计算机与智能科学丛书

在线凸优化

（第 2 版）

[美] 埃拉德 • 哈赞 (Elad Hazan) 著

罗俊仁　张万鹏　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：01-2024-0465
Elad Hazan
Introduction to Online Convex Optimization
EISBN: 9780262046985

图书在版编目(CIP)数据

在线凸优化：第2版 / (美) 埃拉德 • 哈赞(Elad Hazan) 著；罗俊仁，张万鹏译. —北京：清华大学出版社，2024.5
(计算机与智能科学丛书)
书名原文: Introduction to Online Convex Optimization
ISBN 978-7-302-66112-2

Ⅰ. ①在… Ⅱ. ①埃… ②罗… ③张… Ⅲ. ①凸分析—最优化算法 Ⅳ. ①O174.13

中国国家版本馆CIP数据核字(2024)第085114号

责任编辑：王 军
装帧设计：孔祥峰
责任校对：成凤进
责任印制：宋 林

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：艺通印刷（天津）有限公司
经 销：全国新华书店
开 本：148mm×210mm 印 张：7.375 字 数：242 千字
版 次：2024 年 6 月第 1 版 印 次：2024 年 6 月第 1 次印刷
定 价：99.80 元

产品编号：101424-01

译 者 序

近年来，随着机器学习和计算机技术的不断发展，以及在 Web 上收集大量数据的普及，在线广告优化、在线投资组合、在线博弈学习等应用已成为工业和学术界关注的热点。而在线凸优化是一种专门用于处理在线学习过程中凸优化问题的优化理论。我们非常高兴介绍由普林斯顿大学教授埃拉德•哈赞 (Elad Hazan) 撰写的《在线凸优化 (第 2 版)》，书中包括了在线凸优化的基本概念和方法，并详细讨论了在线凸优化的性能保证问题，还介绍了一些最新的在线凸优化算法。本书涵盖了在线凸优化领域的许多关键问题，为我们提供了一份全面的指南。

第 1、2 章简要介绍了在线凸优化的基础知识和基本概念。这些知识为后续章节的学习和理解奠定了坚实基础。前两章的内容包括凸函数与凸集的定义、在线学习的形式化表示以及在线凸优化的算法框架。第 3、4 章系统介绍了一阶算法和二阶算法这两类在线凸优化的方法。这两章涵盖了重要而广泛的算法，并讨论了随机梯度下降和在线牛顿步等不同算法的优缺点。这两章深入算法细节，对于对该领域有深入理解的读者尤其有帮助。第 5 章讨论了在线凸优化中的正则化技术。正则化技术是在线凸优化的核心技术之一，可以帮助我们提高模型的泛化能力。第 6 章详细介绍了经典的在线凸优化框架“赌博机凸优化”。这个框架在许多实际应用中非常有效，本章通过范例和解释，深入理解了赌博机凸优化的工作原理。第 7 章讲解了无投影算法。这些算法已在许多在线学习问题中得到了广泛应用，尤其是在大规模数据集上的在线学习问题中。第 8 章从博弈论的角度讲解在线凸优化理论。本章介绍了在线学习和博弈论之间的联系，并解释了在线凸优化如何解决这些问题。

第 9 章讲解了与在线凸优化有关的统计学习理论。这些理论可以帮助我们更好地理解在线凸优化问题和算法，以及它们如何与机器学习领域的其他问题相结合。第 10 章介绍了在现实多变的环境中涉及在线凸优化的实际应用问题。这些问题包括在线广告、在线内容推荐等。本章通过相应案例的讲解，帮助读者更好地理解在线凸优化在实际应用中的价值。第 11 章主要介绍了机器学习算法 Boosting 和在线凸优化算法的衡量指标“遗憾”，通过几个例子阐述这个话题的主要内容，涵盖了与“遗憾”相关的许多问题。第 12 章讲解了在线 Boosting 方法及其用途。本章涵盖了机器学习算法 Boosting 的基础知识和主要算法，在此基础上讲解了在线 Boosting 的原理和应用。最后，第 13 章介绍了 Blackwell 可接近性定理。

作为本书的译者，我深刻理解了在线凸优化的重要性。它不仅是解决实际问题的有力工具，而且在许多领域中都具有重要的理论支撑意义。在线凸优化能够帮助我们在在线学习过程中实时调整模型，以最大化预测准确性和泛化性能。同时，我们想向埃拉德•哈赞博士致以诚挚的敬意，他在在线凸优化领域作出了突出的贡献，为机器学习的发展奠定了坚实的基础。我们也要感谢清华大学出版社提供的翻译机会，让我们有机会将这本书带给中文读者。最后，希望这本书成为在线凸优化研究和实践的参考资料，为那些对在线学习和数据处理感兴趣的读者提供有价值的见解和帮助。

前　言

本书是在线凸优化 (Online Convex Optimization，OCO) 扩展理论的导论。它是为研究生基础课程编写的高等教材，可作为深入优化与机器学习交叉领域的研究人员的参考书。

Technion 于 2010—2014 年开设了这门课程，每年略有变化，普林斯顿大学于 2015—2020 年开设了这门课程。本书全面涵盖了这些课程的核心材料，并给出让学生完成部分证明的练习，或者参加课程的人觉得具有启发性和发人深省的练习。大部分材料都给出了应用实例，这些应用实例穿插在各个主题中，包括专家建议的预测、投资组合选择、矩阵补全和推荐系统，以及支持向量机 (Support Vector Machine，SVM) 的训练。

我们希望这份材料和练习纲要对研究人员和教育工作者有用。

把这本书放在机器学习图书馆中

机器学习的广阔领域，如在线学习 (online learning)、提升 (boosting)、博弈中的遗憾最小化 (regret minimization in games)、通用预测 (universal prediction) 和其他相关主题的子学科，近年来已经出现了大量的入门书籍。在此，我们很难对所有这些进行公正的评价，但也许可以列出与机器学习、博弈学习和优化主题最相关的书籍，它们的交集是我们的主要关注点。

最密切相关的书是 Cesa-Bianchi and Lugosi (2006)，它对整个博弈学习领域起到了启发作用。在数学优化理论的文献中，有许多关于凸优化和凸分析的介绍性文章，例如以下作者的文章：Boyd and

Vandenberghe, 2004; Nesterov, 2004; Nemirovski and Yudin, 1983; Nemirovski, 2004; Borwein and Lewis, 2006; Rockafellar, 1997。作者推荐了自己学习数学优化理论的书籍 (Nemirovski, 2004)。关于机器学习的书籍太多了，在这里无法一一列举。

本书的主要目的是作为 OCO 和机器学习凸优化方法专门课程的教科书。在线凸优化已经产生了足够的影响，出现在多个综述和导论文献中 (Hazan, 2011; Shalev-Shwartz, 2011; Rakhlin, 2009; Rakhlin and Sridharan, 2014)。我们希望这份材料和练习的汇编将进一步丰富这些文献。

本书结构

本书旨在作为计算机科学/电气工程/运筹学/统计学及相关领域研究生独立课程的参考。因此，它的组织遵循了在 Technion 教授的“决策分析”课程的结构，以及后来在普林斯顿大学教授的“理论机器学习”课程的结构。

每章应该花费一到两周的课时，具体取决于课程的深度和广度。第 1 章为该领域的导论，不像本书其他部分那么严谨。

粗略地说，本书可以分成三个部分。第一部分是第 2 章到第 4 章，包含了在线凸优化的基本定义、框架和核心算法。第二部分是第 5 章到第 7 章，包含更高阶的算法、框架的深入分析以及其他计算和信息访问模型的扩展。本书的其余部分 (即第三部分) 涉及更高级的算法、更困难的设置以及与知名机器学习范式的关系。

本书可以帮助教育工作者设计关于在线凸优化主题的完整课程，也可以作为机器学习综合课程的组成部分。

第2版新增

本书第 2 版的主要增补内容如下：

- 在第 2 章中扩大了优化范围，对 Polyak 步长进行了统一的梯度下降分析。
- 在第 9 章中扩展了学习理论的涵盖范围，介绍了压缩及其在泛化理论中的应用。
- 扩展的第 4 章，增加了指数 (exp) 凹损失函数的指数加权优化器。
- 修订后的第 5 章，增加了镜像下降分析，自适应梯度方法 (5.6 节) 也进行了修订。
- 新的第 10 章包括自适应遗憾的概念和面向具有近似最优自适应遗憾界的 OCO 算法。
- 新的第 11 章包括 Boosting(提升)及其与在线凸优化的关系。从遗憾最小化推导 Boosting 算法。
- 新的第 12 章包括在线 Boosting。
- 新的第 13 章包括 Blackwell 可接近性及其与在线凸优化的紧密联系。

致　　谢

第 1 版

首先，非常感谢 2010—2014 年在 Technion 学习“决策分析”课程的学生以及 2015—2016 年在普林斯顿大学学习“机器学习理论”的学生所作出的众多贡献和给出的见解。

我要感谢朋友、同事和学生，他们提出了许多建议和指正。部分名单包括：Sanjeev Arora、Shai Shalev-Shwartz、Aleksander Madry、Yoram Singer、Satyen Kale、Alon Gonen、Roi Livni、Gal Lavee、Maayan Harel、Daniel Khasabi、Shuang Liu、Jason Altschuler、Haipeng Luo、Zeyuan Allen-Zhu、Mehrdad Mahdavi、Jaehyun Park、Baris Ungun、Maria Gregori、Tengyu Ma、Kayla McCue、Esther Rolf、Jeremy Cohen、Daniel Suo、Lydia Liu、Fermi Ma、Mert Al、Amir Reza Asadi、Carl Gabel、Nati Srebro、Abbas Mehrabian、Chris Liaw、Nikhil Bansal、Naman Agarwal、Raunak Kumar、Zhize Li、Sheng Zhang、Swati Gupta、Xinyi Chen、Liang Zeng 和 Kunal Mittal。

感谢 Udi Aharoni 为这本书的算法进行了艺术创作并绘制插图。

永远感激我的老师和导师 Sanjeev Arora，如果没有他，这本书不可能完成。

最后，感谢我的妻子和孩子们的爱和支持：Dana、Hadar、Yoav 和 Oded。

埃拉德•哈赞

普林斯顿大学

第2版

非常感谢与我合作进行研究的学生和同事，其中一些研究出现在本书的第 2 版中。尤其要感谢在 Boosting 方法方面与我合作的 Nataly Brukhim、Xinyi Chen、Shay Moran、Naman Agarwal 和 Karan Singh。

感谢我的学生帮助我校对新的和已有的部分，他们是 Edgar Minasyan、Paula Gradu、Karan Singh、Nataly Brukhim、Xinyi Chen、Naman Agarwal 和 Udaya Ghai。

感谢 Shay Moran 解释了压缩方案以及它们如何简化 Boosting 的泛化性。

非常感谢 Ahmed Farah、Charlie Cowen-Breen 和“COS 597C: 计算控制理论”的学生对许多问题集给出的有益建议、更正和解决方案。

非常感谢 Wouter Koolen-Wijkstra 在分析在线牛顿步 (Online Newton Step) 算法时提出的有益建议。

感谢 MIT 出版社为这本书找到的非常有帮助和严谨的审稿人，他们给出了极好的建议，并改进了最终的手稿。

和第 1 版一样，更重要的是，感谢我的妻子和孩子们：Dana、Hadar、Yoav 和 Oded，感谢他们的爱和支持。

埃拉德•哈赞

普林斯顿大学

符号列表

通用

$\stackrel{\text{def}}{=}$	定义
arg min{}	最小化括号中表达式时参数取值
$[n]$	整数集合 {1,2,...,n}

几何与微积分

$\mathbb{R}^d$	d 维 Euclidean 空间 (欧几里得空间，欧氏空间)
Δ_d	d 维单纯形，$\{\sum_i \mathbf{x}_i = 1, \mathbf{x}_i \geq 0\}$
$\mathbb{S}$	d 维球体，$\{\|\mathbf{x}\| = 1\}$
$\mathbb{B}$	d 维球，$\{\|\mathbf{x}\| \leq 1\}$
$\mathbb{R}$	实数
$\mathbb{C}$	复数
$\|A\|$	矩阵 A 的行列式

学习理论

$\mathcal{X}$,$\mathcal{Y}$	特征 / 标签集
$\mathcal{D}$	样本 ($\mathbf{x}$, y) 的分布
$\mathcal{H}$	$\mathcal{X} \mapsto Y$ 中的假设类
h	单一假设$h \in \mathcal{H}$
m	训练集大小
error(h)	假设$h \in \mathcal{H}$的泛化误差

优化

$\mathbf{x}$	决策集中的向量
$\mathcal{K}$	决策集
$\nabla^k f$	f 的 k 次微分；记作 $\nabla^k f \in \mathbb{R}^{d^k}$
$\nabla^{-2} f$	f 的逆 Hessian 矩阵
∇f	f 的梯度
∇_t	f 在点 $\mathbf{x}_t$ 处的梯度
$\mathbf{x}^\star$	目标 f 的全局或局部最优
h_t	目标值与最优值的距离，$h_t = f(\mathbf{x}_t) - f(\mathbf{x}^\star)$
d_t	与最优值点的 Euclidean 距离 $d_t = \|\mathbf{x}_t - \mathbf{x}^\star\|$
G	次梯度范数的上界
D	Euclidean 直径的上界
D_p, G_p	次梯度/直径的 p 范数上界

正则化

R	强凸光滑正则化函数
$B_R(\mathbf{x}\|\mathbf{y})$	R 的 Bregman 散度 $R(\mathbf{x}) - R(\mathbf{y}) - \nabla R(\mathbf{y})^\top (\mathbf{x} - \mathbf{y})$
G_R	(次) 梯度范数的上界
D_R^2	平方 R 直径 $\max_{\mathbf{x},\mathbf{y}\in\mathcal{K}} \{R(\mathbf{x}) - R(\mathbf{y})\}$
$\|\mathbf{x}\|_A^2$	平方矩阵范数 $\mathbf{x}^\top A \mathbf{x}$
$\|\mathbf{x}\|_{\mathbf{y}}^2$	局部正则化 $\mathbf{x}^\top \nabla^2 R(\mathbf{y}) \mathbf{x}$ 的局部范数
$\|\mathbf{x}\|^*$	$\|\mathbf{x}\|$ 的对偶范数

目　录

第1章

导　论

本书将优化 (optimization) 看作一个过程 (process)。在许多实际应用中，环境太过复杂，以至于无法建立一个全面的理论模型并应用经典的算法理论和数学优化。随着问题的更多方面被观察到，通过应用所学的优化方法来采取稳健的方法，是必要的，也是有益处的。这种将优化视为一个过程的观点在各个领域都很重要，在建模和部分现代日常生活中取得了巨大成功。

机器学习、统计学、决策科学和数学优化的文献越来越多，模糊了经典意义下确定性建模、随机建模和优化方法之间的区别。本书延续了这一趋势，研究了一个突出的优化框架：在线凸优化 (Online Convex Optimization，OCO) 框架，该框架在数学科学中的确切位置尚不清楚，它最先在机器学习的文献中定义 (参考 1.4 节)。该框架成功的度量标准借鉴了博弈论，与统计学习理论和凸优化密切相关。

我们接受这些富有成效的联系，并且有意在讨论中不使用任何特定的术语。相反，本书将从可以通过 OCO 建模和解决的实际问题开始。我们将继续介绍严格的定义、背景和算法，并始终提供与其他领域文献的联系。我们希望作为读者的你，能从你的专业领域为我们对这些联系的理解作出贡献，并扩展有关这一迷人主题的文献数量。

1.1 在线凸优化设置

在 OCO 中，在线玩家迭代式地作出决策。在作出每一个决策时，玩家都不知道与之相关的结果。

在作出决策后，决策者会遭受损失：每一个可能的决定都会带来 (可能不同的) 损失。决策者事先不知道这些损失。损失可能是对手选择的，甚至取决于决策者采取的行动。

在这一点上，为了使这个框架有意义，有必要给出一些限制。

- 不应允许对手确定的损失是无界的[1]。否则，对手可以在每一步都不断减小损失的规模，并且永远不允许算法从第一步的损失中恢复。因此，假设损失位于某个有界区域。
- 决策集必须在某种程度上是有界的和 / 或结构化的，尽管不一定是有限的。

  要了解为什么这是必要的，请在作出决策时考虑使用无限组可能的决策。对手可以无限期地为玩家选择的所有策略分配高损失，同时将一些策略设置为零损失。这排除了任何有意义的性能指标。

令人惊讶的是，有趣的语句和算法可以在不超过这两个限制的情况下推导。OCO 框架将决策集建模为 Euclidean(欧几里得) 空间中的凸集，表示为 $\mathcal{K} \subseteq \mathbb{R}^n$。代价被建模为 $\mathcal{K}$ 上的有界凸函数。

OCO 框架可以看作一个结构化的重复博弈。该学习框架的协议如下。

在迭代 t，在线玩家选择 $\mathbf{x}_t \in \mathcal{K}$。在玩家作出这一选择后，揭示了凸代价函数 $f_t \in \mathcal{F} : \mathcal{K} \mapsto \mathbb{R}$。这里，$\mathcal{F}$ 是对手可用的有界代价函数族。在线玩家产生的代价为 $f_t(\mathbf{x}_t)$，即选择 $\mathbf{x}_t$ 的代价函数的值。用 T 表示博弈迭代的总数。

是什么使一个算法成为一个好的 OCO 算法？由于该框架天然具有

1 或者，同样地，此“设置”(设定) 的任何性能指标都应取决于最大损失的大小。这是本书后面采用的观点。

博弈性和对抗性，因此合理的性能指标也来自博弈论：将决策者的遗憾 (Regret) 定义为所产生的总代价与事后最佳固定决策的总代价之间的差额。在 OCO 中，通常对算法的最坏遗憾情况的上限感兴趣。

令 $\mathcal{A}$ 为 OCO 算法，它将某个特定博弈中的历史决策映射到决策集中：

$$\mathbf{x}_t^{\mathcal{A}} = \mathcal{A}(f_1, ..., f_{t-1}) \in \mathcal{K}$$

将 T 次迭代后 $\mathcal{A}$ 的遗憾形式化定义为：

$$\text{Regret}_T(\mathcal{A}) = \sup_{\{f_1,...,f_T\}\subseteq\mathcal{F}} \left\{ \sum_{t=1}^{T} f_t(\mathbf{x}_t^{\mathcal{A}}) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}) \right\} \quad (1.1)$$

如果算法在语境中是清楚的，那么将省略上标，并将算法在时间 t 的决策简单地表示为 $\mathbf{x}_t$。直观地讲，如果遗憾是 T 的次线性函数 [即 $\text{Regret}_T(\mathcal{A})=o(T)$]，则算法表现良好，因为这意味着平均而言，该算法的表现与事后来看的最佳固定策略一样好。

对于 T 迭代重复博弈中的迭代 $t\in[T]$[1]，OCO 算法的运行时间被定义为产生 $\mathbf{x}_t$ 的最坏预期时间。通常，运行时间将取决于 n(决策集 $\mathcal{K}$ 的维度)、T(博弈迭代的总次数) 以及代价函数和基本凸集的参数。

1.2　可用OCO建模的问题示例

近年来，OCO 成为领先的在线学习框架的主要原因可能是其强大的建模能力：来自不同领域的问题，如在线路由、搜索引擎的广告选择和垃圾邮件过滤，都可以建模为特例。本节简要介绍一些特例，以及它们如何融入 OCO 框架。

1.2.1　从专家建议中预测

也许预测理论中最广为人知的问题是专家问题 (experts problem)。决策者必须在几位专家的建议中作出选择。在作出选择后，产生了介于

1　在这里和以后，将整数集 $\{1,...,n\}$ 表示为 $[n]$。

0 到 1 之间的损失。这种情况是反复重复的，在每次迭代中，各种专家的代价是任意的 (甚至可能是对抗性的，试图误导决策者)。决策者的目标是在事后做得和最好的专家一样好。

OCO 设置将此作为一个特例：决策集是 n 个元素上所有分布的集合 (专家)；即 n 维单纯形 $\mathcal{K} = \Delta_n = \{\mathbf{x} \in \mathbb{R}^n \ , \ \sum_i \mathbf{x}_i = 1 \ , \ \mathbf{x}_i \geq 0\}$。设迭代 t 中，第 i 个专家的代价为 $\mathbf{g}_t(i)$，并设 $\mathbf{g}_t$ 为所有 n 个专家的代价向量。则代价函数是根据分布 $\mathbf{x}$ 选择专家的预期代价，由线性函数 $f_t(\mathbf{x}) = \mathbf{g}_t^\top \mathbf{x}$ 给出。

因此，根据专家建议进行预测是 OCO 的一种特例，其中决策集是单纯形，代价函数是线性的，并且在 ℓ_∞ 范数下有界，至多为 1。代价函数的界是从代价向量 $\mathbf{g}_t$ 的元素上的界推导的。

专家问题在机器学习中的根本重要性值得特别关注，本章结束时将回到这一问题并详细分析。

1.2.2 在线垃圾邮件过滤

考虑一个在线垃圾邮件过滤系统。电子邮件重复到达系统并被分类为垃圾邮件或有效邮件。显然，这样的系统必须应对对抗生成的数据，并随着输入的变化而动态变化——这是 OCO 模型的标志。

该模型的线性变体是利用“单词袋”将电子邮件表示为向量来捕获的。每封电子邮件都表示为向量 $\mathbf{a} \in \mathbb{R}^d$，其中 d 是字典中的单词数。这个向量中的其他条目都是 0，只有那些与电子邮件中出现的单词相对应的坐标被赋予值 1。

为了预测电子邮件是否是垃圾邮件，学习一个过滤器，例如向量 $\mathbf{x} \in \mathbb{R}^d$。通常，这个向量的 Euclidean 范数上的界是由先验决定的，并且在实践中是一个非常重要的参数。

过滤器 $\mathbf{x} \in \mathbb{R}^d$ 对电子邮件 $\mathbf{a} \in \mathbb{R}^d$ 的分类是由这两个向量之间的内积的符号给出的，即 $\hat{b} = \text{sign}(\mathbf{x}^\top \mathbf{a})$(例如，+1 表示有效邮件，−1 表示垃圾邮件)。

在在线垃圾邮件过滤的 OCO 模型中，决策集被认为是所有这些范数有界线性过滤器的集合，即一定半径的 Euclidean 球。代价函数是根据到达系统的传入电子邮件流及其标签 (可能是系统已知的、部分已知的或根本不知道的) 确定的。设 $(\mathbf{a},b)$ 是一个电子邮件 / 标签对。然后过滤器上相应的代价函数由$f(\mathbf{x}) = \ell(\hat{b},b)$给出。这里 $\hat{b}$ 是由过滤器 $\mathbf{x}$ 给出的分类，b 是真标签，ℓ 是凸损失函数，例如，缩放平方损失$\ell(\hat{b},b) = \frac{1}{4}(\hat{b}-b)^2$。

在这一点上，读者可能会想：为什么要使用平方损失而不是任何其他函数？最自然的选择可能是，如果 $b=\hat{b}$，则损失为 1，否则为 0。

要回答这个问题，首先要注意，如果 b 和 $\hat{b}$ 都是二进制的，并且以 $\{-1,1\}$ 为单位，那么平方损失实际上是 1 或 0。然而，使用连续函数使我们在决策过程中具有更大的灵活性。例如，可以允许算法根据其置信度返回区间 $[-1,1]$ 中的数字。

另一个原因与找到一个好解的算法效率有关。后面会介绍。

1.2.3　在线最短路径

在在线最短路径问题中，给决策者一个有向图 $G=(V,E)$ 和 $u,v\in\mathrm{V}$ 的源-宿对。在每次迭代 $t\in[T]$ 中，决策者选择路径$p_t \in \mathcal{P}_{u,v}$，其中$\mathcal{P}_{u,v} \subseteq E^{|V|}$是图中所有 u-v 路径的集合。对手独立选择图的边上的权重 (长度)，该权重由从边到实数的函数$\mathbf{w}_t : E \mapsto \mathbb{R}$给出，该函数可以表示为向量$\mathbf{w}_t \in \mathbb{R}^m$，其中$m = |E|$。决策者遭受并观察到损失，该损失是所选路径$\sum_{e\in p_t} \mathbf{w}_t(e)$的加权长度。

将这个问题离散地描述为专家问题，即每条路径都有一个专家，这对效率提出了挑战。就图形表示的大小而言，可能存在许多呈指数级增长的路径。

此外，在线最短路径问题可以在如下的 OCO 框架中求解。回想一下图中路径 (流) 上所有分布的集合的标准描述，即$\mathbb{R}^m$中的凸集，具有$O(m+|V|)$约束 (见图 1.1)。用 $\mathcal{K}$ 表示这个流多面体 (flow polytope)。给定流 $\mathbf{x}\in\mathcal{K}$(路径上的分布) 的预期代价是线性函数，由$f_t(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$给

出，其中，$\mathbf{w}_t(e)$是边 $e\in E$ 的长度。这种固有的简洁公式可以得到计算效率高的算法。

$$\sum_{e=(u,w),w\in V} \mathbf{x}_e = 1 = \sum_{e=(w,v),w\in V} \mathbf{x}_e \qquad \text{流量加和为 } 1$$

$$\forall w \in V \setminus \{u,v\} \quad \sum_{e=(v,x)\in E} \mathbf{x}_e = \sum_{e=(x,v)\in E} \mathbf{x}_e \qquad \text{流量守恒}$$

$$\forall e \in E \quad 0 \le \mathbf{x}_e \le 1 \qquad \text{容量约束}$$

图 1.1 定义流多面体的线性等式和不等式，是所有 u-v 路径的凸胞体

1.2.4 投资组合选择

在本节中，考虑一个投资组合选择模型，该模型不对股票市场做任何统计假设 (与股票价格的标准几何 Brownian 运动模型相反)，称为通用投资组合选择 (universal portfolio selection) 模型。

在每次迭代 $t\in[T]$ 时，决策者选择财富在 n 种资产 $\mathbf{x}_t \in \Delta_n$ 上的分配。对手独立选择资产的市场回报，即所有元素为正的向量 $\mathbf{r}_t \in \mathbb{R}^n$，使得每个坐标 $\mathbf{r}_t(i)$ 是迭代 t 和 $t+1$ 之间第 i 个资产的价格比。投资者在 $t+1$ 和 t 迭代时的财富之比为 $\mathbf{r}_t^\top \mathbf{x}_t$，因此此设置中的收益被定义为财富变化比率的对数 $\log(\mathbf{r}_t^\top \mathbf{x}_t)$。注意，由于 $\mathbf{x}_t$ 是投资者财富的分配，即使 $\mathbf{x}_{t+1}=\mathbf{x}_t$，投资者可能仍需要进行交易以调整价格变化。

遗憾最小化的目标，在这种情况下对应最小化差值 $\max_{\mathbf{x}^\star\in\Delta_n} \sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}^\star) - \sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}_t)$，具有直观的解释。第一项是事后最佳分布 $\mathbf{x}^\star$ 所累积财富的对数。由于该分布是固定的，因此它对应于在每个交易期后重新平衡头寸的策略，因此被称为恒定再平衡投资组合 (constant rebalanced portfolio)。第二项是在线决策者积累的财富的对数。因此，遗憾最小化对应于最大化投资者的财富与投资策略库中最佳基准的财富之比。

一种通用投资组合选择算法 (universal portfolio selection algorithm) 被定义为：在这种情况下，遗憾收敛到 0。这种算法虽然需要指数时间，但首先由 Cover 描述 (见 1.4 节)。OCO 框架已经给出了

基于牛顿方法的更有效的算法，第 4 章将详细研究这些方法。

1.2.5　矩阵补全和推荐系统

大规模媒体交付系统的流行，如 Netfix 在线视频库、Spotify 音乐服务和许多其他系统，产生了超大规模的推荐系统。矩阵补全模型是自动推荐最流行和最成功的模型之一。在这个数学模型中，推荐被认为是组成一个矩阵。用户表示为行向量，不同的媒体是各列，在与特定用户/媒体对相对应的元素位置处有一个值，表示用户对该特定媒体的偏好评分。

例如，对于音乐的二进制推荐，有一个矩阵$X \in \{0,1\}^{n\times m}$，其中$n$是考虑的人数，$m$是库中的歌曲数量，0/1 分别表示不喜欢/喜欢：

$$X_{ij} = \begin{cases} 0, & \text{用户 } i \text{ 不喜欢歌曲 } j \\ 1, & \text{用户 } i \text{ 喜欢歌曲 } j \end{cases}$$

在在线设置中，对于每次迭代，决策者输出一个偏好矩阵 $X_t \in \mathcal{K}$，其中$\mathcal{K} \subseteq \{0,1\}^{n\times m}$是所有可能的 0/1 矩阵的子集。然后，对手选择用户 / 歌曲对 (i_t, j_t) 以及该对$y_t \in \{0,1\}$的“真实”偏好。因此决策者所遭受的损失可以用凸损失函数描述。

$$f_t(X) = (X_{i_t,j_t} - y_t)^2$$

在这种情况下，自然的比较器是一个低秩矩阵，这对应于偏好由少数未知因素决定的直观假设。相对于该比较器，遗憾值意味着平均而言，执行的偏好预测误差与最佳低秩矩阵一样少。

第 7 章将回到这个问题，并探讨它的有效算法。

1.3　混合的开始：从专家建议中学习

考虑以下基本的迭代决策问题。

在每个时间步长 t=1,2,...,T，决策者面临两个动作 A 或 B 之间的

选择(即购买或出售特定股票)。决策者得到了 N 个“专家”的帮助，这些“专家”提供建议。在两个动作之间选择后，决策者接收与每个决策相关联的损失形式的反馈。为了简单起见，其中一个动作的损失为0(即“正确”决策)，另一个动作损失为1。

下面进行初步观察。

(1) 决策者在每次迭代中随机一致地选择一个动作，通常会损失 $\frac{T}{2}$，并且在50%的时间内是“正确的”。

(2) 就错误数量而言，在最坏的情况下，没有任何算法能做得更好！在后面的练习中，将设计一个随机设置，其中任何算法的预期错误数至少为 $\frac{T}{2}$。

因此，有动机考虑一个相对性能指标(relative performance metric)：事后看来，决策者能像最好的专家一样少犯错误吗？定理1.1表明，对于确定性决策者来说，在最坏的情况下，答案是否定的。

定理 1.1 *令 $L \leq \frac{T}{2}$ 表示最好的专家事后所犯错误的数量。那么，不存在一种可以保证错误少于 $2L$ 的确定性算法。*

证明 假设只有两位专家，其中一位总是选择 A 选项，而另一位总是选择 B 选项。考虑对手总是选择与我们预测相反方向的设置(可以这样做，因为我们的算法是确定的)。那么，该算法所犯的错误总数就是 T。然而，最好的专家犯的错误不超过 $\frac{T}{2}$(在每次迭代中，两个专家中只有一个错误)。因此，没有一种算法能够始终保证错误少于 $2L$。

这一观察结果推动了随机决策算法的设计，事实上，OCO框架在连续概率空间上优雅地对决策建模。因此可证明定理1.2，表述如下。

定理 1.2 *设 $\varepsilon \in (0, \frac{1}{2})$。假设最好的专家犯了 L 次错误。则：*

(1) *存在一种有效的确定性算法，可以保证少于 $2(1+\varepsilon)L + \frac{2\log N}{\varepsilon}$ 次错误；*

(2) *有一种有效的随机算法，其预期错误数最多为 $(1+\varepsilon)L + \frac{\log N}{\varepsilon}$。*

1.3.1　加权多数算法

加权多数 (Weighted Majority，WM) 算法可以直观地描述：每位专家 i 在每个迭代 t 处被分配权重 $W_t(i)$。最初，为所有专家 $i\in[N]$ 设置$W_1(i)=1$。对于所有 $t\in[T]$，设$S_t(A), S_t(B)\subseteq[N]$是在时间 t 选择 A(和选择 B) 的专家集。定义

$$W_t(A)=\sum_{i\in S_t(A)}W_t(i)\qquad\qquad W_t(B)=\sum_{i\in S_t(B)}W_t(i)$$

根据下式预测

$$a_t=\begin{cases}A & \text{如果 } W_t(A)\geq W_t(B)\\ B & \text{其他}\end{cases}$$

接下来，按照如下方式更新权重 $W_t(i)$：

$$W_{t+1}(i)=\begin{cases}W_t(i) & \text{如果专家 } i \text{ 是对的}\\ W_t(i)(1-\varepsilon) & \text{如果专家 } i \text{ 是错的}\end{cases}$$

其中 ε 是将影响其性能的算法的参数。这就结束了对加权多数算法的描述。下面研究该算法犯错的界。

引理 1.1　*用 M_t 表示算法到时间 t 为止所犯的错误数，用 $M_t(i)$ 表示专家 i 到时间 t 为止所犯的错误数。那么，对于任何专家 $i\in[N]$，有*

$$M_T\leq 2(1+\varepsilon)M_T(i)+\frac{2\log N}{\varepsilon}$$

可以优化 ε 以最小化上述界。右侧的表达式的形式为$f(x)=ax+b/x$，在$x=\sqrt{b/a}$ 处达到其最小值。因此，界在$\varepsilon^\star=\sqrt{\log N/M_T(i)}$ 处被最小化。利用这一最佳 ε 值，可知对于最好的专家 $i^\star$，有

$$M_T\leq 2M_T(i^\star)+O\left(\sqrt{M_T(i^\star)\log N}\right)$$

当然，$\varepsilon^\star$ 的值不能提前使用，因为不知道哪位专家是最好的专家[因此不知道 $M_T(i^\star)$ 的值]。然而，稍后将看到，即使没有这些先验知识，也可以获得相同的渐近界。

现在证明引理 1.1。

证明 对于所有 $t\in[T]$，令 $\Phi_t=\sum_{i=1}^{N}W_t(i)$，且注意 $\Phi_1=N$。

注意，$\Phi_{t+1}\leq\Phi_t$。然而，对于加权多数算法错误的迭代，有

$$\Phi_{t+1}\leq\Phi_t(1-\frac{\varepsilon}{2})$$

原因是总专家权重中至少有一半的专家错了（不然加权多数算法不会出错），因此

$$\Phi_{t+1}\leq\frac{1}{2}\Phi_t(1-\varepsilon)+\frac{1}{2}\Phi_t=\Phi_t(1-\frac{\varepsilon}{2})$$

根据所有观察，有

$$\Phi_t\leq\Phi_1(1-\frac{\varepsilon}{2})^{M_t}=N(1-\frac{\varepsilon}{2})^{M_t}$$

另一方面，根据任一专家 i 的定义，有

$$W_T(i)=(1-\varepsilon)^{M_T(i)}$$

由于 $W_T(i)$ 的值总是小于所有权重的和 Φ_T，可得

$$(1-\varepsilon)^{M_T(i)}=W_T(i)\leq\Phi_T\leq N(1-\frac{\varepsilon}{2})^{M_T}$$

两边取对数，可得

$$M_T(i)\log(1-\varepsilon)\leq\log N+M_T\log(1-\frac{\varepsilon}{2})$$

接下来，使用近似

$$-x-x^2\leq\log(1-x)\leq-x\qquad 0<x<\frac{1}{2}$$

根据对数函数的泰勒级数，可得

$$-M_T(i)(\varepsilon+\varepsilon^2)\leq\log N-M_T\frac{\varepsilon}{2}$$

引理证毕。

1.3.2 随机加权多数

在加权多数算法的随机版本 (记作 RWM) 中，时刻 t 时选择专家 i

的概率为$p_t(i)=W_t(i)/\sum_{j=1}^{N}W_t(j)$。

引理 1.2　令 M_t 表示直至迭代 t，RWM 算法的错误次数，则对任意专家 $i\in[N]$，有

$$\mathbf{E}[M_T]\leq(1+\varepsilon)M_T(i)+\frac{\log N}{\varepsilon}$$

这个引理的证明与引理 1.1 的证明非常相似，由于使用了随机性，常数因子 2 省略了。

证明　如引理 1.1，对于所有 $t\in[T]$，令$\Phi_t=\sum_{i=1}^{N}W_t(i)$，并记$\Phi_1=N$。令$\tilde{m}_t=M_t-M_{t-1}$为指示变量，若 RWM 算法在迭代 t 时出错，则指示变量为 1。如果专家 i 在迭代 t 犯错，则令 $m_t(i)$ 等于 1，否则为 0。考查权重和，得

$$\begin{aligned}\Phi_{t+1}&=\sum_i W_t(i)(1-\varepsilon m_t(i))\\&=\Phi_t(1-\varepsilon\sum_i p_t(i)m_t(i)) && p_t(i)=\tfrac{W_t(i)}{\sum_j W_t(j)}\\&=\Phi_t(1-\varepsilon\,\mathbf{E}[\tilde{m}_t])\\&\leq\Phi_t e^{-\varepsilon\,\mathbf{E}[\tilde{m}_t]} && 1+x\leq e^x\end{aligned}$$

另一方面，根据定义，对于任一专家 i 有

$$W_T(i)=(1-\varepsilon)^{M_T(i)}$$

由于 $W_T(i)$ 的值总是小于所有权重的和Φ_T，可得

$$(1-\varepsilon)^{M_T(i)}=W_T(i)\leq\Phi_T\leq Ne^{-\varepsilon\,\mathbf{E}[M_T]}$$

两边取对数，可得

$$M_T(i)\log(1-\varepsilon)\leq\log N-\varepsilon\,\mathbf{E}[M_T]$$

接下来，使用近似

$$-x-x^2\leq\log(1-x)\leq-x,\ \ 0<x<\frac{1}{2}$$

可得

$$-M_T(i)(\varepsilon+\varepsilon^2) \le \log N - \varepsilon\,\mathbf{E}[M_T]$$

引理证毕。

1.3.3 Hedge

RWM 实际上更为通用：可以考虑通过非负实数$\ell_t(i)$度量专家的性能，而不是考虑离散的错误数量，我们将$\ell_t(i)$称为专家 i 在迭代 t 时的损失。随机加权多数算法保证了决策者遵循其建议后，平均期望损失将接近事后最好的专家的平均期望损失。

从历史上看，这是由一种名为 Hedge 的不同且密切相关的算法观察到的 (见算法 1.1)，其总损失界将在本书稍后介绍。

算法 1.1 Hedge

1：初始化：$\forall i \in [N],\ W_1(i)=1$

2：**for** t=1 至 T **do**

3：　选择$i_t \sim_R W_t$，即 i_t=i 的概率为$\mathbf{x}_t(i) = \frac{W_t(i)}{\sum_j W_t(j)}$

4：　计算损失$\ell_t(i_t)$

5：　更新权重$W_{t+1}(i) = W_t(i)e^{-\varepsilon\ell_t(i)}$

6：**end for**

由此，用矢量符号表示算法的预期损失

$$\mathbf{E}[\ell_t(i_t)] = \sum_{i=1}^{N} \mathbf{x}_t(i)\ell_t(i) = \mathbf{x}_t^\top \ell_t$$

定理 1.3 令ℓ_t^2表示平方损失的 N 维向量，即$\ell_t^2(i) = \ell_t(i)^2$，令$\varepsilon > 0$，并假设所有的损失都是非负的。Hedge 算法对于任意专家$i^\star \in [N]$，满足:

$$\sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t \le \sum_{t=1}^{T} \ell_t(i^\star) + \varepsilon \sum_{t=1}^{T} \mathbf{x}_t^\top \ell_t^2 + \frac{\log N}{\varepsilon}$$

证明　如前，对于所有的 $t\in[T]$，令 $\Phi_t=\sum_{i=1}^{N}W_t(i)$，并记 $\Phi_1=N$。

考查权重的和：

$$\begin{aligned}
\Phi_{t+1}&=\sum_i W_t(i)e^{-\varepsilon\ell_t(i)}\\
&=\Phi_t\sum_i \mathbf{x}_t(i)e^{-\varepsilon\ell_t(i)} && \mathbf{x}_t(i)=\tfrac{W_t(i)}{\sum_j W_t(j)}\\
&\le\Phi_t\sum_i \mathbf{x}_t(i)(1-\varepsilon\ell_t(i)+\varepsilon^2\ell_t(i)^2)) && \text{对于 } x\ge 0,\ e^{-x}\le 1-x+x^2\\
&=\Phi_t(1-\varepsilon\mathbf{x}_t^\top\ell_t+\varepsilon^2\mathbf{x}_t^\top\ell_t^2)\\
&\le\Phi_t e^{-\varepsilon\mathbf{x}_t^\top\ell_t+\varepsilon^2\mathbf{x}_t^\top\ell_t^2} && 1+x\le e^x
\end{aligned}$$

另一方面，根据定义，对于专家 $i^\star$，有

$$W_{T+1}(i^\star)=e^{-\varepsilon\sum_{t=1}^{T}\ell_t(i^\star)}$$

由于 $W_T(i^\star)$ 的值总小于所有权重的和 Φ_t，可得

$$W_{T+1}(i^\star)\le\Phi_{T+1}\le Ne^{-\varepsilon\sum_t\mathbf{x}_t^\top\ell_t+\varepsilon^2\sum_t\mathbf{x}_t^\top\ell_t^2}$$

两边取对数，可得

$$-\varepsilon\sum_{t=1}^{T}\ell_t(i^\star)\le\log N-\varepsilon\sum_{t=1}^{T}\mathbf{x}_t^\top\ell_t+\varepsilon^2\sum_{t=1}^{T}\mathbf{x}_t^\top\ell_t^2$$

通过化简定理得证。

1.4　文献评述

OCO 模型最早由 Zinkevich(2003) 定义，此后在学习社区产生了广泛的影响，并在此后得到了大力推广 (见论文与综述：Hazan，2006，2011；Shalev-Shwartz，2011)。

根据专家建议进行预测的问题和加权多数算法是由 Littlestone and Warmuth(1989, 1994) 设计的。这项开创性的工作是乘法更新方法的首次使用之一。乘法更新方法是计算和学习中普遍存在的元算法，更多细节请参阅综述 Arora et al.(2012)。Hedge 算法是由 Freund and Schapire(1997) 提出的。

通用投资组合模型在 Cover(1991)中提出，是最坏情况下在线学

习模型的首批示例之一。Cover 给出了一个在指数时间内运行的通用投资组合选择的最优遗憾算法。Kalai and Vempala(2003) 给出了一个多项式时间算法，Agarwal et al.(2006)、Hazan et al.(2007) 进一步加速了该算法。该模型的许多扩展也出现在文献中，包括交易代价的增加 (Blum and Kalai, 1999) 以及与股票价格的几何布朗运动模型的关系 (Hazan and Kale, 2009)。

Awerbuch and Kleinberg(2008) 在他们有影响力的论文中提出了 OCO 在在线路由中的应用。从那时起，人们投入了大量的工作来改进初始界，并将其推广到一个具有有限反馈的完整决策框架中。该框架是 OCO 的扩展，称为 Bandit 凸优化 (Bandit Convex Optimization，BCO)。

1.5 练习

1. (Claude Shannon 的贡献)

构建两只股票的市场回报率，其中任何一只股票积累的财富呈指数级下降，而最佳的恒定再平衡投资组合的财富则呈指数级增加。更准确地说，构造两个在 $(0, \infty)$ 范围内的数列，用于表示回报，这样：

(a) 投资任何一只股票都会导致财富呈指数级下降，这意味着这些序列中每个序列的数字前缀乘积呈指数级递减。

(b) 对这两种资产进行平均投资，并在每次迭代后进行再平衡，财富将成倍增加。

2. (a) 考虑专家问题，其中损失在 0 和正实数 $G>0$ 之间。假定有一个达到预期损失上界的算法：

$$\sum_{t=1}^{T} \mathbf{E}[\ell_t(i_t)] \leq \min_{i^\star \in [N]} \sum_{t=1}^{T} \ell_t(i^\star) + c\sqrt{T \log N}$$

对于你能找到的最佳常数 c(常数 c 应该与博弈迭代次数 T 和专家数量 N 无关。假设 T 是预先知道的)。

(b) 假设上限 G 事先未知，给出一个算法，其性能渐近地与 (a) 中算法一样好，最多加上和 / 或乘一个与 T、N、G 无关的常数。证明你的结论。

3. 考虑专家问题，其中损失可能是负数，并且是 [−1,1] 范围内的实数。给出一个遗憾保证为$O(\sqrt{T \log N})$的算法，证明你的结论。

第2章

凸优化基本概念

本章简要介绍凸优化，并提出求解凸数学规划的一些基本算法。尽管离线凸优化不是本书的主要主题，在开始研究 OCO 之前，回顾基本的定义和结果很有用。这将有助于评估 OCO 的优势和局限性。此外，本章还介绍一些开展后续研究的工具。

本章的内容并不新鲜，存在广泛而详细的文献，读者可以参考 2.6 节。在这里只给出最基本的分析，并将重点放在以后对我们有用的技术上。

2.1 基本定义和设置

本章的目标是最小化 Euclidean 空间的凸子集上的连续凸函数。因此，令 $\mathcal{K} \subseteq \mathbb{R}^d$ 是 Euclidean 空间中的有界凸集和闭集。用 D 表示 $\mathcal{K}$ 直径的上界：

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{K},\ \|\mathbf{x} - \mathbf{y}\| \leq D$$

如果对于任何 $\mathbf{x}, \mathbf{y} \in \mathcal{K}$，连接 $\mathbf{x}$ 和 $\mathbf{y}$ 的线段上的所有点也属于 $\mathcal{K}$(即下式成立)，则集合 $\mathcal{K}$ 是凸的。

$$\forall \alpha \in [0, 1],\ \alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in \mathcal{K}$$