

■ 示例3.8

资产的价值由其连续的当前价格决定。可以使用离散到最接近美元的价格来估计资产。

在许多应用中，聚合天然分层。例如，在漂泊的货车司机问题中，可能希望根据3个属性来估计货车的价值：位置、家庭住所和车队类型。前两个表示地理位置，可以用3个聚合级别表示(在本例中)：400个子区域、100个区域和10个地区。表3.1说明了可能使用的5个聚合级别。在此示例中，每个较高级别可以表示为先前级别的聚合。

表3.1 漂泊的货车司机问题的状态空间聚合示例(“-”表示忽略特定维度)

聚合级别	位置	车队类型	住所	状态空间大小
0	子区域	车队	区域	$400 \times 5 \times 100 = 200\,000$
1	区域	车队	区域	$100 \times 5 \times 100 = 50\,000$
2	区域	车队	地区	$100 \times 5 \times 10 = 5000$
3	区域	车队	-	$100 \times 5 \times 1 = 500$
4	地区	-	-	$10 \times 1 \times 1 = 10$

聚合对于连续变量也很有用。假设状态变量是持有的现金量，可能高达1000万美元。可能将状态空间离散化为100万美元、10万美元、1万美元、1000美元、100美元和10美元。这种离散化产生了一个自然的层次结构，因为在一个聚合级别上的10个段会自然地分组为下一个聚合级别中的一个段。

分层聚合是生成一系列估计的自然方法，但大多数情况下，没有理由假设结构是分层的。事实上，甚至可以使用重叠聚合(有时称为“软”聚合)，其中相同的状态s聚合为$\mathcal{S}^g$中的多个元素。例如，假设s表示连续空间中的坐标(x,y)，其已离散为点集$(x_i,y_i)_{i\in\mathcal{I}}$。进一步假设有一个距离指标$\rho((x,y),(x_i,y_i))$测量从任何点$(x,y)$到每个聚合点$(x_i,y_i)$，$i\in\mathcal{I}$的距离。可以在点$(x,y)$上观察，以便在每个$(x_i,y_i)$处更新估计，权重随$\rho((x,y),(x_i,y_i))$减小。

3.6.2 不同聚合水平的估计

假设要近似一个函数$f(x)$, $x\in\mathcal{X}$。应先定义一系列聚合函数：

$$G^g:\mathcal{X}\to\mathcal{X}^{(g)}$$

$\mathcal{X}^{(g)}$代表域$\mathcal{X}$的第g个聚合级别。设：

$\mathcal{G}=$对应于聚合级别的一组索引

本节假设有一个聚合函数G，将解聚状态$x\in\mathcal{X}=\mathcal{X}^{(0)}$映射到聚合空间$\mathcal{X}^{(g)}$。3.6.3节将设$g\in\mathcal{G}=\{0,1,2,...\}$，并同时处理所有级别的聚合。

开始聚合研究前，应先描述如何在解聚层面采样值x。为此，可假设有两个外生过程：在第n次迭代，第一个过程选择要采样的值(表示为x^n)，而第二个过程产生对处于如下状态的价值的观察：

$$\hat{f}^n(x^n)=f(x^n)+\varepsilon^n$$

稍后，将假设x^n是由某策略决定的，但目前，可以将其视为纯粹的外因。

需要描述函数估计中出现的误差。设：

$f_x^{(g)}$ = 对原始函数$f(x)$的第g个聚合的真实估计

假设$f^{(0)}(x) = f(x)$，这意味着第0级聚合是真函数。

设：

$\bar{f}_x^{(g,n)}$ = n次观察后，对处于第g个聚合级别的$f(x)$值的估计

在整个讨论中，变量上的横线意味着它是根据样本观察计算出来的。尖角意味着变量是一个外生观察。

当研究最解聚的层面($g = 0$)时，测量的状态s是观察到的状态$s = \hat{s}^n$。对于$g > 0$，$\bar{f}_x^{(g,n)}$的下标x指的是$G^g(x^n)$，或者$f(x)$在$x = x^n$时的第g个聚合水平。给出一个观察$(x^n, \hat{f}^n(x^n))$，我们将采用下式更新$f^{(g)}(x)$的估计值：

$$\bar{f}_x^{(g,n)} = (1 - \alpha_{x,n-1}^{(g)})\bar{f}_x^{(g,n-1)} + \alpha_{x,n-1}^{(g)}\hat{f}^n(x)$$

此处的步长$\alpha_{x,n-1}^{(g)}$明确表示对决策x以及聚合水平的依赖。这意味着这也是通过n次迭代更新$\bar{f}_x^{(g,n)}$的次数的函数，而非n自身的函数。

为了说明这一点，可假设漂泊的货车司机由向量x=(Loc, Equip, Home, DOThrs, Days)描述，其中，Loc是位置，Equip表示货车类型(长、短、冷藏)，Home是司机的住所位置，DOThrs是一个表示司机在过去8天中每天工作小时数的向量，Days是司机离家的天数。为x的不同聚合级别估计值$f(x)$，聚合时忽略s的特定维度。从最初的解聚观察$\hat{f}(x)$开始，将其写作：

$$\hat{f}\begin{pmatrix}\text{Loc}\\ \text{Equip}\\ \text{Home}\\ \text{DOThrs}\\ \text{Days}\end{pmatrix} = f(x) + \varepsilon$$

现在，希望使用属性为x的司机的估计以得到不同聚合级别的价值函数。可以通过简单地使用不同聚合级别的估计来使该解聚估计变得平滑，例如：

$$\bar{f}^{(1,n)}\begin{pmatrix}\text{Loc}\\ \text{Equip}\\ \text{Home}\end{pmatrix} = (1 - \alpha_{x,n-1}^{(1)})\bar{f}^{(1,n-1)}\begin{pmatrix}\text{Loc}\\ \text{Equip}\\ \text{Home}\end{pmatrix} + \alpha_{x,n-1}^{(1)}\hat{f}\begin{pmatrix}\text{Loc}\\ \text{Equip}\\ \text{Home}\\ \text{DOThrs}\\ \text{Days}\end{pmatrix}$$

$$\bar{f}^{(2,n)}\begin{pmatrix}\text{Loc}\\ \text{Equip}\end{pmatrix} = (1 - \alpha_{x,n-1}^{(2)})\bar{f}^{(2,n-1)}\begin{pmatrix}\text{Loc}\\ \text{Equip}\end{pmatrix} + \alpha_{x,n-1}^{(2)}\hat{f}\begin{pmatrix}\text{Loc}\\ \text{Equip}\\ \text{Home}\\ \text{DOThrs}\\ \text{Days}\end{pmatrix}$$

$$\bar{f}^{(3,n)}\begin{pmatrix}\text{Loc}\end{pmatrix} = (1 - \alpha_{x,n-1}^{(3)})\bar{f}^{(3,n-1)}\begin{pmatrix}\text{Loc}\end{pmatrix} + \alpha_{x,n-1}^{(3)}\hat{v}\begin{pmatrix}\text{Loc}\\ \text{Equip}\\ \text{Home}\\ \text{DOThrs}\\ \text{Days}\end{pmatrix}$$

第一个式子基于五维状态向量x对司机的值进行平滑处理，其近似值由三维状态向量索引。第二个式子使用由二维状态向量索引的价值函数近似进行相同的处理，第三个式子使用一维状态向量进行相同的处理。记住，步长必须反映状态更新的次数，这一点非常重要。

需要估计$\bar{f}_x^{(g,n)}$的方差。设：

$(s_x^2)^{(g,n)}$ =n次观察后，使用聚合级别g的数据对x处的函数观察值的方差估计

$(s_x^2)^{(g,n)}$是在$x = x^n$处观察到函数聚合到x(即$G^g(x^n) = x$)时，观察值$\hat{f}$方差的估计值。我们对平均值$\bar{f}_x^{(g,n)}$估计值的方差非常感兴趣。3.5节中有：

$$\begin{aligned}(\bar{\sigma}_x^2)^{(g,n)} &= Var[\bar{f}_x^{(g,n)}] \\ &= \lambda_x^{(g,n)}(s_x^2)^{(g,n)}\end{aligned} \tag{3.28}$$

其中，$(s_x^2)^{(g,n)}$是在第g个聚合水平对观察值$\hat{f}^n$方差的估计(计算如下)，$\lambda_s^{(g,n)}$可以通过递归计算：

$$\lambda_x^{(g,n)} = \begin{cases}(\alpha_{x,n-1}^{(g)})^2 & n = 1, \\ (1-\alpha_{x,n-1}^{(g)})^2\lambda_x^{(g,n-1)} + (\alpha_{x,n-1}^{(g)})^2 & n > 1\end{cases}$$

注意，如果步长$\alpha_{x,n-1}^{(g)}$变为零，则$\lambda_x^{(g,n)}$也会变为零，$(\bar{\sigma}_x^2)^{(g,n)}$亦是如此。现在需要计算$(s_x^2)^{(g,n)}$，这是观察值$\hat{f}^n$在点$x^n$处的方差的估计值，为此，$G^g(x^n) = x$(状态的观察值聚合到$x$)。设$\bar{v}_x^{(g,n)}$为总变化量，有：

$$\bar{v}_x^{(g,n)} = (1-\eta_{n-1})\bar{v}_x^{(g,n-1)} + \eta_{n-1}(\bar{f}_x^{(g,n-1)} - \hat{f}_x^n)^2$$

其中，η_{n-1}遵循一些步长规则(可能只是一个常数)。$\bar{v}_x^{(g,n)}$指总变化量，因为它获得了由于测量噪声(计算$\hat{f}^n(x)$时的随机性)和测量偏差(因为$\bar{f}_x^{(g,n-1)}$是对$\hat{f}^n(x)$平均值的有偏差估计)而产生的偏差。

最终需要计算下式以得出聚合偏差的估计值：

$$\bar{\beta}_x^{(g,n)} = \bar{f}_x^{(g,n)} - \bar{f}_x^{(0,n)} \tag{3.29}$$

可以使用下式分离出偏差的影响，以获得误差方差的估计值：

$$(s_x^2)^{(g,n)} = \frac{\bar{v}_x^{(g,n)} - (\bar{\beta}_x^{(g,n)})^2}{1+\lambda^{n-1}} \tag{3.30}$$

下一节将使用聚合偏差的估计值$\bar{\beta}_x^{(g,n)}$。

这些关系如图3.3所示。图3.3显示了在单个连续状态(例如资产价格)上定义的简单函数。如果选择一个特定的状态s，就会发现该状态只有两个观察值，而函数的那部分却有7个观察值。如果使用聚合近似，将在该函数范围内得到单一的数字，从而在真实函数和聚合估计之间产生偏差。如图3.3所示，偏差的大小取决于该区域中函数的形状。

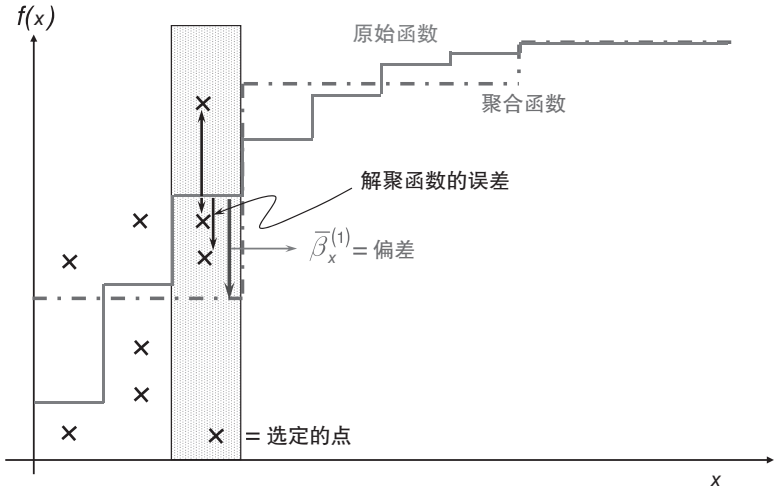

图3.3 解聚函数、聚合近似值和一组样本(为特定状态s显示估计值和偏差)

选择最优聚合级别的一种方法是选择最小化$(\bar{\sigma}_s^2)^{(g,n)} + (\bar{\beta}_s^{(g,n)})^2$的级别，它捕捉了偏差和方差。3.6.3节将使用偏差和方差来开发一种同时使用所有聚合级别的估计值的方法。

3.6.3 组合多个聚合级别

与其试图选择最优的聚合级别，不如直观地使用不同聚合级别的估计值的加权和。最简单的策略是使用下式：

$$\bar{f}_x^n = \sum_{g \in \mathcal{G}} w^{(g)} \bar{f}_x^{(g)} \tag{3.31}$$

其中，$w^{(g)}$是适用于第g个聚合级别的权重。我们希望权重是正的并且加起来是1，但也可以将它们视为回归函数中的系数。这种情况下，通常将回归写作：

$$\bar{F}(x|\theta) = \theta_0 + \sum_{g \in \mathcal{G}} \theta_g \bar{f}_x^{(g)}$$

关于线性模型的介绍参见3.7节。该策略的问题在于，权重不取决于x值。直觉上，应该对具有更多观察值或估计方差较低的点x赋予更高的权重。如果权重不取决于x，则不尽然。

在实践中，通常会更频繁地观察某些状态，以表明权重应该取决于x。要做到这一点，需要使用：

$$\bar{f}_x^n = \sum_{g \in \mathcal{G}} w_x^{(g)} \bar{f}_x^{(g,n)}$$

现在，权重取决于所估计的点，当进行大量观察时，可以对解聚估计赋予更高的权重。这显然是最自然而然的，但当域$\mathcal{X}$很大时，就会面临计算数千(甚至数十万)权重的挑

战。这种情况则需要一个相当简单的方法来计算权重。

可以将估计值$(\bar{f}^{(g,n)})_{g\in\mathcal{G}}$视为估计相同量的不同方法。关于这个问题，已有大量的统计文献。例如，众所周知，在式(3.31)中最小化$\bar{f}_x^n$的方差的权重为：

$$w_x^{(g)} \propto \left((\bar{\sigma}_x^2)^{(g,n)}\right)^{-1}$$

由于权重之和应为1，因此有：

$$w_x^{(g)} = \left(\frac{1}{(\bar{\sigma}_x^2)^{(g,n)}}\right)\left(\sum_{g\in\mathcal{G}}\frac{1}{(\bar{\sigma}_x^2)^{(g,n)}}\right)^{-1} \tag{3.32}$$

如果估计无偏，则这些权重有效，但事实显然并非如此。这很容易通过使用总变化(方差加上偏差的平方)来修正，从而产生权重：

$$w_x^{(g,n)} = \frac{1}{\left((\bar{\sigma}_x^2)^{(g,n)} + \left(\bar{\beta}_x^{(g,n)}\right)^2\right)}\left(\sum_{g'\in\mathcal{G}}\frac{1}{\left((\bar{\sigma}_x^2)^{(g',n)} + \left(\bar{\beta}_x^{(g',n)}\right)^2\right)}\right)^{-1} \tag{3.33}$$

这些权重是针对每个聚合级别$g \in \mathcal{G}$计算的。此外，为每个点x计算一组不同的权重。可以使用式(3.28)和式(3.29)递归计算$(\bar{\sigma}_x^2)^{(g,n)}$和$\bar{\beta}_x^{(g,n)}$，这使得该方法非常适合大规模应用。注意，如果用于平滑$\hat{f}^n$的步长为零，则方差$(\bar{\sigma}_x^2)^{(g,n)}$也将随$n\to\infty$变为零。然而，偏差$\bar{\beta}_x^{(g,n)}$通常不会变为零。

图3.4显示了对于特定应用，每个聚合级别的平均权重(当对所有输入x求平均时)。该行为说明了一个直观的特性，即当只有少量观察时，聚合级别上的权重最高，随着算法的进展，权重会转移到更解聚的级别。这是递归近似函数时非常重要的行为。仅仅用几个数据点是不可能产生好的函数近似的，因此有必要使用只有几个参数的简单函数。

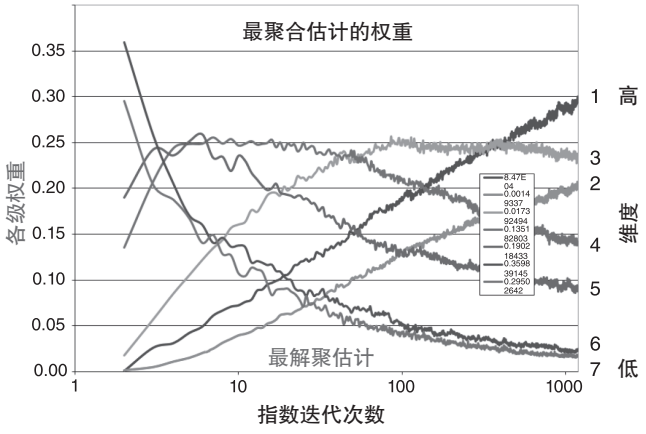

图3.4 使用式(3.33)计算的每个聚合级别的平均权重(所有状态)