A Modern Introduction to Online Learning
在线学习现代导论

Francesco Orabona(著)
Boston University
francesco@orabona.com
子仁 (译)
luojunren17@nudt.edu.cn

2021 年 5 月 13 日

目录

List of Definitions

List of Algorithms

摘要

在这本专著中, 我通过在线凸优化的现代观点介绍了在线学习的基本概念. 这里的在线学习是指最坏情况假设下的后悔最小化框架. 我提出了一阶和二阶算法, 用于在欧几里德和非欧几里德设置下的凸损失在线学习. 所有的算法都清晰地呈现为在线镜像下降或跟随正则化领导及其变体的实例. 通过自适应和无参数在线学习算法, 特别关注调整算法参数和在无界域中学习的问题. 非凸损失通过凸替代损失和随机化来处理. 此外, 还简要讨论了机械臂的背景, 涉及敌对和随机多臂的问题. 这些笔记不需要凸分析的先验知识, 所有需要的数学工具都经过严格解释. 此外, 所有的证明都经过精心挑选, 尽可能简单和简短.

Chapter 1

什么是在线学习

考虑以下重复试验: 在每一轮 $t = 1, \ldots, T$

- 对手在 $y_t \in [0,1]$ 选择一个实数, 并对其保密;
- 你试着在 $x_t \in [0,1]$ 中选择一个数对其进行猜测;
- 对手的号码显示出来, 你支付差额的平方 $(x_t - y_t)^2$.

基本上, 我们希望尽可能精确地猜测一系列数字. 要成为一个游戏, 我们现在要决定什么是“获胜条件”. 让我们看看什么是有意义的获胜条件. 首先, 让我们简化一下游戏. 让我们假设对手正在从 $[0,1]$ 上的某个固定分布中提取数字. 但是, 他仍然可以在游戏开始时自由决定采取哪个分布. 如果我们知道分布, 我们就可以预测每一轮分布的平均值, 并知道我们支付 $\sigma^2 T$, 其中 σ^2 是分布的方差. 我们不能做得比这更好了！然而, 由于我们不知道分布情况, 自然要根据最优策略来衡量我们的策略. 也就是说, 测量优劣是自然的

$$\mathbb{E}_Y \left[\sum_{t=1}^{T} (x_t - Y)^2 \right] - \sigma^2 T, \tag{1.1}$$

或者等效地考虑平均值

$$\frac{1}{T} \mathbb{E}_Y \left[\sum_{t=1}^{T} (x_t - Y)^2 \right] - \sigma^2 \ . \tag{1.2}$$

显然, 这些量是正的, 它们似乎是一个很好的度量, 因为通过分布的方差, 它们相对于对手产生的数字的“难度”被标准化了. 这不是衡量我们“成功”的唯一选择, 但肯定是一个合理的选择. 如果 (1.1) 中的差异随时间呈次线性增长, 并且, 等效地, 如果 (1.2)中的差异随着回合数 T 变得无穷大而变为零, 则认为策略“成功”是有意义的. 也就是说, 我们希望我们的算法能够在平均轮数上接近最佳性能.

最小后悔值 考虑到我们已经对好的成功的算法标准达成一致. 现在让我们用相同的方式重写 (1.1)

$$\mathbb{E} \left[\sum_{t=1}^{T} (x_t - Y)^2 \right] - \min_{x \in [0,1]} \mathbb{E} \left[\sum_{t=1}^{T} (x - Y)^2 \right] \ .$$

现在, 最后一步: 让我们删除关于如何生成数据的假设, 考虑任意的 y_t 序列, 并继续使用相同的成功度量标准. 当然, 我们可以去掉期望因为不再存在随机性了. 所以, 我们知道我们会赢得比赛, 如果

$$\mathrm{Regret}_T := \sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2$$

随着 T 近似线性增长. 上面的数量被称为悔值, 因为它衡量了算法后悔的程度, 因为没有坚持所有回合的最优选择. 我们将表示为 Regret_T. 我们的推理应该为这个度量提供充分的理由, 然而, 在下面我们将看到, 从凸优化和机器学习的角度来看, 这也是有意义的.

现在让我们进一步概括一下这个在线游戏, 考虑到算法输出一个向量 $\boldsymbol{x}_t \in V \subseteq \mathbb{R}^d$ 它损失了 $\ell_t : V \to \mathbb{R}$ 这衡量了算法在每一轮中的预测有多好. 集合 V 称为可行集. 同样, 让我们考虑一个任意的预测因子 $\boldsymbol{u}$ 在[1] $V \subseteq \mathbb{R}^d$ 让我们把悔值参数化: $\mathrm{Regret}_T(\boldsymbol{u})$. 因此, 总而言之, 在线学习只不过是设计和分析算法, 将针对任意竞争对手的丢失函数序列的后悔值最小化

$\boldsymbol{u} \in V \subseteq \mathbb{R}^d$:

$$\mathrm{Regret}_T(\boldsymbol{u}) := \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \ .$$

这个框架非常强大, 它允许将机器学习和优化中的一系列不同问题重新表述为类似的游戏. 更一般地说, 使用后悔值框架, 我们可以分析数据不是独立的, 也不是同一分布的情况, 但我想保证算法正在“学习”一些东西. 例如, 在线学习可以用来分析

- 点击预测问题;
- 网络路由;
- 重复博弈均衡的收敛性.

它也可以用来分析随机算法, 例如, 随机梯度下降, 但分析的敌对性质可能会给你次优的结果. 例如, 它可以用于分析动量算法, 但损失的对抗性本质迫使你证明一个收敛保证, 将动量项视为一个消失的干扰, 对算法没有任何帮助.

现在让我们回到猜数字的游戏, 让我们尝试一种策略来赢得它. 当然, 这是在线学习最简单的例子之一, 没有真正的应用. 然而, 通过它, 我们将发现在线学习算法及其分析的大部分关键成分.

一个成功的策略 我们能赢猜数字的名字游戏吗? 注意, 我们没有对对手如何决定数字做任何假设. 事实上, 这些数字可以以任何方式产生, 甚至是基于我们的策略的自适应方式. 事实上, 他们可以被选择为对手, 这显然是在试图让我们输掉游戏. 这就是为什么我们称产生数字的机制为“对手”.

数字是对位选择的这一事实意味着我们可以立即排除任何基于数据统计模型的策略. 事实上, 这种方法是行不通的, 因为一旦我们对某样东西进行评估并根据我们的评估采取行动, 对手就会立即改变生成数据的方式, 从而毁了我们. 所以我们得想点别的. 然而, 从统计估计的角度来看, 在线学习算法在很多时候看起来很像经典算法, 即使它们出于不同的原因工作.

[1] In same cases, we can make the game easier for the algorithm letting it choose the prediction from a set $W \supset V$.

现在, 让我们尝试设计一种策略, 使后悔在时间上可以证明是次线性的, 不管对手如何选择数量. 首先要做的是回头看一看最好的策略, 那就是阿格明的第二任期的后悔. 这应该是显而易见的

$$x_T^\star := \operatorname*{argmin}_{x\in[0,1]} \sum_{t=1}^{T}(x-y_t)^2 = \frac{1}{T}\sum_{t=1}^{T} y_t \ .$$

现在, 由于我们不知道未来, 我们当然不能在每一轮都用 $x_T^\star$ 来猜测. 然而, 我们确实知道过去, 所以在每一轮中合理的策略可能是输出过去最好的数字. 为什么这样的策略会奏效? 可以肯定的是, 它之所以行得通, 不是因为我们期望未来会像过去一样, 而是因为未来不是真的! 相反, 我们希望利用最佳猜测在回合之间不应该改变太多这一事实, 这样我们就可以尝试随着时间“跟踪”它.

因此, 在每一轮 t 中, 我们的策略是猜测 $x_t = x_{t-1}^\star = \frac{1}{t-1}\sum_{i=1}^{t-1} y_i$. 该策略通常被称为 *Follow-the-Leader* (FTL), 因为你遵循的是在过去几轮中最优的做法 (领导者).

现在让我们试着证明这个策略确实能让我们赢得比赛. 仅对这个简单的例子, 我们将用第一原则来证明它的后悔保证. 相反, 在下面, 我们将介绍和使用非常普遍的证明方法. 首先, 我们需要一个小引理.

引理 1.1. *设 $V \subseteq \mathbb{R}^d$, $\ell_t : V \to \mathbb{R}$ 一个任意的损失函数序列. 用 $x_t^\star$ 表示 V 中过去 t 轮累计损失的最小值. 我们得到*

$$\sum_{t=1}^{T} \ell_t(x_t^\star) \le \sum_{t=1}^{T} \ell_t(x_T^\star) \ .$$

证明. 我们使用 T 归纳法来证明. 基本的条件是

$$\ell_1(x_1^\star) \le \ell_1(x_1^\star),$$

对于 $T \ge 2$, 假设 $\sum_{t=1}^{T-1} \ell_t(x_t^\star) \le \sum_{t=1}^{T-1} \ell_t(x_{T-1}^\star)$ 为真, 我们必须证明这个不等式

$$\sum_{t=1}^{T} \ell_t(x_t^\star) \le \sum_{t=1}^{T} \ell_t(x_T^\star) \ .$$

这个不等式等价于

$$\sum_{t=1}^{T-1} \ell_t(x_t^\star) \le \sum_{t=1}^{T-1} \ell_t(x_T^\star), \tag{1.3}$$

将总和中的最后一个元素移除, 因为他们是相等的, 观察

$$\sum_{t=1}^{T-1} \ell_t(x_t^\star) \le \sum_{t=1}^{T-1} \ell_t(x_{T-1}^\star),$$

通过归纳假设, 以及

$$\sum_{t=1}^{T-1} \ell_t(x_{T-1}^\star) \le \sum_{t=1}^{T-1} \ell_t(x_T^\star)$$

由于左边的 $x_{T-1}^\star$ 是 V 中的最小值, 且 $x_T^\star \in V$. 把这两个不等式联系起来, 得到 (1.3) 为真, 所以定理被证明了. □

基本上, 上面的引理量化了知道未来并适应它通常比不适应它要好.

有了这个引理, 我们现在可以证明后悔将在次增长, 特别是在时间上它将是对数的. 注意, 我们不会证明我们的策略是极大极小最优的, 即使有可能证明这个问题对时间的对数依赖性是不可避免的.

定理 1.2. 使得 $y_t \in [0,1]$ 在 $t = 1, \ldots, T$ 时为任意的数字序列. 让算法输出 $x_t = x^\star_{t-1} := \frac{1}{t-1}\sum_{i=1}^{t-1} y_i$. 得到

$$\mathrm{Regret}_T = \sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 \le 4 + 4\ln T \ .$$

证明. 我们用引理 1.1 来给悔值定一个上界:

$$\sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 = \sum_{t=1}^{T}(x^\star_{t-1} - y_t)^2 - \sum_{t=1}^{T}(x^\star_T - y_t)^2 \le \sum_{t=1}^{T}(x^\star_{t-1} - y_t)^2 - \sum_{t=1}^{T}(x^\star_t - y_t)^2 \ .$$

让我们看一下上一个方程中每个差的和. 得到

$$\begin{aligned}
(x^\star_{t-1} - y_t)^2 - (x^\star_t - y_t)^2 &= (x^\star_{t-1})^2 - 2y_t x^\star_{t-1} - (x^\star_t)^2 + 2y_t x^\star_t \\
&= (x^\star_{t-1} + x^\star_t - 2y_t)(x^\star_{t-1} - x^\star_t) \\
&\le |x^\star_{t-1} + x^\star_t - 2y_t|\,|x^\star_{t-1} - x^\star_t| \\
&\le 2|x^\star_{t-1} - x^\star_t| \\
&= 2\left|\frac{1}{t-1}\sum_{i=1}^{t-1} y_i - \frac{1}{t}\sum_{i=1}^{t} y_i\right| \\
&= 2\left|\left(\frac{1}{t-1} - \frac{1}{t}\right)\sum_{i=1}^{t-1} y_i - \frac{y_t}{t}\right| \\
&\le 2\left|\frac{1}{t(t-1)}\sum_{i=1}^{t-1} y_i\right| + \frac{2|y_t|}{t} \\
&\le \frac{2}{t} + \frac{2|y_t|}{t} \\
&\le \frac{4}{t} \ .
\end{aligned}$$

因此, 我们得到

$$\sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 \le 4\sum_{t=1}^{T}\frac{1}{t} \ .$$

为了求最后一个和的上界, 请注意, 我们正在试图找到图 1.1中绿色区域的上界, 正如你所看到的, 它的上界可以是 1 加上 $\frac{1}{t-1}$ 从 2 到 $T+1$ 的积分, 得到

$$\sum_{t=1}^{T}\frac{1}{t} \le 1 + \int_2^{T+1}\frac{1}{t-1}dt = 1 + \ln T \ . \qquad \square$$

这个策略没有什么需要强调的. 该策略没有可以调整的参数 (例如学习速率、正则化器). 请注意, 参数的存在在在线学习中是没有意义的: 我们只有一个数据流, 我们不能多次运行我们的算法来选择最好的参数! 此外, 它不需要维护过去的完整记录, 而只需要通过运行平均值来“总结”过去. 这是一种计算效率很高的算法. 当我们设计在线学习算法时, 我们将努力实现所有这些特征. 我想强调的最后一点是, 这个算法不使用梯度: 梯度很有用, 我们会经常使用它们, 但它们并不构成在线学习的整个世界.

在继续之前, 我想提醒大家, 如上所述, 这与统计机器学习的经典设置不同. 例如,“过拟合”在这里完全没有意义.“泛化差距”和与培训/测试场景相关的类似想法也是一样的. 在下一章中, 我们将介绍几种用于在线优化的算法, 其中一种将是我们在上面示例中使用的策略的严格概括.

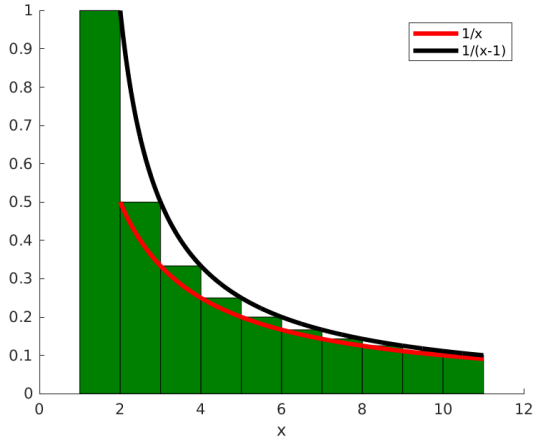

图 1.1: 积分和的上界.

Chapter 2

在线子梯度下降

在这一章中, 我们将介绍在线次梯度下降算法, 这是一种解决带有凸损失的在线问题的通用在线算法. 首先, 我们将介绍凸可微函数的在线梯度下降法.

2.1 带凸可微损失的在线学习

为了总结我们在第一章中所说的, 让我们将在线学习定义为以下一般游戏

- 对于 $t = 1, \ldots, T$
  - 输出 $\boldsymbol{x}_t \in V \subseteq \mathbb{R}^d$
  - 收获 $\ell_t : V \to \mathbb{R}$
  - 付出 $\ell_t(\boldsymbol{x}_t)$
- 结束

这个博弈的目的是最小化关于任何竞争者 u $\boldsymbol{u} \in V$ 的悔值:

$$\mathrm{Regret}_T(\boldsymbol{u}) := \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \ .$$

同时,ℓ_t 的损失确定是对抗性的. 现在, 没有额外的假设我们不可能解决这个问题. 因此, 我们必须了解我们可以做出哪些合理的假设. 通常, 我们会试图以某种方式限制损失函数的选择. 这被认为是合理的, 因为大多数时候我们可以决定从哪个集合中选择损失函数. 例如, 我们将只考虑凸损失函数. 然而, 凹凸性可能还不够, 因此我们可以将该类进一步限制为, 例如,Lipschitz 凸函数. 另一方面, 假设知道一些关于未来的事情被认为是不合理的假设, 因为我们很少能够控制未来. 一般来说, 假设越强, 我们所能证明的后悔的上限就越高. 我们将看到的最好的算法将保证我们对最弱的假设有次线性的后悔, 同时保证对容易的对手有较小的后悔.

同样重要的是要记住为什么最小化后悔是一个好的目标: 既然我们对对手如何生成损失函数没有任何假设, 那么最小化后悔是一个考虑到问题难度的良好指标. 如果一个在线学习算法能够保证一个次线

性的后悔, 这意味着它的平均性能将接近任何固定策略的性能. 如前所述, 我们将看到, 在许多情况下, 如果对手是“弱的”, 例如, 它是一个固定的损失函数随机分布, 为最坏的情况做好准备, 无论如何也不会妨碍我们得到最好的保证.

在接下来的一段时间内, 我们关注到 ℓ_t 是凸的, 这个问题叫做在线凸优化 **Online Convex Optimization** (OCO). 随后, 我们再讨论具体的非凸在线问题.

附注 2.1. *现在我要介绍一些数学概念. 如果你有凸分析的背景, 这对你来说会很容易. 另一方面, 如果你从未见过这些东西, 它们可能看起来有点吓人. 让我来告诉你看待它们的正确方式: 这些工具将使我们的工作更容易. 如果没有这些工具, 基本上就不可能设计出任何在线学习算法. 而且, 不, 在一些机器学习数据集上测试随机算法是不够的, 因为固定数据集不具有对抗性. 如果没有正确的证明, 你可能永远都不会意识到你的在线算法会因为特定的损失序列而失败, 就像 [70] 所经历的那样, 我向你保证, 一旦你理解了关键的数学概念, 在线学习实际上很容易.*

2.1.1 凸分析: 凸性

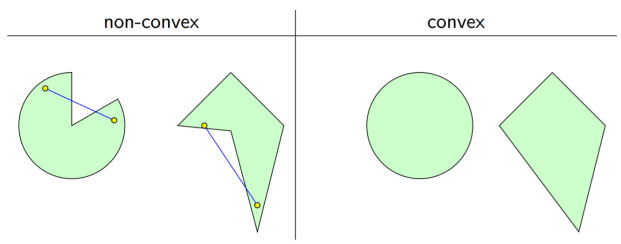

图 2.1: 凸集以及非凸集.

定义 2.2 (凸集)**.** *$V \subset \mathbb{R}^d$ 是* **凸的** *若对于任意 $\boldsymbol{x}, \boldsymbol{y} \in V$ 以及任意 $\lambda \in (0,1)$, 有 $\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in V$.*

换言之, 这意味着集合 V 没有空缺, 看图 2.1.

我们将利用 **扩展实值函数**, 这是一个可以在 $\mathbb{R} \cup \{-\infty, +\infty\}$ 取值的函数. 对于 f , 一个在 $\mathbb{R}^d$ 上的扩展实值函数, 它的域是 $\operatorname{dom} f = \{\boldsymbol{x} \in \mathbb{R}^d : f(x) < +\infty\}$.

扩展实值函数使我们可以很容易地考虑约束集, 是凸优化的标准符号, 见 Boyd and Vandenberghe [18]. 例如, 如果我想要算法的预测值 x_t 和对手 $\boldsymbol{u}$ 在一个集合 $V \subset \mathbb{R}^d$, 我可以将 $i_V(\boldsymbol{x})$ 加到损失函数, $i_V : \mathbb{R}^d \to (-\infty, +\infty]$ 是集合 V 的指标函数, 定义如下:

$$i_V(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in V, \\ +\infty, & \text{otherwise.} \end{cases}$$

这样一来, 算法和竞争对手遭受有限损失的唯一方法就是在集合 V 内进行预测. 同时, 扩展实值函数将 Fenchel 共轭的使用更为直接, 见 5.2.1.

凸函数将是在线学习的一个重要组成部分.

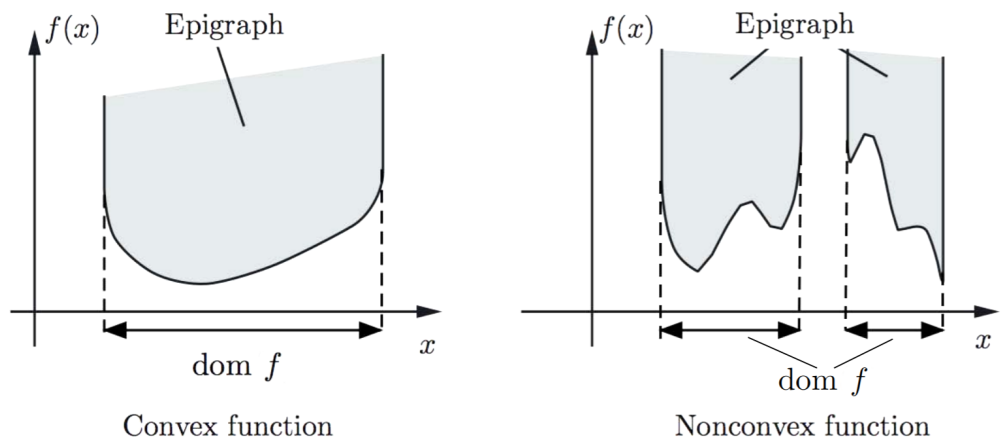

图 2.2: 凸和非凸函数.

定义 2.3 (定义)**.** *设 $f:\mathbb{R}^d \to [-\infty,+\infty]$. 如果函数的前提条件 $\{(\boldsymbol{x},y)\in\mathbb{R}^{d+1}|y\geq f(x)\}$ 是凸的,f 是凸的.*

我们可以在图2.2中看到这个定义的可视化, 注意, 这个定义意味着凸函数的定义域是凸的.

同时, 如果 $f:V\subseteq\mathbb{R}^d\to\mathbb{R}$ 是凸的, $f+i_V:\mathbb{R}^d\to(-\infty,+\infty]$ 也是凸的. 注意 $i_V(\boldsymbol{x})$ 是凸的, 如果 V 是凸的, 每个凸集都与一个凸函数相关.

对于不假定值为 $-\infty$ 的凸函数, 上述定义可得到以下刻画.

定理 2.4 ([72, 引理 4.1])**.** *设 $f:\mathbb{R}^d\to(-\infty,+\infty]$ 和 $\operatorname{dom} f$ 是凸集. 那么 f 是凸集, 对于任意的 $0<\lambda<1$, 有*

$$f(\lambda\boldsymbol{x}+(1-\lambda)\boldsymbol{y})\leq\lambda f(\boldsymbol{x})+(1-\lambda)f(\boldsymbol{y})\ \forall\boldsymbol{x},\boldsymbol{y}\in\operatorname{dom} f\ .$$

例 2.5. *凸函数最简单的例子是仿射函数: $f(\boldsymbol{x})=\langle\boldsymbol{z},\boldsymbol{x}\rangle+b$.*

例 2.6. *规范总是凸的, 证明是留作练习的.*

如何识别凸函数? 在最普遍的情况下, 你必须依赖定义. 然而, 在大多数情况下, 我们会认为它们是由保持凸性的操作组成的. 例如

- f 和 g 是凸的, 那么非负权值的线性组合也是凸的.
- 使用仿射变换的复合保持凹凸性.
- 凸函数的点态上界为凸.

证明留作练习. 可微凸函数的一个重要性质是可以构造函数的线性下界.

定理 2.7 ([72, 定理 25.1 and 推论 25.1.1])**.** *设 $f:\mathbb{R}^d\to(-\infty,+\infty]$ 是凸函数, 同时使 $\boldsymbol{x}\in\operatorname{int}\operatorname{dom} f$. 如果 f 在 $\boldsymbol{x}$ 处是可导的, 那么*

$$f(\boldsymbol{y})\geq f(\boldsymbol{x})+\langle\nabla f(\boldsymbol{x}),\boldsymbol{y}-\boldsymbol{x}\rangle,\ \boldsymbol{y}\in\mathbb{R}^d\ .$$

我们也将使用凸函数的一阶最优性条件:

定理 2.8. *设 V 是一个凸的非空集合, $\boldsymbol{x}^\star \in V$ 以及 f 是一个凸函数, 在包含 V 的开集上可微. 那么 $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}\in V}\ f(\boldsymbol{x})$ 若 $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{y} - \boldsymbol{x}^\star \rangle \geq 0,\ \forall \boldsymbol{y} \in V$.*

总而言之, 在受约束最小的时候, 梯度使得比 90° 或者少于 90°, 90° or less with all the feasible variations $\boldsymbol{y} - \boldsymbol{x}^\star$, 我们不能在 V 里面最小化更多的函数.

凸函数的另一个关键性质是 Jensen 的不等式.

定理 2.9. *设 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 是一个可测量的凸函数且 $\boldsymbol{x}$ 是一个在一些概率空间上的 $\mathbb{R}^d$-valued 随机变量, $\mathbb{E}[\boldsymbol{x}]$ 存在且 $\boldsymbol{x} \in \operatorname{dom} f$ 的概率为 1. 则*

$$E[f(\boldsymbol{x})] \geq f(E[\boldsymbol{x}])\ .$$

现在我们来看凸可微的情况下的第一个 OCO 算法.

2.1.2 在线梯度下降

在第一章中, 我们看到了在猜谜游戏中获得对数后悔的一个简单策略. 当时的策略是用过去最好的, 那就是跟随最好的策略. 在公式,

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}} \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x}),$$

在第一轮, 我们可以打任何允许的点. 有人可能会怀疑这种策略是否总是有效, 但答案是否定的!

例 2.10 (FTL 的缺陷)**.** *设 $V = [-1, 1]$, 考虑损失函数序列 $\ell_t(x) = z_t x + i_V(x)$,*

$$\begin{aligned} z_1 &= -0.5, \\ z_t &= 1,\ t = 2, 4, \ldots \\ z_t &= -1,\ t = 3, 5, \ldots \end{aligned}$$

第一轮中对 FTL 的预测是 $[-1, 1]$ 中的任意一个, 当 t 为奇数时, FTL 的预测会是 $x_t = 1$, t 为偶数时, $x_t = -1$ 因为 FTL 算法的累计损失为 T , 而固定解 $u = 0$ 的累计损失为 0. 故, FTL 的后悔值为 T.

因此, 我们将展示一种替代策略, 它保证凸 Lipschitz 函数的次线性后悔. 之后我们还将证明对 T 的依赖是最优的. 该策略被称为投影在线梯度下降, 或简称在线梯度下降, 参见算法 2.1. 它包含更新的预测算法在每一个时间步的负梯度方向移动接收和损失预测的可行集. 它是类似于随机梯度下降法, 但它不是一回事: 这里的损失函数是不同的每一步. 我们稍后将看到在线梯度下降法也可以用作随机梯度下降法.

首先, 我们展示以下两个引理. 第一个引理证明了欧几里得投影总是减小与集合内点的距离.

命题 2.11. *设 $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{y} \in V$, $V \subseteq \mathbb{R}^d$ 一个非空的闭凸集, 定义 $\Pi_V(\boldsymbol{x}) := \operatorname{argmin}_{\boldsymbol{y}\in V}\ \|\boldsymbol{x} - \boldsymbol{y}\|_2$, 则 $\|\Pi_V(\boldsymbol{x}) - \boldsymbol{y}\|_2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2$.*

Algorithm 2.1 Projected Online Gradient Descent

Require: Non-empty closed convex set $V \subseteq \mathbb{R}^d$, $\boldsymbol{x}_1 \in V$, $\eta_1, \cdots, \eta_T > 0$

```
for t = 1 to T do
    Output x_t
    Receive ℓ_t : R^d → (−∞, +∞] and pay ℓ_t(x_t)
    Set g_t = ∇ℓ_t(x_t)
    x_{t+1} = Π_V(x_t − η_t g_t) = argmin_{y∈V} ‖x_t − η_t g_t − y‖_2
end for
```

证明. 由定理 2.8的最优性条件, 得

$$\langle \Pi_V(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y} - \Pi_V(\boldsymbol{x}) \rangle \geq 0 \ .$$

则

$$\begin{aligned}
\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 &= \|\boldsymbol{y} - \Pi_V(\boldsymbol{x}) + \Pi_V(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\
&= \|\boldsymbol{y} - \Pi_V(\boldsymbol{x})\|_2^2 + 2\langle \boldsymbol{y} - \Pi_V(\boldsymbol{x}), \Pi_V(\boldsymbol{x}) - \boldsymbol{x} \rangle + \|\Pi_V(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\
&\geq \|\boldsymbol{y} - \Pi_V(\boldsymbol{x})\|_2^2 \ .
\end{aligned}$$

□

下一个引理上界为算法的一次迭代中的后悔.

引理 2.12. 设 $V \subseteq \mathbb{R}^d$ 是一个非空闭凸集, $\ell_t : V \to \mathbb{R}$ 是一个在包含 V 的开集上可微的凸函数, 设 $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{x}_t)$. 则, $\forall \boldsymbol{u} \in V$, 以下的不等式成立.

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 + \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|_2^2 \ .$$

证明. 从 2.11命题, 以及 2.7定理, 得

$$\begin{aligned}
\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 - \|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 &\leq \|\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t - \boldsymbol{u}\|_2^2 - \|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 \\
&= -2\eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle + \eta_t^2 \|\boldsymbol{g}_t\|_2^2 \\
&\leq -2\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) + \eta_t^2 \|\boldsymbol{g}_t\|_2^2 \ .
\end{aligned}$$

重新排序, 我们有给定的界. □

我们可以证明以下的后悔保证.

定理 2.13. 设 $V \subseteq \mathbb{R}^d$ 是一个直径为 D 的非空闭凸集, *i.e.* $\max_{\boldsymbol{x},\boldsymbol{y} \in V} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq D$. 设 $\ell_1, \cdots, \ell_T$ 是任意的凸函数序列 $\ell_t : V \to \mathbb{R}$ 对于 $t = 1, \ldots, T$ 时, 在包含 V 的开集上可微, 选择任何一个 $\boldsymbol{x}_1 \in V$ 假设 $\eta_{t+1} \leq \eta_t$, $t = 1, \ldots, T$. 则, $\forall \boldsymbol{u} \in V$, 以下是悔值的约束

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{D^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \ .$$

更多地, 如果 η_t 不变, *i.e.* $\eta_t = \eta\ \forall t = 1, \cdots, T$, 则

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{\|\boldsymbol{u} - \boldsymbol{x}_1\|_2^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \ .$$

证明. 将引理 2.12中的不等式除以 η_t , $t = 1, \cdots, T$, 得

$$\begin{aligned}
\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\left(\frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_t}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{1}{2\eta_1}\|\boldsymbol{x}_1 - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_T}\|\boldsymbol{x}_{T+1} - \boldsymbol{u}\|_2^2 + \sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&\leq \frac{1}{2\eta_1}D^2 + D^2\sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{1}{2\eta_1}D^2 + D^2\left(\frac{1}{2\eta_T} - \frac{1}{2\eta_1}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{D^2}{2\eta_T} + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \ .
\end{aligned}$$

同样, 当 η_t 不变时, 得

$$\begin{aligned}
\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\left(\frac{1}{2\eta}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2\right) + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{1}{2\eta}\|\boldsymbol{x}_1 - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{x}_{T+1} - \boldsymbol{u}\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \\
&\leq \frac{1}{2\eta}\|\boldsymbol{x}_1 - \boldsymbol{u}\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \ .
\end{aligned}$$

□

我们能立即观察到一些东西.

- 如果我们想要使用时变学习速率, 你需要一个有界的域 V. 然而, 这个假设在大多数机器学习应用中是错误的. 然而, 在随机设置中, 如果使用非均匀平均, 您仍然可以在无界域的 SGD 中使用时变学习率. 我们将在第 3看到这一点.

- 另一个重要的观察结果是, 悔恨约束可以帮助我们选择学习率 η_t. 事实上, 这是我们唯一的指导方针. 任何其他没有经过后悔分析的选择都是不合理的

正如我们所说, 学习率等参数的存在在在线学习中毫无意义. 所以, 我们必须决定一个策略来设置它们. 一个简单的选择是找到一个常数学习率, 使固定次数的迭代的边界最小. 我们必须考虑表达式

$$\frac{\|\boldsymbol{u} - \boldsymbol{x}_1\|_2^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2$$

对 η 求最小值, 很容易得出最优的 η 是 $\frac{\|\boldsymbol{u}-\boldsymbol{x}_1\|_2}{\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2}}$, 给后悔值进行约束

$$\|\boldsymbol{u}-\boldsymbol{x}_1\|_2 \sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2} \ .$$

然而, 我们有一个问题: 为了使用这个步长, 我们必须知道所有未来的梯度以及最优解和初始点之间的距离! 这显然是不可能的: 记住, 对手可以选择函数序列. 因此, 它可以观察你对学习速率的选择, 并决定顺序, 这样你的学习速率现在是错误的!

事实上, 事实证明, 这种利率是完全不可能的, 因为它被一个下限所排除. 然而, 我们将看到, 使用自适应 (Section 4.2) 和无参数算法 (Chapter 9) 确实可以实现非常相似的速率. 目前, 我们可以观察到, 我们可能乐意最小化一个较宽松的上界. 假设梯度的上界以 L 为界, 即 $\|\boldsymbol{g}_t\|_2 \le L$. 同时, 假设直径有界, 我们可以通过 D 来提高 $\|\boldsymbol{u}-\boldsymbol{x}_1\|_2$ 的界. 得到

$$\eta^\star = \operatorname*{argmin}_\eta \ \frac{D^2}{2\eta} + \frac{\eta L^2 T}{2} = \frac{D}{L\sqrt{T}},$$

得到悔值的界为

$$DL\sqrt{T} \ . \tag{2.1}$$

所以, 后悔的确是次线性的时间. 在下一节中, 我们将看到如何通过使用次梯度来消除可微性假设.

2.2 在线次梯度下降

在上一节中, 我们介绍了投影在线梯度下降法. 然而 ℓ_t 的可微性假设是相当强的. 当损失是凸的但不可微时, 会发生什么? 例如 $\ell_t(x) = |x-10|$. 请注意, 这种情况比人们想象的更常见. 例如,hinge 损失, $\ell_t(\boldsymbol{w}) = \max(1-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle, 0)$, 以及用于神经网络的 ReLU 激活函数, $\ell_t(x) = \max(x, 0)$, 是不可微的. 我们可以用在线梯度下降法, 用次梯度代替梯度. 为此, 我们需要更多的凸分析!

2.2.1 凸分析: 次梯度

首先, 我们需要一个技术定义.

定义 2.14 (本征函数)**.** *如果一个函数 f 不是 $-\infty$ 且在某处是有限的, 则 f 称为本征函数.*

在这本书中, 我们主要感兴趣的是凸本征函数, 它基本上更符合我们对凸函数的直觉. 我们先正式定义什么是次梯度.

定义 2.15 (Subgradient)**.** *对于一个本征函数 $f : \mathbb{R}^d \to (-\infty, +\infty]$, 在 $\boldsymbol{x} \in \mathbb{R}^d$ 定义 f 的一个次梯度, 当向量 $\boldsymbol{g} \in \mathbb{R}^d$ 满足*

$$f(\boldsymbol{y}) \ge f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle, \ \forall \boldsymbol{y} \in \mathbb{R}^d \ .$$

基本上, f 在 $\boldsymbol{x}$ 中的次梯度允许我们构造到 f 的线性下界的任何向量 $\boldsymbol{g}$. 这个次梯度不是惟一的, 我们用 $\partial f(\boldsymbol{x})$ 表示 f 在 $\boldsymbol{x}$ 中的次梯度集, 称为 f 在 $\boldsymbol{x}$ 的次微分.

观察到如果 f 是本征和凸的, 那么 $\partial f(\boldsymbol{x})$ 对于 $\boldsymbol{x} \notin \operatorname{dom} f$ 是空的, 因为当 $f(\boldsymbol{x}) = +\infty$, 不等式不能被满足. 同时, 用 $\operatorname{dom} \partial f$ 表示的 ∂f 域, 是所有 $\boldsymbol{x} \in \mathbb{R}^d$ 集合, 例如 $\partial f(x)$ 非空; 它是 $\operatorname{dom} f$ 的一个子集. 一个固有凸函数 f 在 $\operatorname{int} \operatorname{dom} f$ [72, Theorem 23.4] 中总是次可微的.

如果函数是凸的, 在 $\boldsymbol{x}$ 中可微, f 在 $\boldsymbol{x}$ 中是有限的, 则有, 该子微分由一个等于 $\nabla f(\boldsymbol{x})$ [72, Theorem 25.1] 的唯一元素组成. 同样, 我们也可以计算函数和的次梯度.

定理 2.16 ([72, Theorem 23.8],[11, Corollary 16.39])**.** *设 $f_1, \ldots, f_m$ 是 $\mathbb{R}^d$ 上的固有凸函数, 且 $f = f_1 + \cdots + f_m$. 则 $\partial f(\boldsymbol{x}) \supset \partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}), \forall \boldsymbol{x}$. 若 $\operatorname{dom} f_m \cap \bigcap_{i=1}^{m-1} \operatorname{int} \operatorname{dom} f_i \neq \{\}$, 则 $\partial f(\boldsymbol{x}) = \partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}), \forall x$.*

例 2.17. *设 $f(x) = |x|$, 则次微分集 $\partial f(x)$ 为*

$$\partial f(x) = \begin{cases} \{1\}, & x > 0, \\ [-1, 1], & x = 0, \\ \{-1\}, & x < 0 \ . \end{cases}$$

例 2.18. *我们来计算一个非空凸集的指示函数的次梯度 $V \subset \mathbb{R}^d$. 通过定义, $\boldsymbol{g} \in \partial i_V(\boldsymbol{x})$ 若*

$$i_V(\boldsymbol{y}) \geq i_V(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle, \ \forall \boldsymbol{y} \in \mathbb{R}^d \ .$$

*该条件表明 $\boldsymbol{x} \in V$ 和 $0 \geq \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle, \forall \boldsymbol{y} \in V$ (因为 $\boldsymbol{y} \notin V$ 总是得到验证). The set of 所有满足以上不等式的 $\boldsymbol{g}$ 被称为 V **在 $\boldsymbol{x}$ 上的法锥**. 对于任意的法锥 $\boldsymbol{x} \in \operatorname{int} V = \{\boldsymbol{0}\}$ (提示: take $\boldsymbol{y} = \boldsymbol{x} + \epsilon \boldsymbol{g}$). 例如, 对于 $V = \{\boldsymbol{x} \in \mathbb{R}^d | \|\boldsymbol{x}\|_2 \leq 1\}$, $\partial i_V(\boldsymbol{x}) = \{\alpha \boldsymbol{x} | \alpha \geq 0\}$ 有 $\boldsymbol{x} : \|\boldsymbol{x}\| = 1$.*

另一个有用的定理是计算凸函数的点态极大值的次微分.

定理 2.19 ([11, Theorem 18.5])**.** *设 $(f_i)_{i \in I}$ 是在 $\mathbb{R}^d$ to $(-\infty, +\infty]$ 上的有限凸函数集, 假设 $\boldsymbol{x} \in \bigcap_{i \in I} \operatorname{dom} f_i$ 和 f_i 一直在 $\boldsymbol{x}$ 上. 设 $F = \max_{i \in I} f_i$, 设 $A(\boldsymbol{x}) = \{i \in I | f_i(\boldsymbol{x}) = F(\boldsymbol{x})\}$ 活动函数的集合, 则*

$$\partial F(\boldsymbol{x}) = \operatorname{conv} \bigcup_{i \in A(\boldsymbol{x})} \partial f_i(\boldsymbol{x}) \ .$$

例 2.20 (Subgradients of the Hinge loss)**.** *考虑损失函数 $\ell(\boldsymbol{x}) = \max(1 - \langle \boldsymbol{z}, \boldsymbol{x} \rangle, 0)$ 对于 $\boldsymbol{z} \in \mathbb{R}^d$. 次可微集合为*

$$\partial \ell(\boldsymbol{x}) = \begin{cases} \{\boldsymbol{0}\}, & 1 - \langle \boldsymbol{z}, \boldsymbol{x} \rangle < 0 \\ \{-\alpha \boldsymbol{z} | \alpha \in [0, 1]\}, & 1 - \langle \boldsymbol{z}, \boldsymbol{x} \rangle = 0 \\ \{-\boldsymbol{z}\}, & \textit{otherwise} \end{cases} \ .$$

定义 2.21 (Lipschitz Function)**.** *设 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 是 **L-Lipschitz** 在集合 V 上 w.r.t a norm $\|\cdot\|$ if $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L \|\boldsymbol{x} - \boldsymbol{y}\|, \ \forall \boldsymbol{x}, \boldsymbol{y} \in V$.*

我们还得到了凸 Lipschitz 函数的次梯度范数的上界.

定理 2.22. *设 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 是合理和凸的. 则 f 是在 $\operatorname{int} \operatorname{dom} f$ w.r.t 上的 L-Lipschitz 函数. the L_2 norm iff 对于所有的 $\boldsymbol{x} \in \operatorname{int} \operatorname{dom} f$ 和 $\boldsymbol{g} \in \partial f(\boldsymbol{x})$ 有 $\|\boldsymbol{g}\|_2 \leq L$.*

证明. 设 f L-Lipschitz, 则 $|f(\boldsymbol{x})-f(\boldsymbol{y})| \leq L\|\boldsymbol{x}-\boldsymbol{y}\|_2$, $\forall \boldsymbol{x},\boldsymbol{y} \in \operatorname{dom} f$. 对于 $\epsilon > 0$ 足够小 $\boldsymbol{y} = \boldsymbol{x} + \epsilon \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2} \in \operatorname{int}\operatorname{dom} f$, 则

$$L\epsilon = L\|\boldsymbol{x}-\boldsymbol{y}\|_2 \geq |f(\boldsymbol{y})-f(\boldsymbol{x})| \geq f(\boldsymbol{y})-f(\boldsymbol{x}) \geq \langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle = \epsilon\|\boldsymbol{g}\|_2,$$

表明 $\|\boldsymbol{g}\|_2 \leq L$.

另一方面, 次梯度的定义和 Cauchy-Schwartz 不等式得到

$$f(\boldsymbol{x})-f(\boldsymbol{y}) \leq \|\boldsymbol{g}\|_2\|\boldsymbol{x}-\boldsymbol{y}\|_2 \leq L\|\boldsymbol{x}-\boldsymbol{y}\|_2,$$

对于任意 $\boldsymbol{x},\boldsymbol{y} \in \operatorname{int}\operatorname{dom} f$. 得 $\boldsymbol{g} \in \partial f(\boldsymbol{y})$, 同时

$$f(\boldsymbol{y})-f(\boldsymbol{x}) \leq L\|\boldsymbol{x}-\boldsymbol{y}\|_2,$$

得证. □

2.2.2 次梯度分析

正如我向你保证的, 有了合适的数学工具, 在线算法分析变得很容易. 事实上, 从梯度到次梯度的转换是免费的! 事实上, 我们对具有可微损失的 OGD 的分析是使用次梯度而不是梯度. 原因是梯度的唯一性质我们在定理2.13 的证明中用到的是

$$\ell_t(\boldsymbol{x}) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle,$$

其中 $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{x}_t)$. 然而, 当 $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$ 时, 相同的性质也成立. 所以, 我们可以这样表述在线次梯度下降算法, 唯一的区别是第 4 行. 同样, 我们证明的后悔界也成立, 就是用次微分改变可微性, 用次梯度改变梯度. 特别地, 我们有以下引理.

Algorithm 2.2 Projected Online Subgradient Descent

Require: Non-empty closed convex set $V \subseteq \mathbb{R}^d$, $\boldsymbol{x}_1 \in V$, $\eta_1, \cdots, \eta_T > 0$

1: **for** $t = 1$ **to** T **do**
2: Output $\boldsymbol{x}_t$
3: Receive $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ and pay $\ell_t(\boldsymbol{x}_t)$
4: Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
5: $\boldsymbol{x}_{t+1} = \Pi_V(\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t) = \operatorname{argmin}_{\boldsymbol{y}\in V} \ \|\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t - \boldsymbol{y}\|_2$
6: **end for**

引理 2.23. 设 $V \subseteq \mathbb{R}^d$ 是一个非空闭凸集, 且 $\ell_t : V \to \mathbb{R}$ 是一个包含 V 的, 且在开集上次可微的凸函数, 设 $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. 则, $\forall \boldsymbol{u} \in V$, 以下不等式成立

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle \leq \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 + \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|_2^2 \ .$$

2.3 从凸损失到线性损失

让我们更深入地了解这一步

$$\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle, \forall \boldsymbol{u} \in \mathbb{R}^d \ .$$

随着时间的推移, 我们得到了

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle, \forall \boldsymbol{u} \in \mathbb{R}^d \ .$$

现在, 定义线性 (和凸) 损失 $\tilde{\ell}_t(\boldsymbol{x}) := \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$, 得

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{u}) \ .$$

这比它看起来的更强大: 我们对凸损失的后悔有上限 ℓ_t , 对另一个线性损失序列的后悔有上限. 这一点很重要, 因为这意味着我们可以构建只处理线性损失的在线算法, 通过上述减少, 它们可以无缝地用作 OCO 算法! 注意, 这并不意味着这种减少总是最优的, 不是这样的! 但是, 它允许我们在许多有趣的情况下轻松地构造最优 OCO 算法. 所以, 我们通常只考虑最小化线性后悔的问题.

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle, \forall \boldsymbol{u} \in V \subseteq \mathbb{R}^d \ .$$

这个问题称为在线线性优化 (OLO).

例 2.24. *考虑第一类的猜谜游戏, 我们可以用在线梯度下降法很容易地解决它. 实际上, 我们只需要计算梯度, 证明它们有界, 并找到一种方法来计算实数在 $[0,1]$ 中的投影, 则 $\ell_t'(x) = 2(x - y_t)$, 这是 $x, y_t \in [0,1]$ 的界限. 在 $[0,1]$ 的投影为 $\Pi_{[0,1]}(x) = \min(\max(x,0),1)$. 在最优学习率下, 产生的后悔是 $O(\sqrt{T})$. 这比我们在第一节课中发现的情况更糟, 这表明减少并不总是会带来最好的后悔.*

例 2.25. *再次考虑第一类的猜测游戏, 但现在改变损失函数的绝对损失的差异: $\ell_t(x) = |x - y_t|$. 现在我们需要使用在线次梯度下降法, 因为函数是不可微的. 我们可以很容易地看到*

$$\partial \ell_t(x) = \begin{cases} \{1\}, & x > y_t \\ [-1,1], & x = y_t \\ \{-1\}, & x < y_t \ . \end{cases}$$

同样, 在这个问题上以最优学习率运行在线次梯度下降会立即给我们一个 $O(\sqrt{T})$ 的悔值, 而不需要考虑一个特定的策略.

2.4 历史片段

具有时变学习率的投影在线次梯度下降和“在线凸优化”是 Zinkevich [90]提出的, 但该框架早前由 Gordon [37]提出. 注意, 如果我们考虑优化文献, 优化人员从不把自己限制在有界的情况下.

Chapter 3

从在线到批处理变换

这是一个很好的时机, 休息一下在线学习理论, 看看在线学习在其他领域的一些应用. 例如, 我们可能想知道在线学习和随机优化之间有什么联系. 假设在线投影 (Sub) 梯度下降与随机投影 (Sub) 梯度下降看起来基本相同, 他们肯定有一些相同点. 例如, 我们可以将凸函数的随机优化简化为 OCO. 让我们来看看.

定理 3.1. *设 $F(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\xi})]$ 它的期望是 w.r.t. 从 ρ 得出的 $\boldsymbol{\xi}$ 在某个向量空间 X 上, 且 $f : \mathbb{R}^d \times X \to (-\infty, +\infty]$ 在第一个论证中为凸. 画出 T 个 $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_T$ i.i.d. 从 ρ 中, 构造损失序列 $\ell_t(\boldsymbol{x}) = \alpha_t f(\boldsymbol{x}, \boldsymbol{\xi}_t)$, 其中 $\alpha_t > 0$ 是确定的. 对损失 ℓ_t 运行任何 OCO 算法, 构建预测序列 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$. 则, 我们得到*

$$\mathbb{E}\left[F\left(\frac{1}{\sum_{t=1}^T \alpha_t} \sum_{t=1}^T \alpha_t \boldsymbol{x}_t\right)\right] \leq F(\boldsymbol{u}) + \frac{\mathbb{E}[\mathrm{Regret}_T(\boldsymbol{u})]}{\sum_{t=1}^T \alpha_t}, \ \forall \boldsymbol{u} \in \mathbb{R}^d,$$

where the expectation is with respect to $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_T$.

证明. 我们首先展示

$$\mathbb{E}\left[\sum_{t=1}^T \alpha_t F(\boldsymbol{x}_t)\right] = \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)\right] \ . \tag{3.1}$$

实际上, 从期望的线性度来看

$$\mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)\right] = \sum_{t=1}^T \mathbb{E}\left[\ell_t(\boldsymbol{x}_t)\right] \ .$$

根据总期望定律, 我们得到

$$\mathbb{E}\left[\ell_t(\boldsymbol{x}_t)\right] = \mathbb{E}\left[\mathbb{E}[\ell_t(\boldsymbol{x}_t)|\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{t-1}\right] = \mathbb{E}\left[\mathbb{E}[\alpha_t f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)|\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{t-1}]\right] = \mathbb{E}\left[\alpha_t F(\boldsymbol{x}_t)\right],$$

我们利用 $\boldsymbol{x}_t$ 和 α_t 仅仅取决于 $\boldsymbol{\xi}_1, \ldots, \boldsymbol{x}_{t-1}$ 的事实. 因此, (3.1) 被证明.

它只需要用到 Jensen 不等式, 以及 F 是凸的事实, 则

$$F\left(\frac{1}{\sum_{t=1}^T \alpha_t} \sum_{t=1}^T \alpha_t \boldsymbol{x}_t\right) \leq \frac{1}{\sum_{t=1}^T \alpha_t} \sum_{t=1}^T \alpha_t F(\boldsymbol{x}_t) \ .$$

将悔值除以 $\sum_{t=1}^T \alpha_t$, 同时结合上面的定理, 得到上述定理. □

现在让我们看看这个结果的一些应用: 让我们看看如何使用上面的定理来将在线次梯度下降转换为随机次梯度下降, 以最小化分类器的训练误差.

例 3.2. *考虑一个二元分类的问题, 输入 $\boldsymbol{z}_i \in \mathbb{R}^d$ 以及输出 $y_i \in \{-1, 1\}$. 损失函数是铰链损失: $f(\boldsymbol{x}, (\boldsymbol{z}, y)) = \max(1 - y\langle \boldsymbol{z}, \boldsymbol{x}\rangle, 0)$. 假设你想在 N 个样本的训练集上最小化训练误差, $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^N$. 同时假设样本 L_2 最大是 R. 即, 我们想要最小化*

$$\min_{\boldsymbol{x}} \ F(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^N \max(1 - y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle, 0) \ .$$

使用 OGD 对 T 迭代运行定理 3.1 中描述的简化版本. 每次迭代, 构造 $\ell_t(\boldsymbol{x}) = \max(1 - y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle, 0)$, 从 1 到 N 均匀随机抽样一个训练点. 设 $\boldsymbol{x}_1 = \boldsymbol{0}$ 和 $\eta = \frac{1}{R\sqrt{T}}$. 有

$$\mathbb{E}\left[F\left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{x}_t\right)\right] - F(\boldsymbol{x}^\star) \le R\frac{\|\boldsymbol{x}^\star\|_2^2 + 1}{2\sqrt{T}} \ .$$

换言之, 我们使用 OCO 算法对函数进行随机优化, 将后悔保证转化为收敛速度保证.

在这最后一个例子中, 我们必须使用一个常数学习速率能够最小化训练误差在整个空间 $\mathbb{R}^d$. 下一个, 我们将看到一个不同的方法, 允许使用不同学习速率而不需要的有界可行集.

例 3.3. *考虑与前一个例子相同的设置, 让我们改变构建在线损失的方式. 使用 $\ell_t(\boldsymbol{x}) = \frac{1}{R\sqrt{t}}\max(1 - y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle, 0)$ 以及步长 $\eta = 1$. 因此, 有*

$$\mathbb{E}\left[F\left(\sum_{t=1}^T \frac{1}{R\sqrt{t}}\boldsymbol{x}_t\right)\right] - F(\boldsymbol{x}^\star) \le \frac{\|\boldsymbol{x}^\star\|_2^2}{2\sum_{t=1}^T \frac{1}{R\sqrt{t}}} + \frac{1}{2\sum_{t=1}^T \frac{1}{R\sqrt{t}}}\sum_{t=1}^T \frac{1}{t} \le R\frac{\|\boldsymbol{x}^\star\|_2^2 + 1 + \ln T}{4\sqrt{T+1} - 4},$$

我们使用 $\sum_{t=1}^T \frac{1}{\sqrt{t}} \ge 2\sqrt{T+1} - 2$.

我强调了这样一个事实, 即定义后悔的唯一有意义的方式是使用可行集中的任意点. 在我们考虑无约束的 OLO 的情况下, 这一点很明显, 因为最优竞争对手是无界的. 但是, 在不受约束的 OCO 中也是如此. 让我们来看一个例子.

例 3.4. *考虑一个二元分类问题, 输入 $\boldsymbol{z}_i \in \mathbb{R}^d$ 输出 $y_i \in \{-1, 1\}$. 损失函数是逻辑损失: $f(\boldsymbol{x}, (\boldsymbol{z}, y)) = \ln(1 + \exp(-y\langle \boldsymbol{z}, \boldsymbol{x}\rangle))$. 假设你想在 N 个样本的训练集上最小化训练误差, $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^N$. 同时, 同时假设样本 L_2 最大是 R. 即, 我们想要最小化*

$$\min_{\boldsymbol{x}} \ F(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^N \ln(1 + \exp(-y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle)) \ .$$

使用 OGD 对 T 迭代运行定理 3.1 中描述的简化版本. 在每次迭代中, 构造 $\ell_t(\boldsymbol{x}) = \ln(1+\exp(-y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle))$, 从 1 到 N 均匀随机抽样一个训练点. 设 $\boldsymbol{x}_1 = \boldsymbol{0}$ 和 $\eta = \frac{1}{R\sqrt{T}}$. 得到

$$\mathbb{E}\left[F\left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{x}_t\right)\right] \le \frac{R}{2\sqrt{T}} + \min_{\boldsymbol{u}\in\mathbb{R}^d} F(\boldsymbol{u}) + R\frac{\|\boldsymbol{u}\|^2}{2\sqrt{T}} \ .$$

也就是说, 我们将处于正则化经验风险最小化问题的最优值的 $\frac{R}{2\sqrt{T}}$. 其中正则化的权值为 $\frac{R}{2\sqrt{T}}$. 现在, 让我们考虑训练集是线性可分的情况, 这意味 F 是 0 , 最优解不存在, 也就是说, 它的范数等于无穷. 因此, 任何依赖于 $\boldsymbol{x}^\star$ 的趋同保证都是空洞的. 另一方面, 我们上面的保证仍然是完全合理的. 注意, 上面的例子只处理训练错误. 然而, 在线到批处理转换还有一个更有趣的应用, 那就是直接最小化泛化误差.

注意, 上面的例子只处理训练错误. 然而, 在线到批处理转换还有一个更有趣的应用, 那就是直接最小化泛化误差.

3.1 不可知论 PAC 学习

我们现在考虑的是与目前所见不同的背景. 我们将假设我们有一个由向量 $\boldsymbol{x}$ 参数化的预测策略 $\phi_{\boldsymbol{x}}$, 我们想了解输入 $\boldsymbol{z}$ 和它相关的标签 y 之间的关系. 此外, 我们将假设 $(\boldsymbol{z}, y)$ 是从一个联合概率分布 ρ 得出的. 此外, 我们配备了一个损失函数, 用来衡量我们的预测 $\hat{y} = \phi_{\boldsymbol{x}}(\boldsymbol{z})$ 与真正的标签 y 相比有多好, 即 $\ell(\hat{y}, y)$. 因此, 学习关系可以作为最小化预测器的预期损失

$$\min_{\boldsymbol{x} \in V} \ \mathbb{E}_{(\boldsymbol{z},y)\sim\rho}[\ell(\phi_{\boldsymbol{x}}(\boldsymbol{z}), y)] \ .$$

在机器学习术语中, 上面的对象只不过是我们的预测器的测试误差. 请注意, 上面的设置假设有标记的样本, 但考虑到 Vapnik 的一般学习设置, 我们可以将其进一步推广, 其中我们将预测函数折叠, 并在一个独特的函数中丢失. 例如, 这允许以相同的统一方式对待监督学习和非监督学习. 所以, 我们想把风险降到最低

$$\min_{\boldsymbol{x} \in V} \ (\mathrm{Risk}(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}\sim\rho}[f(\boldsymbol{x}, \boldsymbol{\xi})]) \, ,$$

ρ 是 D 上的一个未知分布, 且 $f : \mathbb{R}^d \times D \to \mathbb{R}$ 是可测量的第二个论证. 同理, 所有可以用 V 中的向量 $\boldsymbol{x}$ 表示的预测因子的集合 $\mathbb{F}$ 被称为假设类.

例 3.5. *在一个线性回归任务中, 损失是平方损失, 有 $\boldsymbol{\xi} = (\boldsymbol{z}, y) \in \mathbb{R}^d \times \mathbb{R}$ 和 $\phi_{\boldsymbol{x}}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle$. 因此 $f(\boldsymbol{x}, \boldsymbol{\xi}) = (\langle \boldsymbol{z}, \boldsymbol{x} \rangle - y)^2$.*

例 3.6. *在线性二值分类中, 损失为 hinge 损失, 有 $\boldsymbol{\xi} = (\boldsymbol{z}, y) \in \mathbb{R}^d \times \{-1, 1\}$ 和 $\phi_{\boldsymbol{x}}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle$. 因此 $f(\boldsymbol{x}, \boldsymbol{\xi}) = \max(1 - y\langle \boldsymbol{z}, \boldsymbol{x} \rangle, 0)$.*

例 3.7. *在带逻辑损失的神经网络的二分类中, 有 $\boldsymbol{\xi} = (\boldsymbol{x}, y) \in \mathbb{R}^d \times \{-1, 1\}$ 和 $\phi_{\boldsymbol{x}}$ 是权值 $\boldsymbol{x}$ 对应的网络. 因此, $f(\boldsymbol{x}, \boldsymbol{\xi}) = \ln(1 + \exp(-y\phi_{\boldsymbol{x}}(\boldsymbol{z})))$.*

上述问题的关键难点是我们不知道 ρ 分布. 因此, 完全解决这个问题是没有希望的. 相反, 我们感兴趣的是了解什么是我们能做的最好的, 如果我们可以从 ρ 访问 T 样本. 更详细地说, 我们想要设定超额风险的上限.

$$\mathrm{Risk}(\boldsymbol{x}_T) - \min_{\boldsymbol{x}} \mathrm{Risk}(\boldsymbol{x}),$$

$\boldsymbol{x}_T$ 是使用 T 样本学习的预测因子.

很明显, 这只是一个优化问题, 我们感兴趣的是次优性缺口的上界. 在这种观点下, 机器学习的目标可以被视为一个特定的优化问题.

附注 3.8. *请注意, 这不是解决学习问题的唯一方法. 事实上, 后悔最小化模型是学习的替代模型. 此外, 另一种方法是尝试估计 ρ 分布, 然后解决风险最小化问题, 这是统计中通常采用的方法. 没有一种方法是优于其他的, 每一种都有其优点和缺点.*

鉴于我们可以通过从它抽取的样本来获得 ρ 分布, 任何我们可能认为用来将风险最小化的方法在本质上都是随机的. 这意味着我们不能保证一个确定性的保证. 相反, 我们可以试着证明, 在很高的概率下, 我们的最小化过程将返回一个接近风险最小值的解. 同样直观的是, 我们能够保证的精度和概率必须取决于我们从 ρ 值中提取的样本的数量.

量化精度和失效概率对所使用样本数量的依赖关系是不可知论者可能近似正确 (PAC) 框架的目标, 其中关键字“不可知论”指的是我们对最佳预测器不做任何假设. 更多的细节, 给定一个精度参数 ϵ 和 δ 的故障概率, 我们感兴趣的是描述样本的复杂性假设 $\mathbb{F}$ 类被定义为样本 T 必要保证的概率至少为 $1-\delta$, 最好的学习算法使用假设类 $\mathbb{F}$ 输出解决方案 $\boldsymbol{x}_T$ 存在超额风险, 其界为 ϵ. 注意, 样本复杂度并不依赖于 ρ 值, 因此它是一个最坏情况的度量方法, 适用于所有可能的 22 个分布. 如果你认为我们对 ρ 分布一无所知, 这就说得通了, 所以如果你的保证对最坏的分布成立, 它对任何其他分布也成立. 数学上, 我们将说假设类是不可知论的, 可学习的是这样的样本复杂度函数存在.

定义 3.9 (Agnostic-PAC-learnable)**.** *我们会说, 一个函数类 $F=\{f(\boldsymbol{x},\cdot):\boldsymbol{x}\in\mathbb{R}^d\}$ 是* Agnostic-PAC-learnable *如果存在一个算法 $\mathcal{A}$ 和函数 $T(\epsilon,\delta)$, 比如当使用来自 ρ 的样本 $\mathcal{A}$ is used with $T\geq T(\epsilon,\delta)$ samples drawn from ρ, with probability at least $1-\delta$ the solution $\boldsymbol{x}_T$ returned by the algorithm has excess risk at most ϵ.*

请注意, 不可知论的 PAC 学习设置并没有说明我们应该遵循什么程序来发现这样的样本复杂性. 机器学习中最常用的解决学习问题的方法是所谓的经验风险最小化 (ERM) 问题. 它包括从 ρ 中提取 T 个样本. 和尽量减少经验风险:

$$\widehat{\mathrm{Risk}}(\boldsymbol{x}) := \min_{\boldsymbol{x}\in V}\frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x};\boldsymbol{\xi}_t)\ .$$

换句话说,ERM 就是将训练集上的误差最小化. 然而, 在许多有趣的情况下, 我们可以得出 $\mathrm{argmin}_{\boldsymbol{x}\in V}\ \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x};\boldsymbol{\xi}_t)$ 即使在无穷个样本下, 也可能与最优的 $\mathrm{argmin}_{\boldsymbol{x}\in V}\ \mathbb{E}[f(\boldsymbol{x};\boldsymbol{\xi})]$ 相去不远. 因此, 我们需要以某种方式修改 ERM 公式, 例如, 使用一个正则化项或 $\boldsymbol{x}$ 的贝叶斯先验, 或找到 ERM 起作用的条件.

ERM 方法非常普遍, 以至于机器学习本身常常被错误地认为是某种训练错误的最小化. 我们现在证明了 ERM 不是 ML 的整个世界, 证明了无后悔算法的存在, 这是一个带有次线性后悔的在线学习算法, 保证了不可知 pac 可学习性. 更详细地说, 我们将展示一种带有次线性后悔的在线算法可以用来解决机器学习问题. 这不仅仅是一个好奇, 例如, 这产生了计算效率高的无参数算法, 这可以通过 ERM 只运行两步程序来实现, 即运行 ERM 与不同的参数, 并在其中选择最佳的解决方案. 我们在讨论在线到批处理转换时已经提到了这种可能性, 但这次我们将加强它, 证明高概率的保证而不是预期的保证. 所以, 我们需要更多关于浓度不等式的信息.

3.2 关于浓度不等式的 Bits

我们将使用浓度不等式来证明高概率保证, 但我们需要超越随机变量的和. 特别地, 我们将使用鞅的概念.

定义 3.10 (熵)**.** *一个随机变量序列 $Z_1, Z_2, \ldots$ 被称为* **熵** *如果对于所有的 $t \geq 1$ 满足:*

$$\mathbb{E}[|Z_t|] < \infty, \qquad \mathbb{E}[Z_{t+1}|Z_1, \ldots, Z_t] = Z_t \ .$$

例 3.11. *考虑一个平整的硬币 c_t 和一种下注算法, 每一轮下注 $|x_t|$, 对于硬币来说相当于 $\mathrm{sign}(x_t)$. 我们赢或输的钱是 1:1, 所以在 t 轮次中总共我们赢的钱是 $Z_t = \sum_{i=1}^t c_i x_i$. $Z_1, \ldots, Z_t$, 它是一种熵. 事实上,*

$$\mathbb{E}[Z_t|Z_1, \ldots, Z_{t-1}] = \mathbb{E}[Z_{t-1} + x_t c_t|Z_1, \ldots, Z_{t-1}] = Z_{t-1} + \mathbb{E}[x_t c_t|Z_1, \ldots, Z_{t-1}] = 0 \ .$$

对于有界鞅, 我们可以证明有界的随机变量的高概率保证. 下面的定理是我们需要的关键结果.

定理 3.12 (Hoeffding-Azuma inequality)**.** *设 $Z_1, \ldots, Z_T$ 是 T 个随机变量的熵, 满足 $|Z_t - Z_{t+1}| \leq B, t = 1, \ldots, T-1$ 通常为真, 则*

$$\mathbb{P}[Z_T - Z_0 \geq \epsilon] \leq \exp\left(-\frac{\epsilon^2}{2B^2T}\right) \ .$$

同样的上界也成立 $\mathbb{P}[Z_0 - Z_T \geq \epsilon]$.

3.3 从后悔到不可知论的 PAC

我们现在展示之前介绍的在线批量转换如何为我们的机器学习问题提供高概率保证.

定理 3.13. *设 $V \subseteq \mathbb{R}^d$, $\mathrm{Risk}(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\xi})]$, 期望为 w.r.t. $\boldsymbol{\xi}$ drawn from ρ with support over some vector space D, and $f : V \times D \to [0, 1]$. Draw T samples $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_T$ i.i.d. from ρ and construct the sequence of losses $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{\xi}_t)$. Let $\mathcal{A}$ any online learning algorithm over the losses ℓ_t that outputs the sequence of predictions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1}$ and guarantees $\mathrm{Regret}_T(\boldsymbol{u}) \leq R(\boldsymbol{u}, T)$ for all $\boldsymbol{u} \in V$, for a function $R : V \times \mathbb{N} \to \mathbb{R}$. 概率至少为 $1 - \delta$, 以下成立*

$$\frac{1}{T}\sum_{t=1}^T \mathrm{Risk}(\boldsymbol{x}_t) \leq \min_{\boldsymbol{u} \in V} \ \mathrm{Risk}(\boldsymbol{u}) + \frac{R(\boldsymbol{u}, T)}{T} + 2\sqrt{\frac{2\ln\frac{2}{\delta}}{T}} \ .$$

证明. 定义 $Z_t = \sum_{i=1}^t (\mathrm{Risk}(\boldsymbol{x}_i) - \ell_i(\boldsymbol{x}_i))$. 我们声明 Z_t 是一个鞅. 实际上, 有

$$\mathbb{E}[\ell_t(\boldsymbol{x}_t)|\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{t-1}] = \mathbb{E}[f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)|\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{t-1}] = \mathrm{Risk}(\boldsymbol{x}_t),$$

事实上 $\boldsymbol{x}_t$ 仅仅取决于 $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{t-1}$ 因此, 有

$$\mathbb{E}[Z_{t+1}|Z_1, \ldots, Z_t] = Z_t + \mathbb{E}[\mathrm{Risk}(\boldsymbol{x}_{t+1}) - \ell_{t+1}(\boldsymbol{x}_{t+1})|Z_1, \ldots, Z_t] = Z_t,$$

得证因此, 由定理3.12, 有

$$\mathbb{P}\left[\sum_{t=1}^{T}(\mathrm{Risk}(\boldsymbol{x}_t)-\ell_t(\boldsymbol{x}_t))\geq\epsilon\right]=\mathbb{P}\left[Z_T-Z_0\geq\epsilon\right]\leq\exp\left(-\frac{\epsilon^2}{2T}\right)\ .$$

这表明概率至少为 $1-\delta/2$, 有

$$\sum_{t=1}^{T}\mathrm{Risk}(\boldsymbol{x}_t)\leq\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t)+\sqrt{2T\ln\frac{2}{\delta}}\ .$$

或者同样的

$$\frac{1}{T}\sum_{t=1}^{T}\mathrm{Risk}(\boldsymbol{x}_t)\leq\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t)+\sqrt{\frac{2\ln\frac{2}{\delta}}{T}}\ .$$

我们现在使用悔值 w.r.t 的定义. 任意 $\boldsymbol{u}$, 有

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t)=\frac{\mathrm{Regret}_T(\boldsymbol{u})}{T}+\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})\leq\frac{R(\boldsymbol{u},T)}{T}+\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})\ .$$

最后一步是建立风险为 $\mathrm{Risk}(\boldsymbol{u})$ 的高概率 $\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})$ 的上界. 这比之前的上界简单, 因为我们将 $\boldsymbol{u}$ 设定为使 V 中风险 $\mathrm{Risk}(\boldsymbol{x})+\frac{R(\boldsymbol{x},T)}{T}$ 最小化的固定的向量. 因此, $\ell_t(\boldsymbol{u})$ 是 i.i.d. 随机变量, $Z_t=\sum_{i=1}^{t}(\mathrm{Risk}(\boldsymbol{u})-\ell_i(\boldsymbol{u}))$ 形成一个鞅. 此, 根据上面的推理, 我们有至少 $1-\delta/2$ 概率.

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})\leq\mathrm{Risk}(\boldsymbol{u})+\sqrt{\frac{2\ln\frac{2}{\delta}}{T}}\ .$$

把它们放在一起, 使用联合边界, 我们就得到了指定边界. □

定理上界是 T 预测器的平均风险, 而我们感兴趣的是产生一个单一的预测器. 如果风险是凸函数, 且 V 是凸函数, 则可以将定理中不等式的 l.h.s. 下界, 并以 $\boldsymbol{x}_t$. 平均值来评估风险. 这是

$$\mathrm{Risk}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t\right)\leq\frac{1}{T}\sum_{t=1}^{T}\mathrm{Risk}(\boldsymbol{x}_t)\ .$$

如果风险不是凸函数, 我们需要一种方法来生成一个具有小风险的单一解. 一种可能是构造一个随机分类器, 以均匀概率对 $\boldsymbol{x}_t$ 中的一个进行采样, 并用它进行预测. 对于这个分类器, 我们立即有

$$\mathrm{Risk}(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T\})=\frac{1}{T}\sum_{t=1}^{T}\mathrm{Risk}(\boldsymbol{x}_t),$$

其中随机分类器的风险定义中的期望也是关于随机指数的.

在 T 预测因子中选择风险最小的, 这是可行的, 因为平均值的下界是最小值. 这很容易实现, 使用 $T/2$ 样本的在线学习过程和 $T/2$ 样本生成一个验证集, 以评估解决方案, 并选择最好的一个. 下面的定理表明, 在验证集上选择经验风险最小的预测器, 将使我们得到一个接近最佳预测器的高概率预测器.

定理 3.14. 我们有一组有限的预测因子 $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|S|}\}$, 以及一组从 ρ 上提取的 T 样本. 用 $\hat{\boldsymbol{x}} = \operatorname{argmin}_{\boldsymbol{x}\in S} \widehat{\operatorname{Risk}}(\boldsymbol{x})$ 表示, 则, 概率至少为 $1-\delta$, 有

$$\operatorname{Risk}(\hat{\boldsymbol{x}}) \leq \min_{\boldsymbol{x}\in S} \operatorname{Risk}(\boldsymbol{x}) + 2\sqrt{\frac{2\ln(2|S|/\delta)}{T}} \ .$$

证明. 我们想计算使验证误差最小化的假设与集合中最好的假设相去甚远的概率. 我们不能直接这样做, 因为我们没有使用浓度所需的独立性. 相反, 我们将上界存在至少一个经验风险远离风险的函数的概率. 所以, 我们有

$$\mathbb{P}\left[\exists \boldsymbol{x} \in S : |\operatorname{Risk}(\boldsymbol{x}) - \widehat{\operatorname{Risk}}(\boldsymbol{x})| > \frac{\epsilon}{2}\right] \leq \sum_{i=1}^{|S|} \mathbb{P}\left[|\operatorname{Risk}(\boldsymbol{x}_i) - \widehat{\operatorname{Risk}}(\boldsymbol{x}_i)| > \frac{\epsilon}{2}\right] \leq 2|S|\exp\left(-\frac{\epsilon^2 T}{8}\right) \ .$$

因此, 当概率至少为 $1-\delta$ 时, 我们得到

$$|\operatorname{Risk}(\boldsymbol{x}) - \widehat{\operatorname{Risk}}(\boldsymbol{x})| \leq \sqrt{\frac{8\ln(2|S|/\delta)}{T}}, \ \forall \boldsymbol{x} \in S,$$

我们能够确定 $\hat{\boldsymbol{x}}$ 的风险上界, 仅仅利用上面的事实, 也适用于 $\hat{\boldsymbol{x}}$. 定义 $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}\in S} \operatorname{Risk}(\boldsymbol{x})$, 有

$$\operatorname{Risk}(\hat{\boldsymbol{x}}) \leq \widehat{\operatorname{Risk}}(\hat{\boldsymbol{x}}) + \epsilon/2 \leq \widehat{\operatorname{Risk}}(\boldsymbol{x}^\star) + \epsilon/2 \leq \operatorname{Risk}(\boldsymbol{x}^\star) + \epsilon,$$

在上一个不等式中我们使用了 $\hat{\boldsymbol{x}}$ 使经验风险最小化的事实. □

使用该理论, 我们可以使用 $T/2$ 个样本进行训练, $T/2$ 个样本进行验证, 其中 $T \geq 2$. 在在线程序生成的 $T/2$ 样本中, 用 $\hat{x}_T$ 表示验证集上经验风险最好的预测器, 我们有至少 $1-2\delta$ 概率,

$$\operatorname{Risk}(\hat{\boldsymbol{x}}_T) \leq \min_{\boldsymbol{u}\in V} \operatorname{Risk}(\boldsymbol{u}) + \frac{2R(\boldsymbol{u}, T/2)}{T} + 8\sqrt{\frac{\ln(T/\delta)}{T}} \ .$$

值得注意的是, 以上三种方法中的任何一种, 在在线学习过程生成的 T 中选择一个 $\boldsymbol{x}_t$, 我们得到的样本复杂度保证与 ERM 得到的样本复杂度匹配, 直到多对数因子. 换句话说, 与统计学习的在线学习方法相比,ERM 没有什么特别之处. 此外,ERM 暗示了一个假设过程的存在, 它完美地减少了训练误差. 在现实中, 在对 ERM 进行分析时应考虑到优化误差. 另一方面, 在在线学习的方法中, 我们有一个直接的保证计算解. 另外重要的一点是, 上述保证并不意味着在线学习算法对于任何学习问题都存在次线性后悔. 它只是说, 如果它存在, 它也可以用于统计设置.

3.4 历史片段

定理3.1得具体形状是新的, 但如果它出现在文献中, 我也不会感到惊讶. 特别是, Cesa-Bianchi et al. [21]提出了均匀平均算法, 但 Blum et al. [17]提出了绝对损失平均法. 例3.3 的不均匀平均来自 Zhang [88], 即使没有明确提出它是在线到批处理的转换. 相反, 示例 4.12的不均匀平均来自 Lacoste-Julien et al. [47], 但同样没有提议将其作为在线到批处理转换. 用 OSD 和例4.12 的在线批量转换解决支持向量机问题的基本思想是 [78] 算法, 这是多年来最常用的支持向量机优化器.

Cutkosky [28]中引入了一种较新的在线批量转换方法, 该方法独立地重新发现并推广了 Nesterov and Shikhman [62]. 这种新方法允许证明最后迭代的收敛性, 而不是加权平均的收敛性, 在任何在线学习算法中都有微小的变化.

定理3.13 来自 Cesa-Bianchi et al. [21], 但在这里, 我用第二个集中度来说明竞争对手的真实风险, 而不是其经验风险. 定理3.14只不过是对具有有限基数的假设类的 ERM 的不可知论 PAC 学习保证. Cesa-Bianchi et al. [21]也给出了一个替代的过程, 在在线过程中生成的 T 中选择一个单一的假设, 在训练和验证中不需要分割数据. 然而, 得到的保证与我们已经证明的保证相匹配.

Chapter 4

超越 $\sqrt{T}$ 后悔值

4.1 强凸性和在线次梯度下降

现在让我们回到在线的凸优化理论. 第一章的例子告诉我们, 及时得到对数后悔是可能的. 然而, 我们看到在同一款游戏中, 我们只得到 $\sqrt{T}$-悔值的在线次梯度下降 (OSD). 原因是什么? 结果表明, 第一场比赛 $\ell_t(x) = (x - y_t)^2$ 在 $[0, 1]$, 不仅仅是由于 Lipschitz. 它们也有一些曲率, 可以利用来实现更好的后悔. 稍后我们将看到, 我们需要 OSD 的唯一更改是不同的学习速率, 这通常由后悔分析决定. 我们需要的关键概念是强凸性.

4.1.1 凸分析 bits: 强凸性

在这里, 我们引入了一个更强的凸性概念, 它允许为一个函数建立更好的下界. 我们将使用二次下界, 而不是利用次梯度实现的线性下界.

定义 4.1 (强凸函数)**.** *设 $\mu \geq 0$. 一个合适的函数 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 是凸集 $V \subseteq \operatorname{int} \operatorname{dom} f$ w.r.t. $\|\cdot\|$ 上的 μ-strongly convex , 若*

$$\forall \boldsymbol{x}, \boldsymbol{y} \in V, \boldsymbol{g} \in \partial f(\boldsymbol{x}), \quad f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\mu}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \ .$$

附注 4.2. *你可能会发现另一个强凸性的定义, 类似于凸性的定义. 然而, 很容易证明这两个定义是等价的.*

注意, 根据次梯度的定义, 任何凸函数都是 0-strongly convex, 同样任意 μ-strongly 凸函数对于任意的 $0 \leq \lambda \leq \mu$ 也是 λ-strongly convex.

换句话说, 上面的定义告诉我们强凸函数的下界可以是二次函数, 其中线性项是通过次梯度构造的通常项, 而二次项依赖于强凸. 因此, 我们仅仅使用凸性就有了一个更紧密的函数 w.r.t 的下界. 这是我们所期望的, 使用泰勒展开的二次可微凸函数和下边界的黑森的最小特征值. 确实, 我们有以下定理.

定理 4.3 ([74, Lemma 14])**.** *Let $V \subseteq \mathbb{R}^d$ convex and $f : V \to \mathbb{R}$ twice differentiable. Then, a sufficient condition for μ-strong convexity in V w.r.t. $\|\cdot\|$ is that for all $\boldsymbol{x}, \boldsymbol{y}$ we have $\langle \nabla^2 f(\boldsymbol{x})\boldsymbol{y}, \boldsymbol{y} \rangle \geq \mu \|\boldsymbol{y}\|^2$, where $\nabla^2 f(\boldsymbol{x})$ is the Hessian matrix of f at $\boldsymbol{x}$.*

然而, 这里有一个重要的区别, 我们不假定函数是二次可微的. 实际上, 我们甚至不需要纯微性. 因此, 使用子梯度意味着这个下界不必是唯一确定的, 如下一个示例所示.

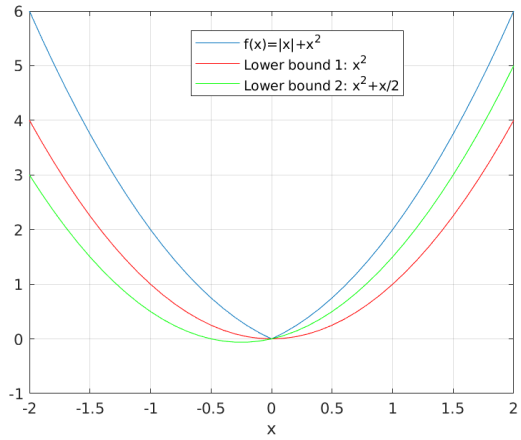

图 4.1: 强凸不可微函数 $f(x) = |x| + x^2$ 的可能下界

例 4.4. *考虑强凸函数 $f(x) = |x| + x^2$. 在图4.1, 我们展示了函数在 $x = 0$ 时的两个可能的二次下界.*

关于强凸函数和, 我们还有以下简单但有用的性质.

定理 4.5. *设 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 是非空凸集 $V \subseteq \operatorname{int}\operatorname{dom} f \cap \operatorname{int}\operatorname{dom} g$ w.r.t. $\|\cdot\|_2$ 上的 μ_1-强凸同时 $g : \mathbb{R}^d \to (-\infty, +\infty]$ 是一个 μ_2-强凸. 然后, $f + g$ 是 V w.r.t. $\|\cdot\|_2$. 上的 $\mu_1 + \mu_2$-强凸.*

证明. 注意, 关于 V 的假设告诉我们, 和的子微分集等于子微分集的和. 因此, 证明是直接从强凸的定义.

□

例 4.6. *设 $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$. 利用引理 4.3, 有 f 是在 $\mathbb{R}^d$ 上的 1-strongly convex w.r.t. $\|\cdot\|_2$.*

4.1.2 在线次梯度下降法求解强凸损失

定理 4.7. *设 V 是 $\mathbb{R}^d$ 上的一个非空闭凸集. 假设函数 $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ 是 μ_t-strongly convex w.r.t $\|\cdot\|_2$ over $V \subseteq \cap_{i=1}^T \operatorname{int}\operatorname{dom} \ell_i$, 其中 $\mu_t > 0$. 在算法2.2 使用 OSD, 其步长等于 $\eta_t = \frac{1}{\sum_{i=1}^t \mu_i}$. 则, 对于任意 $\boldsymbol{u} \in V$, 得到以下悔值保证.*

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{1}{2}\sum_{t=1}^T \eta_t \|\boldsymbol{g}_t\|_2^2 \ .$$

证明. 根据 ℓ_t 函数的 μ_t-strong convexity 假设, 有

$$\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \le \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle - \frac{\mu_t}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 \ .$$

根据 $\eta_t = \frac{1}{\sum_{i=1}^t \mu_i}$, 有

$$\frac{1}{2\eta_1} - \frac{\mu_1}{2} = 0,$$
$$\frac{1}{2\eta_t} - \frac{\mu_t}{2} = \frac{1}{2\eta_{t-1}}, t = 2, \cdots, T \ .$$

因此, 使用引理2.23 并从 $t = 1, \ldots, T$ 开始求和, 获得

$$\begin{aligned}\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\le \sum_{t=1}^T \left(\frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_t}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 - \frac{\mu_t}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|^2 + \frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \right) \\ &= -\frac{1}{2\eta_1}\|\boldsymbol{x}_2 - \boldsymbol{u}\|_2^2 + \sum_{t=2}^T \left(\frac{1}{2\eta_{t-1}}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_t}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 \right) + \sum_{t=1}^T \frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \ .\end{aligned}$$

注意到左边的第一个和是伸缩和, 我们就有了规定的界限. □

附注 4.8. *注意, 该定理需要一个有界域, 否则损失函数将不会是李普希茨, 因为它们也是强凸的.*

推论 4.9. *在定理4.7得假设下, 对于 $t = 1, \ldots, T$, 如果我们认为 $\mu_t = \mu > 0$ 和 ℓ_t 是 L-Lipschitz w.r.t. $\|\cdot\|_2$, 则*

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{L^2}{2\mu}(1 + \ln T) \ .$$

附注 4.10. *推论4.9 并不意味着对于任意的 T 悔值会小于使用学习效率 $\propto \frac{1}{L\sqrt{t}}$. 相反, 渐渐地, 在推论4.9 中的悔值总是比一个有 Lipschitz 损失的 OSD 好.*

例 4.11. *再次考虑第一个类中的例子: $\ell_t(x) = (x - y_t)^2$. 注意, 损失函数是 2-strongly convex w.r.t. $|\cdot|$. 因此, 设置 $\eta_t = \frac{1}{2t}$ 和 $\ell_t'(x) = 2(x - y_t)$ 给出了 $\ln(T) + 1$ 的悔值.*

现在让我们再次使用强凸随机问题的在线批量转换.

例 4.12. *如前所述, 我们可以使用在线到批处理转换来使用推论4.9来获得强凸随机函数的随机次梯度下降算法. 例如, 考虑经典的支持向量机目标*

$$\min_{\boldsymbol{x}} \ F(\boldsymbol{x}) := \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{N}\sum_{i=1}^N \max(1 - y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle, 0),$$

或任何其他正规化公式, 如正规化逻辑回归:

$$\min_{\boldsymbol{x}} \ F(\boldsymbol{x}) := \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{N}\sum_{i=1}^N \ln(1 + \exp(-y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle)),$$

对于 $\boldsymbol{z}_i \in \mathbb{R}^d$, $\|\boldsymbol{z}_i\|_2 \le R$, 和 $y_i \in \{-1, 1\}$. 首先, 注意到两个表达式的最小值必须是在 L_2 中, 半径与 $\sqrt{\frac{1}{\lambda}}$ 成比例, (证明作为练习). 因此, 我们设 V 与这个集合相等. 而后, 设 $\ell_t(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \max(1 - y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle, 0)$ 或者 $\ell_t(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \ln(1 + \exp(-y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle))$ 得到 λ-strongly convex 损失函数. 用推论4.9 和定理3.1 立即得到

$$\mathbb{E}\left[F\left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{x}_t\right)\right] - \min_{\boldsymbol{x}} \ F(\boldsymbol{x}) \le O\left(\frac{\ln T}{\lambda T}\right) \ .$$

但是, 我们可以做得更好. 其实, $\ell_t(\boldsymbol{x}) = \frac{\lambda t}{2}\|\boldsymbol{x}\|_2^2 + t\max(1 - y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle, 0)$ 或 $\ell_t(\boldsymbol{x}) = \frac{\lambda t}{2}\|\boldsymbol{x}\|_2^2 + t\ln(1 + \exp(-y_i\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle))$ 得到 λt-*strongly convex* 损失函数. 利用推论*4.9*, 我们得到 $\eta_t = \frac{2}{\lambda t(t+1)}$ 和定理*3.1* , 立即得到

$$\mathbb{E}\left[F\left(\frac{2}{T(T+1)}\sum_{t=1}^{T} t\,\boldsymbol{x}_t\right)\right] - \min_{\boldsymbol{x}}\ F(\boldsymbol{x}) \le O\left(\frac{1}{\lambda T}\right),$$

渐近地, 这是更好的因为它没有对数项.

4.2 自适应算法: $L^\star$ 边界和 AdaGrad

在本节中, 我们将进一步探讨在哪些条件下可以获得比 $O(DL\sqrt{T})$. 更好的后悔上界. 同时, 我们将自动获得这种改进的保证. 也就是说, 该算法将适应损失函数序列的特征, 而不必依赖于未来的信息.

4.2.1 在线次梯度下降的自适应学习速率

考虑最小化线性的后悔:

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle\ .$$

利用在线次梯度下降 (OSD), 我们说有界域的后悔可以为上界

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \le \frac{D^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T}\eta_t\|\boldsymbol{g}_t\|_2^2\ .$$

在固定的学习速率下, 使后悔上限最小化的学习速率是

$$\eta_T^\star = \frac{D}{\sqrt{\sum_{t=1}^{T}\|g_t\|_2^2}}\ .$$

不幸的是, 正如我们所说, 这种学习率不能使用, 因为它假定了未来回合的知识. 然而, 我们可能很幸运, 我们可以尝试在每一轮中使用时间 t 之前的知识来近似它, 也就是说, 我们可以尝试使用

$$\eta_t = \frac{D}{\sqrt{\sum_{i=1}^{t}\|\boldsymbol{g}_i\|_2^2}}\ . \tag{4.1}$$

观察 $\eta_T = \eta_T^\star$, 所以, 第一阶段的后悔正是我们所需要的! 对于另一项, 最优学习率会给我们

$$\frac{1}{2}\sum_{t=1}^{T}\eta_T^\star\|\boldsymbol{g}_t\|_2^2 = \frac{1}{2}D\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2}\ .$$

现在我们看看我们的近似得到了什么.

$$\frac{1}{2}\sum_{t=1}^{T}\eta_t\|\boldsymbol{g}_t\|^2 = \frac{1}{2}D\sum_{t=1}^{T}\frac{\|\boldsymbol{g}_t\|_2^2}{\sqrt{\sum_{i=1}^{t}\|\boldsymbol{g}_i\|_2^2}}\ .$$

我们需要找到上界的方法. 处理这些和的方法, 就像我们在其他例子中做的一样, 是用积分来近似它们. 所以, 我们可以使用下面这个非常有用的引理, 它概括了很多类似的特定引理

引理 4.13. 设 $a_0 \geq 0$ 和 $f : [0, +\infty) \to [0, +\infty)$ 是一个非增函数. 则

$$\sum_{t=1}^{T} a_t f\left(a_0 + \sum_{i=1}^{t} a_i\right) \leq \int_{a_0}^{\sum_{t=0}^{T} a_t} f(x)dx \ .$$

证明. 用 $s_t = \sum_{i=0}^{t} a_i$ 表示.

$$a_t f\left(a_0 + \sum_{i=1}^{t} a_i\right) = a_t f(s_t) = \int_{s_{t-1}}^{s_t} f(s_t)dx \leq \int_{s_{t-1}}^{s_t} f(x)dx \ .$$

对 $t = 1, \cdots, T$ 求和, 我们有一个定界. □

用这个引理, 我们得到了这个

$$\frac{1}{2} D \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_2^2}{\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|_2^2}} \leq D \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} \ .$$

令人惊讶的是, 这个术语只比我们从 $\eta_T^\star$ 的最佳选择中得到的糟糕 2 倍. 然而, 这个学习率可以在不了解未来的情况下计算出来, 而且它确实可以使用! 总的来说, 我们得到了这个选择

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{3}{2} D \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} \ . \tag{4.2}$$

请注意, 通过将学习速率乘以 $\sqrt{2}$ 有可能将边界前面的常数提高到 $\frac{\sqrt{2}}{2}$. 综上所述, 我们得到如下定理.

定理 4.14. 设 $V \subseteq \mathbb{R}^d$ 是一个直径为 D 的非空闭凸集, *i.e.* $\max_{\boldsymbol{x},\boldsymbol{y} \in V} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq D$. 设 $\ell_1, \cdots, \ell_T$ 是一个任意凸函数序列, 当 $t = 1, \ldots, T$ 时, $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ 在包含 V 的开集上子可微. 任选 $\boldsymbol{x}_1 \in V$ 和 $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|_2^2}}$, $t = 1, \ldots, T$. 则 $\forall \boldsymbol{u} \in V$, 以下悔值边界成立.

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \sqrt{2} D \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} = \sqrt{2} \min_{\eta > 0} \ \frac{D^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2 \ .$$

定理中的第二个等式清楚地显示了这种学习率的优势: 我们获得了 (几乎) 同样的保证, 我们可以知道未来的梯度! 这本身就是一个有趣的结果: 它提供了一种原则性的方法, 以几乎最优的保证来设置学习率. 然而, 这种简单的后悔还会带来其他后果. 首先, 我们将这个结果专门化到损失是平滑的情况.

4.2.2 凸分析位: 平滑函数

我们现在考虑一类具有下界为次梯度的平方范数的损失函数. 我们还将引入双规范的概念. 虽然这一主题并不需要双重规范, 但它们提供了更多的通用性, 同时也让我可以慢慢地介绍一些在线镜面下降章节所需要的概念

定义 4.15 (双重标准)**.** $\|\cdot\|$ 标准的**双重标准** $\|\cdot\|_\star$ 被定义为 $\|\boldsymbol{\theta}\|_\star = \max_{\boldsymbol{x}:\|\boldsymbol{x}\| \leq 1} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle$.

例 4.16. L_2 标准的双重标准是 L_2 标准. 其实 $\|\boldsymbol{\theta}\|_\star = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\le 1} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle \le \|\boldsymbol{\theta}\|_2$ *by Cauchy-Schwarz inequality. Also,* 设 $\boldsymbol{v} = \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2}$, 故 $\max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\le 1} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle \ge \langle \boldsymbol{\theta}, \boldsymbol{v}\rangle = \|\boldsymbol{\theta}\|_2$.

如果你不知道算子范数的概念, 对偶范数的概念一开始可能会有点奇怪. 理解它的一种方法是, 它是一种衡量线性泛函有多大的方法. 例如, 考虑线性函数 $f(\boldsymbol{x}) = \langle \boldsymbol{z}, \boldsymbol{x}\rangle$, 我们想尝试理解它有多大. 因此, 我们可以测量 $\max_{\boldsymbol{x}\neq 0} \frac{\langle \boldsymbol{z},\boldsymbol{x}\rangle}{\|\boldsymbol{x}\|}$ 也就是说, 我们测量线性函数的输出与输入 x 相比有多大, 其中 x 是用一些范数测量的. 现在, 你可以证明上面的等价于 $\boldsymbol{z}$ 的对偶范数.

附注 4.17. 对偶范数的定义直接暗示了 $\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle \le \|\boldsymbol{\theta}\|_\star \|\boldsymbol{x}\|$.

现在我们可以引入光滑函数, 使用上面定义的对偶范数.

定义 4.18 (平滑函数)**.** 设 $f : V \to \mathbb{R}$ 可微. 我们 f 是 M***-smooth*** *w.r.t.* $\|\cdot\|$ *if* $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_\star \le M\|\boldsymbol{x} - \boldsymbol{y}\|$ 对于所有的 $\boldsymbol{x}, \boldsymbol{y} \in V$.

记住上面的直觉在双重标准, 采取一个梯度的双重标准是有道理的, 如果你将每个梯度与线性泛函 $\langle \nabla f(\boldsymbol{y}), \boldsymbol{x}\rangle$, 是需要创建一个线性近似 f.

光滑函数有很多属性, 例如一个光滑函数可以由二次的上界. 但是, 在下面我们将需要以下属性.

定理 4.19 ([e.g., Lemma 4.1 79])**.** 设 $f : \mathbb{R}^d \to \mathbb{R}$ 是 M*-smooth and* 遵从以下界限, 则对于所有的 $\boldsymbol{x} \in \mathbb{R}^d$

$$\|\nabla f(\boldsymbol{x})\|_\star^2 \le 2M(f(\boldsymbol{x}) - \inf_{\boldsymbol{y}\in\mathbb{R}^d} f(\boldsymbol{y})) \ .$$

4.2.3 $L^\star$ 界

现在假设损失函数 $\ell_1, \ldots, \ell_T$ 遵从以下界限, 且 M 是平滑的. 在不丧失一般性的情况下, 我们可以假设它们中的每一个从下到上都有 0 的边界. 在这些假设下, 由 (4.2) 中的悔值和定理4.19 我们立即得到

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le 2D\sqrt{M\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)} \ .$$

这是一个隐边界, 让 $\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)$ 出现在不等式的两边. 为了使其显式, 我们将使用以下简单引理 (剩下的证明作为练习).

引理 4.20. 设 $a, c > 0$, $b \ge 0$, 以及 $x \ge 0$ 即 $x - \sqrt{ax+b} \le c$. 那么 $x \le a + c + 2\sqrt{b + ac}$.

故, 我们有以下定理.

定理 4.21. 设 $V \subseteq \mathbb{R}^d$ 是一个直径为 D 的非空闭凸集, *i.e.* $\max_{\boldsymbol{x},\boldsymbol{y}\in V} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \le D$. 设 $\ell_1, \cdots, \ell_T$ 是任意的非负凸函数序列, $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ M*-smooth* 在包含 V 的开集上, 对于 $t = 1, \ldots, T$. 任意 $\boldsymbol{x}_1 \in V$ 和 $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^t \|\boldsymbol{g}_i\|_2^2}}$, $t = 1, \ldots, T$. 则, $\forall \boldsymbol{u} \in V$, 以下悔值边界成立

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le 4MD^2 + 4D\sqrt{M\sum_{t=1}^T \ell_t(\boldsymbol{u})} \ .$$

这种后悔保证非常有趣, 因为在最坏的情况下, 它仍然是 $O(\sqrt{T})$, 但在最好的情况下, 它变成了一个常数! 事实上, 如果存在一个 $\boldsymbol{u} \in V$ 使 $\sum_{t=1}^{T} \ell_t(\boldsymbol{u}) = 0$ 我们就会一直感到后悔. 基本上, 如果损失是"容易的", 算法会适应这种情况, 给我们一个更好的后悔. 这些类型的保证被称为 $L^\star$ 因为它们依赖于可以用 $L^\star$ 表示的竞争对手的累积损失.

4.2.4 AdaGrad

现在我们给出(4.2)中悔值约束的另一个应用. **AdaGrad**, 也就是自适应梯度, 是 McMahan and Streeter [58] 和 Duchi et al. [32]等人独立提出的在线凸优化算法. 它的目的是适应梯度序列. 它通常被认为是一种随机优化算法, 但实际上它被提出用于在线凸优化 (OCO). 要将它用作一种随机算法, 您应该使用在线到批处理的转换, 否则就不能保证收敛.

我们将给出一个只允许超矩形为可行集 V 的证明, 另一方面, 这个限制使得证明几乎是微不足道的. 让我们看看它是如何工作的.

AdaGrad 关键因素:

- 一个协调的学习过程;
- (4.1)中的自适应学习率.

对于第一个因素, 如前所述, 任何 OCO 问题的后悔都可能是在线线性优化 (OLO) 问题的后悔的上限. 也就是说,

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \ .$$

现在, 关键的观察是明确地把内积写成在单个坐标上的乘积的和:

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle = \sum_{t=1}^{T} \sum_{i=1}^{d} g_{t,i} x_{t,i} - \sum_{t=1}^{T} \sum_{i=1}^{d} g_{t,i} u_i = \sum_{i=1}^{d} \left(\sum_{t=1}^{T} g_{t,i} x_{t,i} - \sum_{t=1}^{T} g_{t,i} u_i \right) = \sum_{i=1}^{d} \mathrm{Regret}_{T,i}(u_i),$$

其中我们用 $\mathrm{Regret}_{T,i}(u_i)$ 表示坐标 i 上的一维 OLO 问题的悔值, 即 $\sum_{t=1}^{T} g_{t,i} x_{t,i} - \sum_{t=1}^{T} g_{t,i} u_i$. 换言之, 我们可以将原始后悔分解为 d OLO 悔值最小化问题的总和, 我们可以试着分别关注它们中的每一个.

一维问题的一个很好的候选点是具有 (4.1)中的学习速率的 OSD. 我们可以将 (4.2) 中的后悔专门化为一维情况下的线性损失, 因此我们得到每个坐标 i.

$$\sum_{t=1}^{T} g_{t,i} x_{t,i} - \sum_{t=1}^{T} g_{t,i} u_i \leq \sqrt{2} D_i \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \ .$$

这个选择给出了算法4.1算法, 综上所述, 我们立即作出以下后悔保证.

定理 4.22. 设 $V = \{\boldsymbol{x} : a_i \leq x_i \leq b_i\}$ 每个坐标上的直径等于 $D_i = b_i - a_i$. 设 $\ell_1, \cdots, \ell_T$ 是任意凸函数序列 $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ 在包含 V 的开集上次可微, 对于 $t = 1, \ldots, T$. 任选 $\boldsymbol{x}_1 \in V$ 和

Algorithm 4.1 AdaGrad for Hyperrectangles

Require: $V = \{\boldsymbol{x} : a_i \le x_i \le b_i\}$, $\boldsymbol{x}_1 \in V$

for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t$
  Receive $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  Set $\eta_{t,i} = \frac{\sqrt{2}D_i}{2\sqrt{\sum_{j=1}^t g_{j,i}^2}}$
  $\boldsymbol{x}_{t+1,i} = \max(\min(x_{t,i} - \eta_{t,i} g_{t,i}, b_i), a_i),\ i = 1, \ldots, d$
end for

$\eta_{t,i} = \frac{\sqrt{2}D_i}{2\sqrt{\sum_{j=1}^t g_{j,i}^2}}$, $t = 1, \ldots, T$. 有, $\forall \boldsymbol{u} \in V$, 以下悔值边界成立

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \sqrt{2} \sum_{i=1}^d D_i \sqrt{\sum_{t=1}^T g_{t,i}^2}\ .$$

与定理4.14相比, 这是一个更好的后悔界吗? 视情况而定! 为了比较两者, 我们必须考虑 V 是一个超矩形, 因为上面的 AdaGrad 分析只适用于超矩形. 然后, 我们要比较

$$D\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2} \qquad \text{versus} \qquad \sum_{i=1}^d D_i \sqrt{\sum_{t=1}^T g_{t,i}^2}\ .$$

根据 Cauchy-Schwartz, 得到 $\sum_{i=1}^d D_i \sqrt{\sum_{t=1}^T g_{t,i}^2} \le D\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2}$. 故, 假设相同的次梯度序列,AdaGrad 在超矩形上总是有更好的后悔. 另外, 请注意,

$$\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2} \le \sum_{i=1}^d \sqrt{\sum_{t=1}^T g_{t,i}^2} \le \sqrt{d}\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2}$$

因此, 在 V 是超立方的情况下, 有 $D_i = D_\infty = \max_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$ 和 $D = \sqrt{d}D_\infty$, AdaGrad 的界在 $1/\sqrt{d}$ 和 1 乘以定理4.14得界. 换句话说, 如果我们幸运地有了子梯度, 那么特定的定义域形状可以在保证中为我们节省 $\sqrt{d}$ 的因子.

注意, 对 AdaGrad 的更一般的分析允许考虑任意域, 但它不会改变最适合 AdaGrad 的域是超矩形的一般信息. 我们将在介绍在线镜像下降时, 探讨基于可行集 V 的形状选择在线算法的问题. 在保证中隐藏的是 AdaGrad 最大的优点是坐标无标度的特性. 也就是说, 如果梯度的每个坐标都乘以不同的常数, 学习速率就会自动将它们缩小. 最优解 $\boldsymbol{u}$ 也会相应地缩放, 但可行集的固定直径掩盖了这一点. 在梯度的坐标范围不同的情况下, 这可能是有用的. 事实上, 这在深度神经网络的随机优化中确实发生了, 第一层的梯度比最后一层的梯度更小.

4.3 历史片段

Hazan et al. [39]开创性论文首次显示了推论4.9 . 定理4.7得一般性证明被 Hazan et al. [40]所证明.

(4.1)中的自适应学习率最早出现在 Streeter and McMahan [82]. 然而, 类似的方法在很久以前就被使用了. 实际上, 关键观察近似甲骨文数量估计到时间 t 是自信算法 [9] 的首次提出, 在学习速率的平方根成反比的累计损失算法, 和光滑的损失意味着 $L^\star$ 边界与定理4.21相似.

AdaGrad 是由 McMahan and Streeter [58] 和 Duchi et al. [32]这两个小组在同一次会议上以基本相同的形式独立提出的. 这里提出的分析是 Streeter and McMahan [82], 它不处理一般可行集, 也不支持“全矩阵”, 即全矩阵学习率而不是对角矩阵. 然而, 在机器学习应用程序中,AdaGrad 很少与投影步骤一起使用 (即使可以证明这样做会破坏最坏的性能 [66]). 此外, 在对抗性设置中, 与对角线矩阵相比, 全矩阵似乎不提供后悔方面的优势.

注意,AdaGrad 学习速率通常写成

$$\eta_{t,i} = \frac{D_i}{\epsilon + \sqrt{\sum_{j=1}^{t} g_{j,i}^2}},$$

$\epsilon > 0$ 一个小常数, 用来防止被 0 除, 事实上, 正如 Orabona and Pál [66]所强调的那样. ϵ 并不是必需的, 移除它使得更新变得无尺度.

AdaGrad 激发了大量的克隆游戏, 其中大多数都带有相似的、更糟糕的或不后悔的内容. 关键词“适应性”本身已经随着时间的推移改变了它的含义. 它用来表示算法获得与预先知道数据的特定属性相同的保证的能力 (即适应梯度/噪声/尺度 =(几乎) 与预先知道梯度/噪声/尺度相同的性能). 实际上, 在统计学中, 这个关键字的含义是一样的. 然而, 随着时间的推移,“适应性”的含义已经发生了变化. 如今, 它似乎指的是任何一种不能保证任何特定情况的协调学习速度.

Chapter 5

在线线性优化的下界

在本章中, 我们将介绍在线线性优化 (OLO) 的一些下界. 记住线性损失是凸的, 这立即给了我们在线凸优化 (OCO) 的下界. 我们将同时考虑有约束和无约束的情况. 下界很重要, 因为它们告诉我们什么是最优算法, 以及我们的知识中有哪些空白.

5.1 有界 OLO 的下界

我们将首先考虑有界约束的情况. 找到一个下界意味着为对手找到一种策略, 不管算法做什么, 它都会给算法带来一定的后悔. 我们将使用概率方法来构造我们的下界. 基本方法依赖于这样一个事实: 对于任意带 V 域的随机变量 $\boldsymbol{z}$ 和任意函数 f.

$$\sup_{\boldsymbol{x}\in V} f(\boldsymbol{x}) \geq \mathbb{E}[f(\boldsymbol{z})] \ .$$

对我们来说, 这意味着我们可以将最坏情况下的函数序列的效果的下界与函数的任意分布的期望相结合. 如果明智地选择分配, 我们可以预期不平等的差距会很小. 为什么你依赖于预期而不是构建一个对抗的序列? 因为使用随机损失函数可以很容易地处理任意算法. 特别地, 我们将选择随机损失函数, 使算法的期望损失为 0, 独立于算法的策略.

定理 5.1. *设 $V \subset \mathbb{R}^d$ 是任意的非空有界闭凸子集. 设 $D = \sup_{\boldsymbol{v},\boldsymbol{w}\in V} \|\boldsymbol{v}-\boldsymbol{w}\|_2$ 是 V 的直径. 设 $\mathcal{A}$ 是 V 上的任意可能的 OLO 算法. 设 T 任意的非负整数. 这里有一个向量序列 $\boldsymbol{g}_1,\ldots,\boldsymbol{g}_T$ 伴随 $\|\boldsymbol{g}_t\|_2 \leq L$ 和 $\boldsymbol{u}\in V$ 得到算法 $\mathcal{A}$ 的悔值满足*

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \geq \frac{\sqrt{2}LD\sqrt{T}}{4} \ .$$

证明. 有表达式 $\mathrm{Regret}_T = \max_{\boldsymbol{u}\in V}\mathrm{Regret}_T(\boldsymbol{u})$. 设 $\boldsymbol{v},\boldsymbol{w}\in V$ 得到 $\|\boldsymbol{v}-\boldsymbol{w}\|_2 = D$. 设 $\boldsymbol{z} = \frac{\boldsymbol{v}-\boldsymbol{w}}{\|\boldsymbol{v}-\boldsymbol{w}\|_2}$, 故 $\langle \boldsymbol{z}, \boldsymbol{v}-\boldsymbol{w}\rangle = D$. 设 $\epsilon_1,\ldots,\epsilon_T$ 为 Rademacher 随机变量, 为 $\mathbb{P}[\epsilon_t = 1] = \mathbb{P}[\epsilon_t = -1] = 1/2$ 设损失为

$\boldsymbol{g}_t = L\epsilon_t \boldsymbol{z}$.

$$\begin{aligned}
\sup_{\boldsymbol{g}_1,\dots,\boldsymbol{g}_T} \text{Regret}_T &\geq \mathbb{E}\left[\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{u}\in V}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] = \mathbb{E}\left[-\min_{\boldsymbol{u}\in V}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] \\
&= \mathbb{E}\left[\max_{\boldsymbol{u}\in V}\sum_{t=1}^T -L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] = \mathbb{E}\left[\max_{\boldsymbol{u}\in V}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] \\
&= \mathbb{E}\left[\max_{\boldsymbol{u}\in \{\boldsymbol{v},\boldsymbol{w}\}}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] = \mathbb{E}\left[\frac{1}{2}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{v}+\boldsymbol{w}\rangle + \frac{1}{2}\left|\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{v}-\boldsymbol{w}\rangle\right|\right] \\
&= \frac{L}{2}\mathbb{E}\left[\left|\sum_{t=1}^T \epsilon_t\langle \boldsymbol{z}, \boldsymbol{v}-\boldsymbol{w}\rangle\right|\right] = \frac{LD}{2}\mathbb{E}\left[\left|\sum_{t=1}^T \epsilon_t\right|\right] \geq \frac{\sqrt{2}LD\sqrt{T}}{4} .
\end{aligned}$$

我们让 $\mathbb{E}[\epsilon_t] = 0$ 和 ϵ_t 在第一个等式中不相关, $\max(a,b) = \frac{a+b}{2} + \frac{|a-b|}{2}$ 在第四个等式中不相关, Khintchine 和 ϵ_t 在最后一个不等式中不相等. □

对于学习率 $\eta_t = \frac{D}{L\sqrt{t}}$ or $\eta = \frac{D}{L\sqrt{T}}$, 的在线次梯度下降 (OSD), 我们从其上界证明了下界是一个常数乘因子. 这意味着在两种学习速率设置下 OSD 都是渐近最佳的.

在这一点上, 有一个重要的考虑要做: 当我们设法证明更好的后悔时, 这怎么可能是最佳的后悔, 例如在自适应学习率? 微妙之处在于, 限制对手 L-Lipschitz 损失, 对手总是可以迫使算法至少是定理5.1. 中的后悔. 然而, 我们可以设计算法来利用对手的次优策略. 事实上, 例如, 如果对手的游戏方式是所有的次梯度都有相同的范数等于 L, 那么就没有什么需要适应的了!

5.2 无约束在线优化

现在我们转移到无约束的情况, 然而, 首先我们必须丰富我们的数学工具箱与另一个基本工具, *Fenchel conjugates*.

5.2.1 凸分析 Bits: Fenchel Conjugate

定义 5.2 (闭函数 n)**.** *函数 $f : V \subseteq \mathbb{R}^d \to [-\infty, +\infty]$ 是闭的若 $\{\boldsymbol{x} : f(\boldsymbol{x}) \leq \alpha\}$ 对于任意的 $\alpha \in \mathbb{R}$ 是闭的.*

注意在 Hausdorff 空间一个函数是闭的若它是 lower semicontinuos [11, Lemma 1.24].

例 5.3. *集合 $V \subset \mathbb{R}^d$ 的指标函数, 是闭的如果 V 是闭的.*

定义 5.4 (Fenchel Conjugate)**.** *对于函数 $f : \mathbb{R}^d \to [-\infty, \infty]$, 定义 **Fenchel conjugate** $f^\star : \mathbb{R}^d \to [-\infty, \infty]$ 是*

$$f^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}\in\mathbb{R}^d} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - f(\boldsymbol{x}) .$$

根据这个定义我们立即得到 Fenchel-Young 不等式

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle \leq f(\boldsymbol{x}) + f^\star(\boldsymbol{\theta}),\ \forall \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^d .$$

我们有以下 Fenchel conjugate 的有用的性质.

定理 5.5 ([72, Corollary 23.5.1 and Theorem 23.5])**.** 设 $f:\mathbb{R}^d \to (-\infty,+\infty]$ 凸的, 适当的, 闭的. 则

1. $\boldsymbol{x} \in \partial f^\star(\boldsymbol{\theta})$ *iff* $\boldsymbol{\theta} \in \partial f(\boldsymbol{x})$.

2. $\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - f(\boldsymbol{x})$ 在 $\boldsymbol{x}$ 的 $\boldsymbol{x} = \boldsymbol{x}^\star$ 上达到它的上确界, *iff* $\boldsymbol{x}^\star \in \partial f^\star(\boldsymbol{\theta})$.

例 5.6. 设 $f(x) = \exp(x)$, 因此有 $f^\star(\theta) = \max_x\ x\theta - \exp(x)$. 解决最优化问题, 对于 $\theta < 0$, 我们得到 $x^\star = \ln(\theta)$ 如果 $\theta > 0$ 和 $f^\star(\theta) = \theta\ln\theta - \theta$, $f^\star(0) = 0$, 和 $f^\star(\theta) = +\infty$.

例 5.7. 考虑函数 $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2$, $\|\cdot\|$ 是 $\mathbb{R}^d$ 上的一个标准, $\|\cdot\|_\star$ 也是一个标准. 我们可以证明它的共轭是 $f^\star(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$. 从

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x}\|^2 \le \|\boldsymbol{\theta}\|_\star\|\boldsymbol{x}\| - \frac{1}{2}\|\boldsymbol{x}\|^2$$

对于所有的 $\boldsymbol{x}$. 右边是一个 $\|\boldsymbol{x}\|$ 二次函数, 它有最大值 $\frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$. 因此对于所有的 $\boldsymbol{x}$, 我们有

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x}\|^2 \le \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2,$$

证明了 $f^\star(\boldsymbol{\theta}) \le \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$. 为了证明其他不等式, 设 $\boldsymbol{x}$ 是 $\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle = \|\boldsymbol{\theta}\|_\star\|\boldsymbol{x}\|$ 任意向量, 等比例缩放有 $\|\boldsymbol{x}\| = \|\boldsymbol{\theta}\|_\star$. 对于 $\boldsymbol{x}$, 我们有

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x}\|^2 = \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2,$$

证明了 $f^\star(\boldsymbol{\theta}) \ge \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$.

引理 5.8. 设 f 是一个函数同时设 $f^\star$ 是它的 *Fenchel* 共轭. 对于 $a > 0$ 和 $b \in \mathbb{R}$, $g(\boldsymbol{x}) = af(\boldsymbol{x}) + b$ 的 *Fenchel* 共轭是 $g^\star(\boldsymbol{\theta}) = af^\star(\boldsymbol{\theta}/a) - b$.

证明. 根据共轭函数的定义, 我们有

$$g^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}\in\mathbb{R}^d} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - af(x) - b = -b + a\sup_{\boldsymbol{x}\in\mathbb{R}^d}\left\langle \frac{\boldsymbol{\theta}}{a}, \boldsymbol{x}\right\rangle - f(x) = -b + af^\star\left(\frac{\boldsymbol{\theta}}{a}\right) . \qquad \square$$

引理 5.9 ([11, Example 13.7])**.** 设 $f:\mathbb{R}\to(-\infty,+\infty]$ 是偶函数. 那么 $(f\circ\|\cdot\|_2)^\star = f^\star\circ\|\cdot\|_2$.

5.2.2 无约束情况下的下界

上面的下界只适用于有约束的设置. 在无约束条件下, 我们证明了对于任意 $\boldsymbol{u}\in\mathbb{R}^d$, 当 $\boldsymbol{x}_1 = \boldsymbol{0}$ 并且 $\eta = \frac{1}{L\sqrt{T}}$ 的学习速率为常值, 则 OSD 的悔值为 $\frac{1}{2}L(\|\boldsymbol{u}\|_2^2 + 1)\sqrt{T}$. 这是最优的吗？很明显悔值在 $\|\boldsymbol{u}\|_2$ 上至少是线性的, 但线性就足够了吗?

我将遵循的方法是将 OLO 游戏简化为对硬币下注的在线游戏, 其中的下限是已知的. 接下来, 让我们来介绍这个投币在线游戏:

- 一开始的初始金额为 $\epsilon > 0$.
- 在每一轮中, 该算法将其当前财富的一小部分押注于一枚硬币的结果.
- 硬币的结果会被揭示, 算法会以 1 比 1 赢或输.

这个网络游戏的目的是赢取尽可能多的钱. 同样, 在我们考虑的所有在线游戏中, 我们不会对硬币的结果做出任何假设. 注意, 这个游戏也可以使用日志丢失写成 OCO.

我们将会用 $c_t \in \{-1,1\}, t=1,\dots,T$ 来表示硬币的结果. 我们将用 $\beta_t \in [-1,1]$ 的绝对值来表示要下注的钱的分数, 用它的符号来表示我们下注的一方. 算法从游戏开始到第 t 轮结束, 所赢的的钱将用 r_t 表示, 如果赢钱和输钱的比率是 1: 1, 则

$$\overbrace{r_t+\epsilon}^{\text{Money at the end of round } t} = \overbrace{r_{t-1}+\epsilon}^{\text{Money at the beginning of round } t} + \overbrace{c_t\beta_t(r_{t-1}+\epsilon)}^{\text{Money won or lost}} = \epsilon\prod_{i=1}^{t}(1+\beta_i c_i),$$

我们利用事实 $r_0=0$. 同样我们用 $x_t=\beta_t(\epsilon+r_{t-1})$ 来表示算法对第 t 轮次的下注.

如果每轮的结果都正确的话, 我们就能把每轮的钱翻一倍, 这样 $\epsilon+r_T=\epsilon 2^T$. 然而, 考虑到游戏的对抗性, 我们实际上可以证明, 我们可以获得最大财富的一个更强的下限.

定理 5.10 ([20, 一个简化的定理 9.2])**.** *设 $T\geq 21$. 则对于任意初始筹码 $\epsilon>0$ 的下注策略, 即押注其当前财富的一小部分, 存在一系列硬币结果 c_t, 例如*

$$\epsilon+\sum_{t=1}^{T}c_t x_t \leq \frac{1}{\sqrt{T}}\max_{-1\leq\beta\leq 1}\ \epsilon\prod_{t=1}^{T}(1+\beta c_t) \leq \frac{\epsilon}{\sqrt{T}}\exp\left(\frac{\ln 2(\sum_{t=1}^{T}c_t)^2}{T}\right)\ .$$

现在, 让我们将投币游戏与 OLO 连接起来. 记住, 在 OLO 中证明后悔保证包括显示

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t,\boldsymbol{u}\rangle \leq \psi_T(\boldsymbol{u}),\ \forall \boldsymbol{u}\in\mathbb{R}^d,$$

对于 ψ_T 函数, 我们希望它对 T 的依赖是次线性的. 利用我们学过的芬切尔共轭的新概念, 这个等价于证明.

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle \leq \inf_{\boldsymbol{u}}\ \sum_{t=1}^{T}\langle \boldsymbol{g}_t,\boldsymbol{u}\rangle+\psi_T(\boldsymbol{u}) = -\sup_{\boldsymbol{u}}\ -\left\langle \sum_{t=1}^{T}\boldsymbol{g}_t,\boldsymbol{u}\right\rangle - \psi_T(\boldsymbol{u}) = -\psi_T^{\star}\left(-\sum_{t=1}^{T}\boldsymbol{g}_t\right)\ .$$

因此, 对于一个给定的在线算法我们可以证明后悔范围证明存在一个函数 ψ_T 或者等同于找到它的共轭 $\psi_T^{\star}$. 同样, 证明一个下界在无约束 OLO 条件下意味着找到一个序列 $\boldsymbol{g}_t$ 以及一个函数 ψ_T 或者一个可以分别降低悔值及累计损失算法的下界函数 $\psi_T^{\star}$. 在没有其他信息的情况下, 要猜出什么是增长最慢的函数 ψ_T. 是很有挑战性的. 因此, 我们把注意力限制在在线算法上, 这些算法保证了对零向量的持续后悔. 这立即暗示了以下重要的结果.

定理 5.11. *设 $\epsilon(t)$ 是一个每轮中指数非递减函数, $\mathcal{A}$ 是一个 OLO 算法, 该算法保证任意序列 $\boldsymbol{g}_1,\dots,\boldsymbol{g}_t\in\mathbb{R}^d$ 在 $\|\boldsymbol{g}_i\|_2\leq 1$ 时, 满足 $\mathrm{Regret}_t(\boldsymbol{0})\leq\epsilon(t)$. 则对于 $t=1,\dots,T$, 存在 β_t 使得 $\boldsymbol{x}_t=\boldsymbol{\beta}_t(\epsilon(T)-\sum_{i=1}^{t-1}\langle \boldsymbol{g}_i,\boldsymbol{x}_i\rangle)$ 和 $\|\beta_t\|_2\leq 1$.*

证明. 定义 $r_t=-\sum_{i=1}^{t}\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle$ 是算法的回报. 则, 我们有

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T}\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t,\boldsymbol{u}\rangle = -r_T+\left\langle \sum_{t=1}^{T}g_t,\boldsymbol{u}\right\rangle\ .$$

由于我们假设 $\text{Regret}_t(\mathbf{0}) \le \epsilon(t)$, 总会有 $r_t \ge -\epsilon(t)$. 基于此, 我们假设对于所有的 $t = 1, \ldots, T$, 有 $\|\boldsymbol{x}_t\|_2 \le r_{t-1} + \epsilon(t)$. 假设有序列 $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{t-1}$ 给出 $\|\boldsymbol{x}_t\|_2 > r_{t-1} + \epsilon(t)$. 设 $\boldsymbol{g}_t = \frac{\boldsymbol{x}_t}{\|\boldsymbol{x}_t\|_2}$. 对于这个序列, 我们有 $r_t = r_{t-1} - \|\boldsymbol{x}_t\|_2 < -\epsilon(t)$, 这与观察结果 $r_t \ge -\epsilon(t)$ 相矛盾.

故, 根据事实 $\|\boldsymbol{x}_t\|_2 \le r_{t-1} + \epsilon(t) \le r_{t-1} + \epsilon(T)$ 我们得到存在 β_t 使得 $\boldsymbol{x}_t = \boldsymbol{\beta}_t(\epsilon(T) + r_{t-1})$ 对于 $\boldsymbol{\beta}_t$ 和 $\|\boldsymbol{\beta}_t\|_2 \le 1$. □

这个定理告诉我们一些重要的事情: 任何对 null 竞争对手产生非减少的后悔的 OLO 算法必须以“矢量”投币算法的形式进行预测. 这直接暗示了以下情况.

定理 5.12. *设 $T \ge 21$. 对于任意的 OLO 算法, 在定理5.11的假设下, 存在序列 $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_T \in \mathbb{R}^d$ 满足条件 $\|\boldsymbol{g}_t\|_2 \le 1$ 和 $\boldsymbol{u}^\star \in \mathbb{R}^d$, 使得*

$$\sum_{i=1}^{t} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u}^\star \rangle \ge 0.8 \|\boldsymbol{u}^\star\|_2 \sqrt{T} \left(\sqrt{0.3 \ln \frac{0.8 \|\boldsymbol{u}^\star\|_2 T}{\epsilon(T)}} - 1 \right) + \epsilon(T) \left(1 - \frac{1}{\sqrt{T}} \right) .$$

证明. 这个证明是通过将唯一用户体验游戏简化为投币游戏, 然后使用投币游戏的奖励上限来实现的.

首先, 设 $\boldsymbol{g}_t = [-c_t, 0, \ldots, 0]$, $c_t \in \{-1, 1\}$ 会在下面进行定义, 对于任意的 $\boldsymbol{x} \in \mathbb{R}^d$ 有 $\langle \boldsymbol{g}_t, \boldsymbol{x} \rangle = -c_t x_1$. 给出定理5.11, $\boldsymbol{x}_t$ 须满足

$$x_{t,1} = \beta_{t,1} \left(\epsilon(T) - \sum_{i=1}^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x}_i \rangle \right) = \beta_{t,1} \left(\epsilon(T) + \sum_{i=1}^{t-1} c_i, x_{i,1} \right),$$

对于 $|\beta_{t,1}| \le 1$ 的 $\beta_{t,1}$. 因此, 以上就是一种投币算法, 对于硬币 c_t, 它下注 $x_{t,1}$ 钱, 最开始的钱是 $\epsilon(T)$. 这意味着定理5.10 中的上限适用于它的回报: 存在一个 c_t 序列, 使得

$$\begin{aligned}
\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \epsilon(T) = -\sum_{t=1}^{T} c_t x_{t,1} - \epsilon(T) \ge -\frac{\epsilon(T)}{\sqrt{T}} \exp\left(\frac{\ln 2 (\sum_{t=1}^{T} c_t)^2}{T} \right) = -\sum_{t=1}^{T} c_t u_1^\star + f^\star(|u_1^\star|) \\
= \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle + f^\star(\|\boldsymbol{u}^\star\|_2),
\end{aligned}$$

$g^\star$ 是定理5.5第二部分中的函数 $f(x) = \frac{\epsilon(T)}{\sqrt{T}} \exp\left(\frac{x^2 \ln 2}{T} \right)$ 和函数 $u^\star = f'(\sum_{t=1}^{T} c_t) \in R$ 的 Fenchel 共轭. 根据定理A.3 重新排序这些项, 我们得到了指定的界.

□

由上述定理可知, 对于任意的 $\alpha > 0$, 具有学习速率的 $\eta = \frac{\alpha}{L\sqrt{T}}$OSD 不具有最佳依赖 $\|\boldsymbol{u}\|_2$.

在以后的类中, 我们将看到投币和 OLO 之间的连接也可以用于设计 OLO 算法. 这将为我们提供最优的无约束的 OLO 算法, 令人惊讶的是它根本不需要学习速率.

5.3 历史片段

OCO 的下界是相当标准的, 给出的证明是 Orabona and Pál [66]中的简化版本.

另一方面, 在线学习文献和优化文献几乎都忽略了无约束情况下的下界问题. 投币和 OLO 之间的联系最早是在 Orabona and Pal [65]. 定理 5.11 是 Ashok Cutkosky 未发表的结果, 在他的博士论文

[27] 中证明了类似的和更普遍的结果. 定理5.12 对于我来说是新的. McMahan and Abernethy [54] 也隐式地提出在无约束 OLO 中使用共轭函数作为下界.

在不受约束的下限中有一个警告: 一个更强的声明将是提前选择 $\boldsymbol{u}^\star$ 的规范. 为此, 我们必须明确地构造 c_t 序列. 一种方法是使用 Rademacher 硬币然后对空竞争对手再次利用后悔假设. Streeter and McMahan [81]采用了这条路径, 但其证明依赖于假设一个具有无穷个局部极小值的非凸函数的全局最优值.Orabona [63]给出了避免这一步的正确证明. 然而, 这里提出的证明, 在后悔的下限中转换奖励上限, 在精神上更简单, 而且 (我希望!) 更容易理解. 考虑到这一点, 据我所知, 这是第一次在课堂上讲授不受约束的 OLO 下限, 我更看重简单性而不是一般性.

Chapter 6

在线镜像下降

在本章中, 我们将介绍在线镜像下降 (OMD) 算法. 为了解释它的起源, 我认为有必要了解次梯度的作用. 特别地, 负的次梯度并不总是指向使函数最小的方向.

6.1 次梯度没有提供信息

你们知道, 在在线学习中, 我们接收到一系列的损失函数, 我们必须在观察损失函数之前输出一个向量, 我们将根据它进行评估. 然而, 如果我们考虑一个简单的情况, 即损失函数序列总是一个固定的函数, 即 $\ell_t(\boldsymbol{x}) = \ell(\boldsymbol{x})$. 我们可以得到很多直观的认识. 如果我们假设的在线算法在这种情况下不起作用, 那么它肯定也不会在更普遍的情况下起作用. 也就是说, 我们证明了在线次梯度下降 (OSD) 的收敛性依赖于次梯度的以下关键性质:

$$\ell(\boldsymbol{x}_t) - \ell(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \ . \tag{6.1}$$

换句话说, 为了最小化这个方程的左手边, 只要最小化右手边就足够了, 这就是线性函数 $\langle \boldsymbol{g}_t, \cdot \rangle$. 上的瞬时线性后悔. 这就是 OSD 能够工作的唯一原因! 然而, 我敢肯定你听过无数次梯度指向最小值的 (错误的) 直觉, 你可能会忍不住认为次梯度也有同样的 (甚至更错误的) 直觉. 事实上, 我确信即使我们证明了基于 (6.1)的后悔保证, 在你的脑海中, 你会一直想" 是的, 这是可行的, 因为次梯度告诉了我去哪里最小化函数" 通常这种观点是如此强烈, 以至于我不得不给出明确的反例来完全说服一个人. 所以, 看一下下面的例子, 它说明了这样一个事实: 次梯度并不总是指向函数减小的方向.

例 6.1. *设 $f(\boldsymbol{x}) = \max[-x_1, x_1 - x_2, x_1 + x_2]$, 见图6.1. 向量 $\boldsymbol{g} = [1, 1]$ 是 $f(\boldsymbol{x})$ 在 $\boldsymbol{x} = (1, 0)$ 的次梯度. 无论我们如何选择步长, 朝着负次梯度的方向移动都不会降低目标函数. 更极端的例子是图6.2, 函数 $f(\boldsymbol{x}) = \max[x_1^2 + (x_2 + 1)^2, x_1^2 + (x_2 - 1)^2]$. 在点 $(1, 0)$, 在负次梯度方向上的任何正步骤都将增加目标函数.*

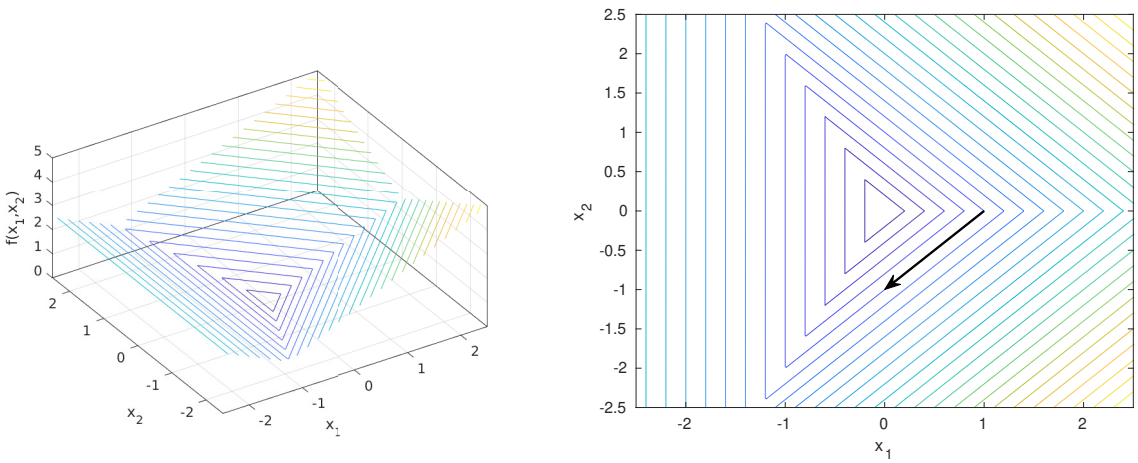

图 6.1: $f(\boldsymbol{x}) = \max[-x_1, x_1 - x_2, x_1 + x_2]$ 的 3D 图 (左) and 水平集 (右). 负的次梯度由黑色箭头表示.

6.2 重新解释在线次梯度下降算法

在线次梯度下降法如何工作? 它的工作方式与我之前告诉您的完全一样: 多亏了(6.1). 但是, 这种不平等到底意味着什么呢? 理解 OSD 算法如何工作的一种方法是认为它最小化了原始目标函数的局部近似. 这在优化算法中并不少见, 例如,Netwon 算法用截断到第二项的泰勒展开构造了一个近似. 由于有了次梯度的定义, 我们可以立即建立一个函数 f 在 $\boldsymbol{x}_0$ 附近的线性下降:

$$f(\boldsymbol{x}) \geq \tilde{f}(\boldsymbol{x}) := f(\boldsymbol{x}_0) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle, \ \forall \boldsymbol{x} \in V \ .$$

所以, 在我们的设置中, 这意味着我们更新在线算法用你收到的损失函数的线性近似的最小值. 不幸的是, 最小化一个线性函数不太可能给我们一个好的在线算法. 的确, 在无界域上线性函数的最小值是 $-\infty$. 那么, 让我们引入另一个关键的概念: 我们只在 $\boldsymbol{x}_0$ 附近限制下界的极小化, 我们有充分的理由相信这个近似更精确. 用 L_2 距离 $\boldsymbol{x}_0$ 小于某个正数 h 的距离编码领域约束, 我们也许会考虑如下的递推.

$$\begin{aligned} \boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x} \in V} \ & f(\boldsymbol{x}_t) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}_t \rangle \\ \text{s.t. } & \|\boldsymbol{x}_t - \boldsymbol{x}\|^2 \leq h \ . \end{aligned}$$

同样, 对于某些 $\eta > 0$, 我们可以考虑无约束公式

$$\operatorname*{argmin}_{\boldsymbol{x} \in V} \ \hat{f}(\boldsymbol{x}) := f(\boldsymbol{x}_0) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle + \frac{1}{2\eta} \|\boldsymbol{x}_0 - \boldsymbol{x}\|_2^2 \ . \tag{6.2}$$

这是一个定义良好的更新方案, 有望使得 $\boldsymbol{x}_t$ 更接近 f 的最佳值. 图6.3 给出了一维的图形表示.

现在我们的故事的最后一个元素: (6.2) 中 argmin 正好是我们在 OSD 中使用的更新! 事实上, 解

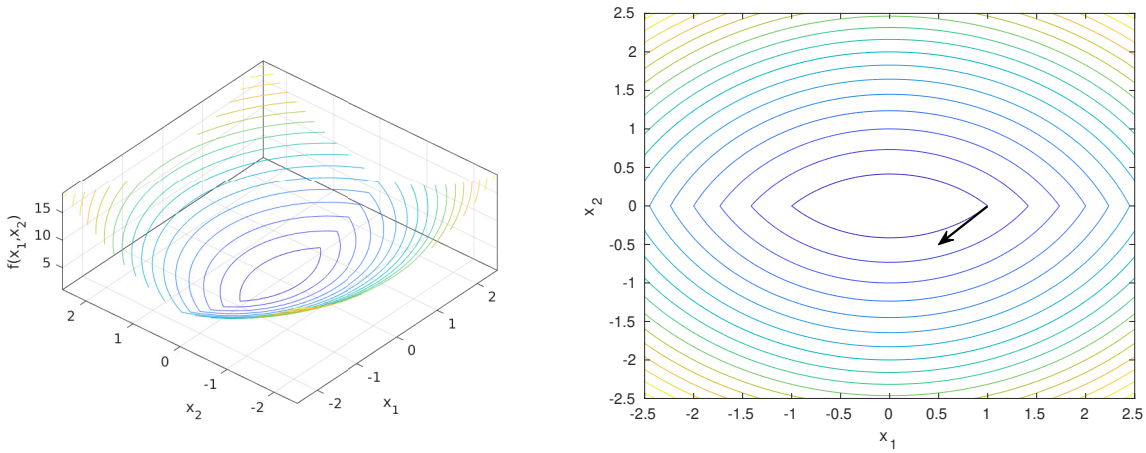

图 6.2: $f(\boldsymbol{x}) = \max[x_1^2 + (x_2+1)^2, x_1^2 + (x_2-1)^2]$ 的 3D 图（左）和水平集 (右), 负的次梯度由黑色箭头表示.

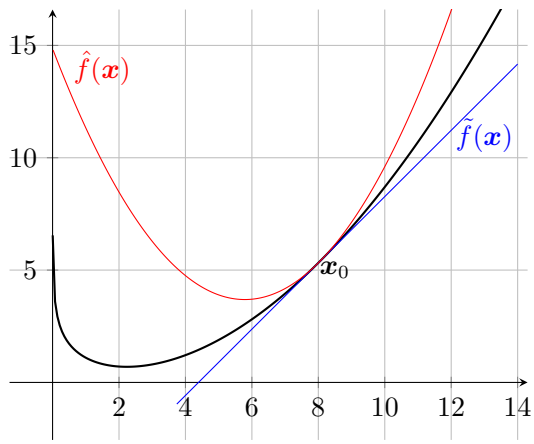

图 6.3: Approximations of $f(\boldsymbol{x})$.

出 argmin 完成平方, 得到

$$\begin{aligned}\operatorname*{argmin}_{\boldsymbol{x}\in V} \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2 &= \operatorname*{argmin}_{\boldsymbol{x}\in V} \ \|\eta_t \boldsymbol{g}_t\|^2 + 2\eta_t\langle \boldsymbol{g}_t, \boldsymbol{x}-\boldsymbol{x}_t\rangle + \|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2 \\ &= \operatorname*{argmin}_{\boldsymbol{x}\in V} \ \|\boldsymbol{x}_t - \eta_t\boldsymbol{g}_t - \boldsymbol{x}\|_2^2 \\ &= \Pi_V(\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t),\end{aligned}$$

Π_V 是 V 上的欧式投影, i.e. $\Pi_V(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y}\in V} \ \|\boldsymbol{x}-\boldsymbol{y}\|_2$.

在 (6.2)中编写 OSD 更新的新方法将是设计在线镜像下降的核心要素. 实际上, 当我们用不同的方法从 $\boldsymbol{x}_t$ 中测量 $\boldsymbol{x}$ 的局部性时. OMD 是这个更新的严格概括. 也就是说, 我们用 L_2 标准的平方来测量到当前点的距离. 如果我们改变常态会发生什么? 我们还需要用标准吗? 要回答这些问题, 我们必须引入另一个有用的数学对象: *Bregman* 散度.

6.3 凸分析 Bits: Bregman 散度

我们首先给出一个新的定义, 一个稍强的凸性概念.

定义 6.2 (严格凸函数)**.** *设 $f: V \subseteq \mathbb{R}^d \to \mathbb{R}$ 和 V 是一个凸集. f 是严格凸函数, 如果*

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) < \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}),\ \forall \boldsymbol{x}, \boldsymbol{y} \in V, \boldsymbol{x} \neq \boldsymbol{y}, 0 < \alpha < 1\ .$$

从定义上可以看出, 任何范数的强凸性都意味着严格凸性. 注意, 对于一个可微函数, 严格凸性也意味着对于 $\boldsymbol{x} \neq \boldsymbol{y}$ [11, Proposition 17.13], 有 $f(\boldsymbol{y}) > f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle$ 我们现在定义了" 距离" 的新概念.

定义 6.3 (Bregman Divergence)**.** *设 $\psi: X \to \mathbb{R}$ 在 $\operatorname{int} X$ 上是严格凸且连续可微的. The **Bregman Divergence** w.r.t. ψ is $B_\psi: X \times \operatorname{int} X \to \mathbb{R}$ 定义为*

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle\ .$$

由定义可知, 由 ψ 的凸性表明, 对于 $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{int} X$, Bregman divergence 总是非负的, 然而, 还有更加强大的力量. 通过 ψ 的严格凸性, 固定一点 $\boldsymbol{y} \in \operatorname{int} X$, 我们得到 $\psi(\boldsymbol{y}) \geq \psi(\boldsymbol{x}) + \langle \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle$, $\forall \boldsymbol{y} \in X$, 只有当 $\boldsymbol{y} = \boldsymbol{x}$ 时才相等. 因此, 严格凸性允许我们使用 Bregman divergence 作为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的相似性度量. 此外, 这种相似性测量与参考点 $\boldsymbol{y}$ 变化. 这也意味着, 从这个定义可以看出,Bregman divergence 是不对称的.

让我给你们一些关于 Bregman divergence 概念的直观理解. 考虑 ψ 在 B 球中围绕 $\boldsymbol{y}$ 和 $\boldsymbol{x} \in B$ 是二阶可导的. 因此, 根据泰勒定理, 存在 $0 \leq \alpha \leq 1$ 使得

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \nabla \psi(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) = \frac{1}{2} (\boldsymbol{x} - \boldsymbol{y})^\top \nabla^2 \psi(\boldsymbol{z}) (\boldsymbol{x} - \boldsymbol{y}),$$

其中 $\boldsymbol{z} = \alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}$. 因此, 我们使用了一个依赖于 ψ 的 Hessian 局部范数. 空间的不同区域会有不同的 Hessian 值, 所以 Bregman 会有不同的表现. 我们将在在线镜像下降 (章节6.5) 和跟踪正则化 (章节7.4) 的局部范数分析中使用这一精确的思想.

当 ψ 是强凸函数时, 我们还可以给出 Bregman 散度的下界, 特别地, 如果 ψ 是 $\operatorname{int} X$ 中的 λ-强凸 w.r.t. a 范数 $\|\cdot\|$, 则有

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) \geq \frac{\lambda}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2\ . \tag{6.3}$$

例 6.4. *若 $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, 则 $B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2 - \frac{1}{2}\|\boldsymbol{y}\|_2^2 - \langle \boldsymbol{y}, \boldsymbol{x} \rangle = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$.*

例 6.5. *若 $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$ 和 $X = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \geq 0, \|\boldsymbol{x}\|_1 = 1\}$, 则 $B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \sum_{i=1}^d x_i \ln \frac{x_i}{y_i}$, that is the 这就是离散分布 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的 Kullback-Leibler 散度.*

我们也有以下引理, 将 3 点之间的 Bregman 散度联系起来.

引理 6.6 ([23])**.** *设 B_ψ 是 Bregman 散度 w.r.t. $\psi: X \to \mathbb{R}$. 则, 对于任意三点 $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{int} X$ 和 $\boldsymbol{z} \in X$, 以下的恒等式成立.*

$$B_\psi(\boldsymbol{z}; \boldsymbol{x}) + B_\psi(\boldsymbol{x}; \boldsymbol{y}) - B_\psi(\boldsymbol{z}; \boldsymbol{y}) = \langle \nabla \psi(\boldsymbol{y}) - \nabla \psi(\boldsymbol{x}), \boldsymbol{z} - \boldsymbol{x} \rangle\ .$$

6.4 在线镜像下降

根据我们之前所说的, 我们可以从 OSD 更新的等效公式开始

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in V} \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2,$$

我们可以用另一种距离来改变最后一项. 特别地, 利用 Bregman 散度 w.r.t. a 函数 ψ, 有

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in V} \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) \ .$$

T 当 $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ 这两个更新是完全相同的.

所以我们在算法6.1中得到在线镜像下降算法.

Algorithm 6.1 Online Mirror Descent

Require: Non-empty closed convex $V \subseteq X \subseteq \mathbb{R}^d$, $\psi : X \to \mathbb{R}$ strictly convex and continuously differentiable on $\operatorname{int} X$, $\boldsymbol{x}_1 \in V$ such that ψ is differentiable in $\boldsymbol{x}_1$, $\eta_1, \cdots, \eta_T > 0$

```
for t = 1 to T do
    Output x_t
    Receive ℓ_t : R^d → (−∞, +∞] and pay ℓ_t(x_t)
    Set g_t ∈ ∂ℓ_t(x_t)
    x_{t+1} = argmin_{x∈V} ⟨g_t, x⟩ + (1/η_t) B_ψ(x; x_t)
end for
```

然而, 如果没有额外的假设, 这个算法有一个问题. 你能看到吗? 问题是 $\boldsymbol{x}_{t+1}$ 可能在 V 的边界上, 下一步, 我们必须对 V 的边界上的一点计算 $B_\psi(\boldsymbol{x}; \boldsymbol{x}_{t+1})$. 给出 $V \subseteq X$, 我们也许会在 X 边界上结束, 此处 Bregman 没有定义!

要解决这个问题, 我们需要以下任意一个假设.

$$\lim_{\boldsymbol{x}\to\partial X} \|\nabla\psi(\boldsymbol{x})\|_2 = +\infty, \tag{6.4}$$

$$V \subseteq \operatorname{int} X \ . \tag{6.5}$$

如果这两个条件中有一个为真, 那么更新将在所有回合中被定义 (证明留作练习).

现在我们有了一个定义良好的算法, 但是它保证了一个次线性的后悔吗? 我们知道, 至少在一种情况下, 它恢复了 OSD 算法, 这是有效的. 因此, 从直观的角度来看, 算法的性能取决于 ψ. 的某些特性. 特别是, ψ 的一个关键特性是强凸性. 为了分析 OMD, 我们首先证明了一个单步关系, 类似于我们证明的在线梯度下降和 OSD 的关系. 注意在这个引理中, 我们会用到很多我们之前介绍过的概念强凸性, 对偶规范, 次梯度, Fenchel-Young 不等式等等. 在某种程度上, 在过去的章节中, 我慢慢地让你们准备好能够证明这个引理.

引理 6.7. *设 B_ψ Bregman 发散 w.r.t. $\psi : X \to \mathbb{R}$, 假设 ψ 是 V 上的对于 $\|\cdot\|$ 是 λ-strongly 凸. 设 $V \subseteq X$ 是一个非空闭凸集. 设 $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$. 假设 (6.4) 或者 (6.5) 成立. 则 $\forall \boldsymbol{u} \in V$ 且满足算法6.1, 以下不等式成立*

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \le \eta_t\langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle \le B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \frac{\eta_t^2}{2\lambda}\|\boldsymbol{g}_t\|_\star^2 \ .$$

证明. 从 OMD 更新的最优条件, 我们得到

$$\langle \eta_t \boldsymbol{g}_t + \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1}\rangle \geq 0,\ \forall \boldsymbol{u} \in V\ . \tag{6.6}$$

根据次梯度的定义, 我们得到了这个

$$\begin{aligned}
\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle &= \langle \nabla\psi(\boldsymbol{x}_t) - \nabla\psi(\boldsymbol{x}_{t+1}) - \eta_t \boldsymbol{g}_t, \boldsymbol{u} - \boldsymbol{x}_{t+1}\rangle + \langle \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1}\rangle + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle \\
&\leq \langle \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1}\rangle + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle && (6.7) \\
&= B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle && (6.8) \\
&\leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \frac{\eta_t^2}{2\lambda}\|\boldsymbol{g}_t\|_\star^2 + \frac{\lambda}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \frac{\eta_t^2}{2\lambda}\|\boldsymbol{g}_t\|_\star^2,
\end{aligned}$$

在第二个不等式中我们使用 (6.6), 在第二个等式中我们使用了引理6.6, 在第三个不等式中我们使用了 $\langle \boldsymbol{x}, \boldsymbol{y}\rangle \leq \frac{1}{2\lambda}\|\boldsymbol{x}\|^2 + \frac{\lambda}{2}\|\boldsymbol{y}\|_\star^2, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \lambda > 0$, 在最后一个不等式中我们使用 (6.3) 因为 ψ 是 λ-strongly 凸 w.r.t. $\|\cdot\|$.

函数值的下界通常是由次梯度的定义决定的. □

下一次, 我们将看到如何使用这一步关系来证明一个后悔约束, 这将最终告诉我们是否以及何时整个构造是一个好主意. 事实上, 值得强调的是, 上述动机无论如何都不足以证明 OMD 算法的存在. 另外, 下次我们将解释为什么该算法被称为在线“镜像”下降. 我们现在可以证明,OMD 是一个后悔.

定理 6.8. *设 $\boldsymbol{x}_1 \in V$ 使得 ψ 在 $\boldsymbol{x}_1$ 上可微. 假设 $\eta_{t+1} \leq \eta_t$, $t = 1, \ldots, T$. 在引理6.7 的假设下, 以及 $\forall \boldsymbol{u} \in V$, 以下悔值边界成立*

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \max_{1\leq t\leq T}\frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_t)}{\eta_T} + \frac{1}{2\lambda}\sum_{t=1}^{T}\eta_t\|\boldsymbol{g}_t\|_\star^2\ .$$

而且, 如果 η_t 是恒定的, i.e. $\eta_t = \eta\ \forall t = 1, \cdots, T$, 我们有

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2\lambda}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_\star^2\ .$$

证明. $\boldsymbol{u} \in V$. 在证明 OGD 时, 用引理6.7中的不等式除以 η_t 并且从 $t = 1, \cdots, T$ 中求和, 得到

$$\begin{aligned}
\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\left(\frac{1}{\eta_t}B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - \frac{1}{\eta_t}B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1})\right) + \sum_{t=1}^{T}\frac{\eta_t}{2\lambda}\|\boldsymbol{g}_t\|_\star^2 \\
&= \frac{1}{\eta_1}B_\psi(\boldsymbol{u}; \boldsymbol{x}_1) - \frac{1}{\eta_T}B_\psi(\boldsymbol{u}; \boldsymbol{x}_{T+1}) + \sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \sum_{t=1}^{T}\frac{\eta_t}{2\lambda}\|\boldsymbol{g}_t\|_\star^2 \\
&\leq \frac{1}{\eta_1}D^2 + D^2\sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2\lambda}\|\boldsymbol{g}_t\|_\star^2 \\
&= \frac{1}{\eta_1}D^2 + D^2\left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_\star^2 \\
&= \frac{D^2}{\eta_T} + \sum_{t=1}^{T}\frac{\eta_t}{2\lambda}\|\boldsymbol{g}_t\|_\star^2,
\end{aligned}$$

用 $D^2 = \max_{1\le t\le T} B_\psi(\boldsymbol{u};\boldsymbol{x}_t)$ 表示.

□

换言之,OMD 让我们得以证明依靠双规范任意组合的后悔保证 $\|\cdot\|$ 和 $\|\cdot\|_\star$. 特别地, 使用原始范数来测量可行集 V 或竞争对手到初始点的距离, 使用对偶范数来测量梯度. 如果你恰好知道这些量的一些情况, 我们可以选择最合适的一对标准来保证一个小的悔值. 你唯一需要的是一个 ψ 函数, 该函数对于你所选择的原始范数 ψ 是强凸的.

总的来说, Lipschitz 函数的悔值界仍然是 $\sqrt{T}$, 唯一的区别是现在的 Lipschitz 常数是用不同的范数来测量的. 同样, 我们为在线次梯度下降 (OSD) 所做的一切都可以在这里简单地使用. 例如, 我们可以使用如下形式的步长

$$\eta_t = \frac{D\sqrt{\lambda}}{\sqrt{\sum_{i=1}^t \|\boldsymbol{g}_i\|_*^2}}$$

来完成 $2\frac{D}{\sqrt{\lambda}}\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2}$ 的悔值界.

下次, 我们将看到 OMD 的实际例子, 它们保证了比 OSD 更好的后悔. 正如我们在 AdaGrad 的案例中所做的那样, 更好的保证将取决于域的形状和次梯度的特征. 相反, 现在我们看到了“镜子”的意思

6.4.1 “镜子”的解释

首先, 我们需要一些凸分析结果.

当我们引入 Fenchel 共轭时, 我们说 $\boldsymbol{x} \in \partial f^\star(\boldsymbol{\theta})$ 如果 $\boldsymbol{\theta} \in \partial f(\boldsymbol{x})$, 也就是意味着 $(\partial f)^{-1} = \partial f^\star$ 在多值映射上有意义. 现在, 我们声明一个更强的结果, 对于函数是强凸的情况.

定理 6.9. *设 $\psi : \mathbb{R}^d \to (-\infty, +\infty]$ 是一个确定的, 凸的闭函数, $\lambda > 0$ 强凸 w.r.t. $\|\cdot\|$. 则,*

1. *$\psi^\star$ 处处有限且可微.*
2. *$\psi^\star$ 是 $\frac{1}{\lambda}$-smooth w.r.t. $\|\cdot\|_\star$.*

我们还将使用下面的最优条件.

定理 6.10. *设 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 是确定的. 则 $\boldsymbol{x}^\star \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$ 如果 $\mathbf{0} \in \partial f(\boldsymbol{x}^\star)$.*

证明. 有

$$\boldsymbol{x}^\star \in \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \Leftrightarrow \forall \boldsymbol{y} \in \mathbb{R}^d,\ f(\boldsymbol{y}) \ge f(\boldsymbol{x}^\star) = f(\boldsymbol{x}^\star) + \langle \mathbf{0}, \boldsymbol{y} - \boldsymbol{x}^\star\rangle \Leftrightarrow \mathbf{0} \in \partial f(\boldsymbol{x}^\star)\ . \qquad \square$$

因此, 我们可以陈述以下定理.

定理 6.11. *设 B_ψ 是 Bregman 发散 w.r.t. $\psi : X \to \mathbb{R}$, ψ 是 $\lambda > 0$ 强凸. 设 $V \subseteq X$ 非空闭凸集使得 $\operatorname{int} X \cap V \neq \{\}$. 则*

$$\operatorname*{argmin}_{\boldsymbol{x}\in V} \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x};\boldsymbol{x}_t) = \nabla\psi_V^\star(\nabla\psi_V(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t), \tag{6.9}$$

ψ_V 是 ψ 对于 V 的限制, 为 $\psi_V := \psi + i_V$.

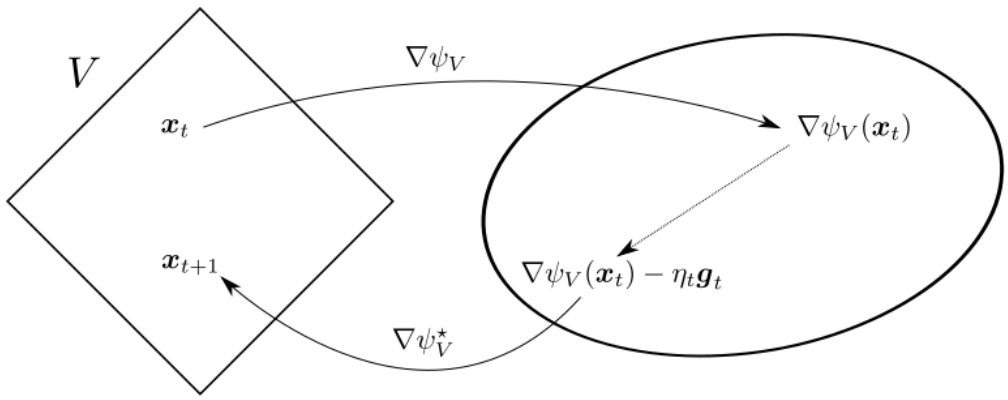

图 6.4: OMD update in terms of duality mappings.

证明. 考虑算法6.1中的更新规则, 让我们看看

$$\begin{aligned}
\boldsymbol{x}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x}\in V} \ \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x};\boldsymbol{x}_t) \\
&= \operatorname*{argmin}_{\boldsymbol{x}\in V} \ \eta_t\langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + B_\psi(\boldsymbol{x};\boldsymbol{x}_t) \\
&= \operatorname*{argmin}_{\boldsymbol{x}\in V} \ \eta_t\langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \psi(\boldsymbol{x}) - \psi(\boldsymbol{x}_t) - \langle \nabla\psi(\boldsymbol{x}_t), \boldsymbol{x}-\boldsymbol{x}_t\rangle \\
&= \operatorname*{argmin}_{\boldsymbol{x}\in V} \ \langle \eta_t\boldsymbol{g}_t - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{x}\rangle + \psi(\boldsymbol{x}) \ .
\end{aligned}$$

现在, 我们要用到一阶最优条件, 所以我们要用到一些次微分. 已知 $\operatorname{int} X \cap V \neq \{\}$, 根据我们看到的次微分定理,, 我们有 $\partial(f+g) = \partial f + \partial g$. 所以, 我们有

$$\mathbf{0} \in \eta_t\boldsymbol{g}_t + \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t) + \partial i_V(\boldsymbol{x}_{t+1}) \ .$$

即

$$\nabla\psi(\boldsymbol{x}_t) - \eta_t\boldsymbol{g}_t \in (\nabla\psi + \partial i_V)(\boldsymbol{x}_{t+1}) = \partial\psi_V(\boldsymbol{x}_{t+1}) \ .$$

或者等同于

$$\boldsymbol{x}_{t+1} \in \partial\psi_V^\star(\nabla\psi(\boldsymbol{x}_t) - \eta_t\boldsymbol{g}_t) \ .$$

根据事实 $\psi_V := \psi + i_V$ 是 λ-strongly 凸的, 我们有 $\partial\psi_V^\star = \{\nabla\psi_V^\star\}$. 因此

$$\boldsymbol{x}_{t+1} = \nabla\psi_V^\star(\nabla\psi(\boldsymbol{x}_t) - \eta_t\boldsymbol{g}_t) \ .$$

注意的是 $\psi_V \equiv \psi$ 对于 V 上的向量, 我们有给定的边界 □

我们来解释一下这个定理. 我们说过, 在线镜像下降法将在线次梯度下降法扩展到非欧几里德范数. 因此, 我们证明的后悔界包含了度量迭代和梯度的双重规范. 我们还说, 使用双重范数来测量梯度是有意义的, 因为这是测量线性函数 $\boldsymbol{x} \to \langle \nabla f(\boldsymbol{y}), \boldsymbol{x}\rangle$ 有多“大”的自然方法. 用一种更正确的方式, 梯度实际上存在于对偶空间中, 也就是在预测 $\boldsymbol{x}_t$. 的不同空间中. 因此, 我们不能将迭代和梯度相加, 就像我们

不能将梨和苹果相加一样. 为什么我们要在 OSD 中做呢? 原因是在这种情况下对偶空间与原始空间重合. 但是, 这是一种非常特殊的情况, 因为我们使用了 L_2 标准. 相反, 在一般情况下, 迭代和梯度位于两个不同的空间.

所以, 在 OMD 中, 我们需要一种从一个空间到另一个空间的方法. 而这正是 $\nabla\psi$ 和 $\nabla\psi^\star$ 的作用, 它们被称为二元映射. 现在可以理解, 该定理告诉我们,OMD 取原始向量 $\boldsymbol{x}_t$, 通过 $\nabla\psi$, 把它转换成一个对偶向量, 在对偶空间中做一个次梯度下降步, 最后通过 $\nabla\psi^\star$ 把向量转换回原始空间. 图6.4总结了这一推理.

例 6.12. 设 $\psi:\mathbb{R}^d\to\mathbb{R}$ 等于 $\psi(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{x}\|_2^2+i_V(\boldsymbol{x})$, 其中 $V=\{\boldsymbol{x}\in\mathbb{R}^d:\|\boldsymbol{x}\|_2\le 1\}$. 则,

$$\psi^\star(\boldsymbol{\theta})=\sup_{\boldsymbol{x}\in V}\langle\boldsymbol{\theta},\boldsymbol{x}\rangle-\frac{1}{2}\|\boldsymbol{x}\|_2^2=\sup_{-1\le\alpha\le 1}\ \alpha\|\boldsymbol{\theta}\|_2-\frac{1}{2}\alpha^2\ .$$

求解约束优化问题, 我们有 $\alpha^\star=\min(1,\|\theta\|_2)$. 因此, 有

$$\psi^\star(\boldsymbol{\theta})=\min\left(\frac{1}{2}\|\boldsymbol{\theta}\|_2^2,\|\boldsymbol{\theta}\|_2-\frac{1}{2}\right),$$

它处处是有限的, 是可微的. 故有 $\nabla\psi(\boldsymbol{x})=\boldsymbol{x}$ 以及

$$\nabla\psi^\star(\boldsymbol{\theta})=\begin{cases}\boldsymbol{\theta}, & \|\boldsymbol{\theta}\|_2\le 1\\ \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2}, & \|\boldsymbol{\theta}\|_2>1\end{cases}=\Pi_V(\boldsymbol{\theta})\ .$$

所以, 利用 (6.9), 我们得到了精确的在线投影次梯度下降的更新.

6.4.2 另一种在线镜像下降更新的方法

还有另一种编写 OMD 更新的方法. 第三种方法使用了 Bregman 投影的概念, 扩展了欧几里得投影的定义, 我们可以定义关于 Bregman 散度的投影. 设 $\Pi_{V,\psi}$ 定义为

$$\Pi_{V,\psi}(\boldsymbol{x})=\operatorname*{argmin}_{\boldsymbol{y}\in V}\ B_\psi(\boldsymbol{y};\boldsymbol{x})\ .$$

在在线学习文献中,OMD 算法通常有两步更新: 首先, 在整个空间上求解 argmin, 然后根据 Bregman 散度投影到 V 上, 下面, 我们将说明, 在大多数情况下, 两步更新等同于 (6.9)中的一步更新.

首先, 我们证明了一个普遍的定理, 允许在整个空间的极小化加上和 Bregman 投影步骤中打破函数的约束极小化.

定理 6.13. 设 $f:\mathbb{R}^d\to(-\infty,+\infty]$ 正确, 闭的, 严格凸, 在 $\operatorname{int}\operatorname{dom}f$ 上可微. 同样设 $V\subset\mathbb{R}^d$ 是一个非空, 闭的凸集且满足 $V\cap\operatorname{dom}f\ne\{\}$ 并且设 $\tilde{\boldsymbol{y}}=\operatorname{argmin}_{\boldsymbol{z}\in\mathbb{R}^d}\ f(\boldsymbol{z})$ 存在以及 $\tilde{\boldsymbol{y}}\in\operatorname{int}\operatorname{dom}f$. 由 $\boldsymbol{y}'=\operatorname{argmin}_{\boldsymbol{z}\in V}\ B_f(\boldsymbol{z};\tilde{\boldsymbol{y}})$ 表示. 则以下成立:

1. $\boldsymbol{y}=\operatorname{argmin}_{\boldsymbol{z}\in V}\ f(\boldsymbol{z})$ 存在且唯一.

2. $\boldsymbol{y}=\boldsymbol{y}'$.

证明. 对于第一点, 根据 [11, Proposition 11.12] 以及 $\tilde{\boldsymbol{y}}$ 的存在, 我们得到 f 是强制的. 故, 依据 Bauschke and Combettes [11, Proposition 11.14], f 的最小值在 V 上存在. 已知 f 是严格凸的, 最小值必须唯一

对于第二点, 根据 $\boldsymbol{y}$ 的定义, 我们有 $f(\boldsymbol{y}) \leq f(\boldsymbol{y}')$. 另一方面, 由一阶最优条件得到 $\nabla f(\tilde{\boldsymbol{y}}) = \mathbf{0}$. 所以, 有

$$f(\boldsymbol{y}') - f(\tilde{\boldsymbol{y}}) = B_f(\boldsymbol{y}'; \tilde{\boldsymbol{y}}) \leq B_f(\boldsymbol{y}; \tilde{\boldsymbol{y}}) = f(\boldsymbol{y}) - f(\tilde{\boldsymbol{y}}),$$

即 $f(\boldsymbol{y}') \leq f(\boldsymbol{y})$. 已知 f 是严格凸的, $\boldsymbol{y}' = \boldsymbol{y}$. □

现在, 注意, 如果 $\tilde{\psi}(\boldsymbol{x}) = \psi(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{x} \rangle$, 则

$$\begin{aligned}
B_{\tilde{\psi}}(\boldsymbol{x}; \boldsymbol{y}) &= \tilde{\psi}(\boldsymbol{x}) - \tilde{\psi}(\boldsymbol{y}) - \langle \nabla \tilde{\psi}(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \\
&= \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \boldsymbol{g} + \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{y} \rangle \\
&= \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \\
&= B_\psi(\boldsymbol{x}; \boldsymbol{y}) \ .
\end{aligned}$$

同时, 定义 $g(\boldsymbol{x}) = B_\psi(\boldsymbol{x}; \boldsymbol{y})$ 和 $\psi(\boldsymbol{x}) + \langle \boldsymbol{z}, \boldsymbol{x} \rangle + K$ 相等, 对于一些 $K \in \mathbb{R}$ 和 $\boldsymbol{z} \in \mathbb{R}^d$. 因此, 在以上定理的假设下, 我们得到 $\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$ 等于

$$\begin{aligned}
\tilde{\boldsymbol{x}}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) \\
\boldsymbol{x}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x} \in V} \ B_\psi(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{t+1}) \ .
\end{aligned}$$

这种更新的优点是, 有时它提供了两个更容易解决的问题, 而不是一个单一的困难问题.

6.5 使用本地规范的 OMD 后悔约束

在引理6.7中, 强凸性基本上告诉我们在所有方向上的最小曲率, 使 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+1}$. 之间的差有上界. 然而, 事实证明, 如果没有这个假设, 我们仍然可以得到一个有意义的后悔上限. 特别是, 对于涉及到使用局部规范的后悔, 我们可以得到一个有趣的表达. 我们将在 (Chapter 10) 关于机械臂的章节中使用这些思想.

引理 6.14. *设 B_ψ 为 Bregman 散度 w.r.t. $\psi : X \to \mathbb{R}$ 并假定 ψ 二阶可导且在其域内具有 Hessian 正定. 设 $V \subseteq X$ 是一个非空闭凸集. 设 $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. 假设 (6.4) 或者 (6.5) 成立. 定义 $\|\boldsymbol{x}\|_A := \sqrt{\boldsymbol{x}^\top A \boldsymbol{x}}$. 则, $\forall \boldsymbol{u} \in V$ 并且满足算法6.1的表示, 存在 $\boldsymbol{z}_t$ 在 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+1}$ 之间的线段上, 以及 $\boldsymbol{z}'_t$ 在 $\boldsymbol{x}_t$ 和 $\tilde{\boldsymbol{x}}_{t+1} := \operatorname{argmin}_{\boldsymbol{x} \in X} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$ 之间的线段上, 使得以下不等式成立*

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \frac{\eta_t^2}{2} \min\left(\|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}_t))^{-1}}, \|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}'_t))^{-1}} \right) \ .$$

证明. 按照引理6.7的证明, 得到不等式(6.8):

$$\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \ . \tag{6.10}$$

根据泰勒定理, 对于一些 $\boldsymbol{z}_t$ 在 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+1}$ 之间的线段上, 有 $B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) = \frac{1}{2}(\boldsymbol{x}_{t+1}-\boldsymbol{x}_t)^\top \nabla^2\psi(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1}-\boldsymbol{x}_t)$. 观察得到 $\frac{1}{2}\|\boldsymbol{x}_{t+1}-\boldsymbol{x}_t\|^2_{\nabla^2\psi(\boldsymbol{z}_t)}$, 因为我们假设 ψ 的 Hessian 为 PD. 因此, 通过 Fenchel-Young 不等式, 我们有

$$\begin{aligned}\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) &\le \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}_t))^{-1}} + \frac{1}{2}(\boldsymbol{x}_{t+1}-\boldsymbol{x}_t)^\top\nabla^2\psi(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1}-\boldsymbol{x}_t) - B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t)\\ &= \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}_t))^{-1}},\end{aligned}$$

它给出了最小值中的第一项.

对于最小值的第二项, 我们观察到

$$\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) \le \max_{\boldsymbol{x}\in X} \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}\rangle - B_\psi(\boldsymbol{x};\boldsymbol{x}_t) = \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_{t+1}\rangle - B_\psi(\tilde{\boldsymbol{x}}_{t+1};\boldsymbol{x}_t) \ .$$

然后, 我们按照第一个上界进行. □

尽管公式显然更困难, 但第二种结果往往更容易使用, 特别是在受限的情况下. 同样, 在定理6.13的假设下, 我们很容易认识到 $\boldsymbol{x}_{t+1}$ 是 $\tilde{\boldsymbol{x}}_{t+1}$ 标识符的 Bregman 投影. 因此, 我们可能会得到一个更糟的点因为它会使 $\boldsymbol{z}'_t$ 的计算变得更为复杂.

6.6 OMD 的例子: 指数梯度

Algorithm 6.2 Exponentiated Gradient

Require: $\eta > 0$

Set $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$
for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t$
  Receive ℓ_t and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$
  $x_{t+1,j} = \frac{x_{t,j}\exp(-\eta g_{t,j})}{\sum_{i=1}^d x_{t,i}\exp(-\eta g_{t,i})},\ j = 1, \dots, d$
end for

设 $X = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \ge 0, \|\boldsymbol{x}\|_1 = 1\}$ 和 $\psi(\boldsymbol{x}) : X \to \mathbb{R}$ 定义为负熵 $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, 定义 $0\ln(0) = 0$. 同样, 我们设置可行性集合 $V = X$. 换句话说, 我们想输出 $\mathbb{R}^d$ 上的离散概率分布.

Fenchel 共轭 $\psi_V^\star(\boldsymbol{\theta})$ 被定义为

$$\psi_V^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}\in V} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \psi(\boldsymbol{x}) = \sup_{\boldsymbol{x}\in V} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \sum_{i=1}^d x_i \ln x_i \ .$$

这是一个有约束的最优化问题, 我们可以用 KKT 条件求解. 我们首先把它写成一个最小化问题

$$\sup_{\boldsymbol{x}\in V} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \sum_{i=1}^d x_i \ln x_i = -\min_{\boldsymbol{x}\in V} \sum_{i=1}^d x_i \ln x_i - \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle \ .$$

KKT 条件是, 对于 $i = 1, \cdots, d$,

$$\begin{aligned} 1 + \ln x_i^\star - \theta_i - \lambda_i^\star + \nu^\star &= 0 \\ \lambda_i^\star &\geq 0 \\ \sum_{i=1}^d x_i^\star &= 1 \\ \lambda_i^\star x_i^\star &= 0 \\ x_i^\star &\geq 0 \ . \end{aligned}$$

根据第一个我们得到 $x_i^\star = \exp(\theta_i + \lambda_i^\star - \nu^\star - 1)$. 根据第三个我们得到 $\exp(\nu^\star + 1) = \sum_{i=1}^d \exp(\theta_i + \lambda_i^\star)$. 根据第四个的互补条件, 有 $\lambda_i^\star = 0$. 将所有的总结起来, 有 $x_i^\star = \frac{\exp(\theta_i)}{\sum_{j=1}^d \exp(\theta_j)}$.

用 $\alpha = \sum_{i=1}^d \exp(\theta_i)$ 表示, 并在我们得到的共轭函数的定义中替换

$$\begin{aligned} \psi_V^\star(\boldsymbol{\theta}) &= \sum_{i=1}^d \left(\frac{1}{\alpha} \theta_i \exp(\theta_i) - \frac{1}{\alpha} \exp(\theta_i)(\theta_i - \ln(\alpha)) \right) \\ &= \ln(\alpha) \frac{1}{\alpha} \sum_{i=1}^d \exp(\theta_i) \\ &= \ln(\alpha) \\ &= \ln \left(\sum_{i=1}^d \exp(\theta_i) \right) \ . \end{aligned}$$

同样有 $(\nabla \psi_V^\star)_j(\boldsymbol{\theta}) = \frac{\exp(\theta_j)}{\sum_{i=1}^d \exp(\theta_i)}$ 和 $(\nabla \psi_V(\boldsymbol{x}))_j = \ln(x_j) + 1$.

将所有放在一起, 我们有熵距离生成功能的在线镜像血迹更新规则.

$$x_{t+1,j} = \frac{\exp(\ln x_{t,j} + 1 - \eta_t g_{t,j})}{\sum_{i=1}^d \exp(\ln x_{t,i} + 1 - \eta_t g_{t,i})} = \frac{x_{t,j} \exp(-\eta_t g_{t,j})}{\sum_{i=1}^d x_{t,i} \exp(-\eta_t g_{t,i})} \ .$$

该算法以算法6.2为概述. 该算法称为指数梯度 Exponentiated Gradient (EG) 因为在更新规则中, 我们采用梯度向量的 component-wise 指数.

让我们来看看我们得到的后悔. 首先, 正如我们所说, 观察到这一点

$$\begin{aligned} B_\psi(\boldsymbol{x}; \boldsymbol{y}) &= \sum_{i=1}^d (x_i \ln x_i - y_i \ln y_i - (\ln(y_i) + 1)(x_i - y_i)) = \sum_{i=1}^d (x_i \ln x_i - y_i \ln y_i - x_i \ln(y_i) + y_i \ln(y_i)) \\ &= \sum_{i=1}^d x_i \ln \frac{x_i}{y_i}, \end{aligned}$$

这是 1 维离散分布 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的 KL 发散. 现在, 以下定理告诉我们 ψ 的强凸性.

引理 6.15 ([74, Lemma 16])**.** *$\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$ 是在集合 $\{\boldsymbol{x} \in \mathbb{R}^d : x_i \geq 0, \|\boldsymbol{x}\|_1 = 1\}$ 上关于 L_1 规范的 1-strongly 凸.*

决定的另一件事是初始点 $\boldsymbol{x}_1$. 设 $\boldsymbol{x}_1$ 是 V 上的最小值 ψ . 这样 $B_\psi(\boldsymbol{u};\boldsymbol{x}_1)$ 简化为 $\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}\in V}\psi(\boldsymbol{x})$. 设 $\boldsymbol{x}_1 = [1/d,\ldots,1/d] \in \mathbb{R}^d$. 有 $B_\psi(\boldsymbol{u};\boldsymbol{x}_1) = \sum_{i=1}^d u_i \ln u_i + \ln d \le \ln d$.

将其结合起来, 有

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \le \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2 \ .$$

假设 $\|\boldsymbol{g}_t\|_\infty \le L_\infty$, 设 $\eta = \sqrt{\frac{2\ln d}{L_\infty^2 T}}$, 获得 $\frac{\sqrt{2}}{2}L_\infty\sqrt{T\ln d}$ 的悔值.

附注 6.16. *注意, 随着熵距离生成功能的 OMD 的时变版本会产生空心束缚, 你能看到为什么吗? 我们将看到 ftrl 如何使用时变正常化器而不是时变的学习率来克服这个问题*

我们还可以使用本地规范获得更严格的绑定. 使用额外的假设 $g_{t,i} \ge 0$, 对于所有 $t = 1,\ldots,T$ 和 $i = 1,\ldots,d$. 概括了引理6.14 从 $t = 1$ 到 T 的不等式, 对于 $\boldsymbol{u} \in V$ 有

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell(\boldsymbol{u}) \le \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}'_t))^{-1}},$$

其中 $\boldsymbol{z}'_t$ 处在 $\boldsymbol{x}_t$ 和 $\tilde{\boldsymbol{x}}_{t+1}$ 之间的线段上. 在这个例子中, 对于 $i = 1,\ldots,d$, 很容易计算 $\tilde{x}_{t+1,i}$ 为 $x_{t,i}\exp(-\eta g_{t,i})$. 而且,$\nabla^2\psi(\boldsymbol{z}'_t)$ 是一个对角线上的对角线矩阵 $\frac{1}{z'_{t,i}}, i = 1,\ldots,d$. 有

$$\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}'_t))^{-1}} = \sum_{i=1}^d g_{t,i}^2 z'_{t,i} \le \sum_{i=1}^d g_{t,i}^2 x_{t,i} \ .$$

将所有结合一起, 最终边界将是

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell(\boldsymbol{u}) \le \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T\sum_{i=1}^d g_{t,i}^2 x_{t,i} \ .$$

这确实是一个更严格的界限因为 $\sum_{i=1}^d g_{t,i}^2 x_{t,i} \le \|\boldsymbol{g}_t\|_\infty^2$.

在线次梯度下降 (OSD) 如何解决同样的问题? 首先, 重要的是要认识到, 没有什么能阻止我们在这个问题上使用 OSD. 我们只需将欧几里得投影实现到单纯形上. 我们从 OSD 得到的后悔界限是

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \le \frac{2}{\eta} + \frac{\eta}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|_2^2,$$

对于任何 $\boldsymbol{u} \in V$, 其中我们设置 $\boldsymbol{x}_1 = [1/d,\ldots,1/d]$ 和 $\|\boldsymbol{u} - \boldsymbol{x}_1\|_2 \le 2$. 假设 $\|\boldsymbol{g}_t\|_\infty \le L_\infty$, 在最坏的情况下, 有 $\|\boldsymbol{g}_t\|_2^2 \le dL_\infty^2$. 设 $\eta = \sqrt{\frac{4}{dL_\infty^2 T}}$, 获得 $2L_\infty\sqrt{dT}$ 的悔值. 因此, 在最坏的情况下, 对于概率单纯形上的在线凸优化, 使用熵距离母函数将对维数的依赖从 $\sqrt{d}$ 转换成 $\sqrt{\ln d}$.

因此, 正如我们已经看到的分析 AdaGrad, 当我们从欧几里得范数转换到其他范数时, 域的形状是重要的因素.

6.7 OMD 示例: p-norm 算法

考虑距离母函数 $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_p^2$, 对于 $1 < p \le 2$ 上的 $X = V = \mathbb{R}^d$. 让我们提醒读者, 向量 $\boldsymbol{x}$ 的 p-范数被定义为 $(\sum_{i=1}^d |x_i|^p)^{\frac{1}{p}}$. 已证 $\psi_V^\star(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_q^2$, 其中 $\frac{1}{p} + \frac{1}{q} = 1$, 故 $q \ge 2$. 让我们计算一下对偶

地图: $(\nabla\psi(\boldsymbol{x}))_j = \mathrm{sign}(x_j)|x_j|^{p-1}\|\boldsymbol{x}\|_p^{2/p-1}$ 和 $(\nabla\psi_V^\star(\boldsymbol{x}))_j = \mathrm{sign}(x_j)|x_j|^{q-1}\|\boldsymbol{x}\|_q^{2/q-1}$. 得到更新规则

$$\begin{aligned}\tilde{x}_{t+1,j} &= \mathrm{sign}(x_{t,j})|x_{t,j}|^{p-1}\|\boldsymbol{x}_t\|_p^{2/p-1} - \eta g_{t,j},\ j=1,\dots,d,\\ x_{t+1,j} &= \mathrm{sign}(\tilde{x}_{t+1,j})|\tilde{x}_{t+1,j}|^{q-1}\left\|\tilde{\boldsymbol{x}}_{t+1}\right\|_q^{2/q-1},\ j=1,\dots,d,\end{aligned}$$

我们将更新分成两个步骤, 以简化符号 (和实现). 从 $\boldsymbol{x}_1 = \boldsymbol{0}$ 开始, 我们有

$$B_\psi(\boldsymbol{u};\boldsymbol{x}_1) = \psi(\boldsymbol{u}) - \psi(\boldsymbol{x}_1) - \langle\nabla\psi(\boldsymbol{x}_1), \boldsymbol{u}-\boldsymbol{x}_1\rangle = \psi(\boldsymbol{u})\ .$$

最后一个因素, 对于 $\|\cdot\|_p$, 实际上 $\psi(\boldsymbol{x})$ 是 $p-1$ strongly 凸的.

引理 6.17 ([74, Lemma 17])**.** *随着 $\|\cdot\|_p$, 对于 $1\le p\le 2$,$\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_p^2$ 是 $(p-1)$-strongly 凸的.*

因此, 悔值边界将变成

$$\sum_{t=1}^T(\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})) \le \frac{\|\boldsymbol{u}\|_p^2}{2\eta} + \frac{\eta}{2(p-1)}\sum_{t=1}^T\|\boldsymbol{g}_t\|_q^2\ .$$

设 $p=2$, 我们得到 (未预测的) 在线次梯度下降. 但是, 我们可以像在 EG 中一样, 将 p 设置为在维度 d 中实现对数依赖性. 让我们再次假设 $\|\boldsymbol{g}_t\|_\infty \le L_\infty$, 有

$$\sum_{t=1}^T\|\boldsymbol{g}_t\|_q^2 \le L_\infty^2 d^{2/q}T\ .$$

同样, 注意 $\|\boldsymbol{u}\|_p \le \|\boldsymbol{u}\|_1$, 得到悔值上界

$$\mathrm{Regret}_T(\boldsymbol{u}) \le \frac{\|\boldsymbol{u}\|_1^2}{2\eta} + \frac{L_\infty^2 d^{2/q}T\eta}{2(p-1)},\ \forall\boldsymbol{u}\in\mathbb{R}^d\ .$$

设 $\eta = \frac{\alpha\sqrt{p-1}}{L_\infty d^{1/q}\sqrt{T}}$, 得到悔值上界

$$\frac{1}{2}\left(\frac{\|\boldsymbol{u}\|_1^2}{\alpha}+\alpha\right)L_\infty\sqrt{T}\frac{d^{1/q}}{\sqrt{p-1}} = \frac{1}{2}\left(\frac{\|\boldsymbol{u}\|_1^2}{\alpha}+\alpha\right)L_\infty\sqrt{T}\sqrt{q-1}d^{1/q} \le \frac{1}{2}\left(\frac{\|\boldsymbol{u}\|_1^2}{\alpha}+\alpha\right)L_\infty\sqrt{T}\sqrt{q}d^{1/q}\ .$$

设 $d\ge 3$, 使得最后一项最小化的 q 的选择是 $q = 2\ln d$, 使得项 $\sqrt{q}d^{1/q} = \sqrt{2e\ln d}$. 我们有 $O(\sqrt{T\ln d})$ 阶的悔值界.

因此, p 范数允许从 OSD 的行为到 EG 的行为进行插值. 注意, 集合 V 是整个空间, 我们仍然可以设 $V = \{\boldsymbol{x}\in\mathbb{R}^d : x_i\ge 0, \|\boldsymbol{x}\|_1 = 1\}$. 虽然这将允许我们得到相同的 EG 的渐近界, 但更新将不再是封闭的形式.

6.8 在线镜像下降的一个应用: 专家建议学习

让我们来介绍一款特殊的在线凸面优化游戏, 名为“带着专家的建议学习”. 在这种情况下, 我们有 d 位专家就每轮比赛给我们一些建议. 反过来, 在每一轮中, 我们必须决定我们想要跟随哪位专家. 在我们做出选择之后, 与每个专家相关的损失被揭示出来, 我们支付与我们选择的专家相关的损失. 这个游

戏的目的是将我们造成的损失与最好的专家的累积损失相比降至最低. 这是一个通用设置, 允许对许多有趣的案例进行建模. 例如, 我们有许多不同的在线学习算法, 我们想要接近其中最好的. 这个问题能解决吗? 如果我们把自己放在对抗性的环境中, 不幸的是, 这是无法解决的! 事实上, 即使有两位专家, 对手也可以强加给我们线性后悔. 让我们看看怎么做. 在每一轮中, 我们必须选择专家 1 或专家 2. 在每一轮中, 对手可以决定我们选择的专家输了 1, 而另一个输了 0. 这意味着算法在 T 轮上的累计损失是 T. 另一方面, 专家 1 和专家 2 的最佳累积损失小于 $T/2$. 这意味着我们的后悔, 无论我们做什么, 都可能和 $T/2$ 一样大.

出现上述问题的原因是对手的力量太大. 降低其能量的一种方法是使用随机化. 我们可以允许算法随机化, 并迫使对手在时间 t 决定损失, 而不知道算法在时间 t 随机化的结果 (但它可以取决于过去的随机化). 这足以使问题变得可解. 此外, 它还会使问题变得凸化, 允许我们在其上使用任何 OCO 算法.

首先, 让我们用原来的公式来写这道题. 我们设置一个离散的可行集 $V = \{\boldsymbol{e}_i\}_{i=1}^d$, 其中 $\boldsymbol{e}_i$ 是在坐标 i 中除了 1 外都是零的向量. 我们的预测和竞争对手都来自 V. 损失是线性损失: $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$, 我们假设 $0 \le g_{t,i} \le 1$, 对于 $t = 1, \ldots, T$ 和 $i = 1, \ldots, d$. 悔值为

$$\mathrm{Regret}_T(\boldsymbol{e}_i) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{e}_i \rangle,\ i = 1, \ldots, d\ . \tag{6.11}$$

唯一使这个问题变得非凸的是可行集, 这显然是一个非凸问题.

现在让我们看看随机化是如何使这个问题变得凸的. 让我们将可行集扩展到 $V' = \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$. 注意 $\boldsymbol{e}_i \in V'$. 对于这个问题, 我们可以使用 OCO 算法来最小化后悔.

$$\mathrm{Regret}'_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle, \forall \boldsymbol{u} \in V'\ .$$

我们能不能找到一种方法, 把这个后悔的上限转换成我们在 (6.11)中关心的那个上限? 一种方法是下面的方法: 在每个时间步上, 构造一个概率为 i_t, 它等于对于 $i = 1, \ldots, d$, 概率为 $x_{t,i}$ 的 i . 然后, 根据 i_t 的结果选择专家. 得

$$\mathbb{E}[g_{t,i_t}] = \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle,$$

和

$$\mathbb{E}[\mathrm{Regret}_T(\boldsymbol{e}_i)] = \mathbb{E}[\mathrm{Regret}'_T(\boldsymbol{e}_i)] = \mathbb{E}\left[\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{e}_i \rangle\right]\ .$$

这意味着我们可以使用随机化的 OCO 算法将非凸后悔最小化. 我们可以在算法6.3中总结这一推理.

例如, 如果使用 EG 作为 OCO 算法, 设 $\boldsymbol{x}_1 = [1/d, \ldots, 1/d]$, 得到以下更新规则

$$x_{t+1,j} = \frac{x_{t,j} \exp(-\eta g_{t,j})}{\sum_{i=1}^d x_{t,i} \exp(-\eta g_{t,i})},\ j = 1, \ldots, d$$

悔值变成

$$\mathbb{E}[\mathrm{Regret}_T(\boldsymbol{e}_i)] \le \frac{\sqrt{2}}{2} L_\infty \sqrt{T \ln d},\ \forall \boldsymbol{e}_i\ .$$

值得强调的是刚刚获得的结果的重要性: 我们可以设计一种算法, 在期望中接近集合中最好的专家, 只需支付集合大小的对数惩罚.

Algorithm 6.3

Require: $\boldsymbol{x}_1 \in \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$

```
for t = 1 to T do
    Draw i_t according to P(i_t = i) = x_{t,i}
    Select expert i_t
    Observe all the experts' losses g_t and pay the loss of the selected expert
    Update x_t with an OCO algorithm with feasible set {x ∈ R^d : x_i > 0, ||x||_1 = 1}
end for
```

在未来, 我们将看到算法对于单纯形中的任何 $\boldsymbol{u}$ 都能达到 $O(\sqrt{T \cdot \mathrm{KL}(\boldsymbol{u}; \boldsymbol{x}_1)})$ 更好的后悔保证. 您应该能够说服自己,EG 中的任何 η 设置都不允许实现这样的后悔保证. 事实上, 该算法将基于一种截然不同的策略.

6.9 历史片段

镜像下降 (MD) 是由 Nemirovsky and Yudin [60] 在批量设置中引入的. 我在这里描述的带有 Bregman 发散的 MD (有一些小的变化) 是由 Beck and Teboulle [13]完成的. 较小的变化在于将 ψ 的结构域 X 与可行集 V 分离. 这允许使用不满足条件(6.4) 但满足 (6.5)的 ψ 函数. 在在线设置中, Warmuth and Jagota [86]首次使用镜像下降方案.

大多数关于 OMD 的在线学习文献都假设 ψ 是 *Legendre* [see, e.g., 20] 这与假设(6.4)相对应. 这个条件可以证明 $\nabla\psi_V^\star = (\nabla\psi_V)^{-1}$. 然而, 结果表明, Legendre 条件是不必要的, 我们只需要函数 ψ 在预测函数 $\boldsymbol{x}_t$ 上是可微的. 事实上, 我们只需要 (6.4) 或者 (6.5) 中的其中一个即可成立. 去掉 Legendre 假设, 可以更容易地将 OMD 与可行集/Bregman 散度的不同组合一起使用. 因此, 我根本没有引入 Legendre 函数的概念, 而是依赖 Beck and Teboulle [13]所描述的 (稍作修改) OMD .

当地规范是 Abernethy et al. [2]等引入的, for Follow-The-Regularized-Leader with self-concordant regularizers.

EG 算法是由 Kivinen and Warmuth [43]引入的, 但并不是 OMD 的具体实例. 作为镜像下降的一个例子, Beck and Teboulle [13] 重新发现了离线情况下的 EG. 后来,Cesa-Bianchi and Lugosi [20] 证明 EG 只是 OMD 的一个实例. 在线预测的 p-norm 算法最初是由 Grove et al. [38]等人提出来的. 设置 $q = 2\ln d$ 的诀窍来自于 Gentile and Littlestone [36] (在线学习) 显然是在 Ben-Tal et al. [14]等人中重新发现的 (优化). 在 Littlestone and Warmuth [52] 和 Vovk [85]中引入了专家设置的在线学习. 算法6.3 中的思想基于乘法加权算法 [52] 和对冲算法 [33]. 到目前为止, 关于与专家一起学习的文献是巨大的, 有大量关于算法和设置的变化.

Chapter 7

FTRL 算法

到目前为止, 我们只关注在线次梯度下降及其推广, 在线镜像下降, 并在第一章中对 Follow-The-Leader (FTL) 进行了简单的分析. 在本章中, 我们将把 FTL 扩展为一种功能强大的通用在线凸优化算法: **Follow-the-Regularized-Leader** (FTRL).

FTRL 是一种非常直观的算法: 在每个时间步, 它将发挥过去损失总和的最小化加上一个时变的正则化. 我们将看到, 需要正则化来使算法在线性损失下“更稳定”, 并避免我们在示例 $ex: failure_ftl$ 中看到的来回跳跃.

7.1 FTRL Algorithm

Algorithm 7.1 Follow-the-Regularized-Leader

Require: Closed and non-empty set $V \subseteq \mathbb{R}^d$, a sequence of regularizers $\psi_1, \ldots, \psi_T : \mathbb{R}^d \to (-\infty, +\infty]$

for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$
  Receive $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ and pay $\ell_t(\boldsymbol{x}_t)$
end for

如上所述, 在 FTRL 中, 我们输出规则化累积过去损失的最小化. 应该清楚的是,FTRL 不是一个算法, 而是一个算法族, 就像 OMD 是一个算法族一样.

在分析算法之前, 让我们先对其进行一些直观的了解. 在 OMD 中, 我们看到算法的“状态”存储在当前迭代 $\boldsymbol{x}_t$ 中, 因为下一次迭代 $\boldsymbol{x}_{t+1}$ 取决于 $\boldsymbol{x}_t$ 和在时间 t 接收到的损失 (学习率的选择对下一次迭代只有很小的影响). 相反, 在 FTRL 中, 下一次迭代 $\boldsymbol{x}_{t+1}$ 取决于截至时间 t 收到的整个损失历史. 这会产生直接的后果: 在 V 有界的情况下,OMD 将只“记住”最后一次 $\boldsymbol{x}_t$, 而不是投影之前的迭代. 另一方面,FTRL 将过去的整个历史保存在内存中, 原则上允许在 V 投影之前恢复迭代.

这种行为上的差异可能会让读者认为 FTRL 的计算量更大, 内存也更昂贵. 的确如此！但是, 我们也将看到, 有一种方法可以考虑近似损失, 使算法与 OMD 一样昂贵, 但严格地保留比 OMD 更多的信

息.

对于 FTRL, 我们证明了一个令人惊讶的结果: 与后悔平起平坐!

引理 7.1. *设 $V \subseteq \mathbb{R}^d$ 是闭的非空集. 由 $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$ 表示. 假设 $\operatorname{argmin}_{\boldsymbol{x}\in V} \ F_t(\boldsymbol{x})$ 非空且 $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}\in V} \ F_t(\boldsymbol{x})$. 则, 对于任意 $\boldsymbol{u}$, 有*

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t)-\ell_t(\boldsymbol{u})) = \psi_{T+1}(\boldsymbol{u})-\min_{\boldsymbol{x}\in V} \ \psi_1(\boldsymbol{x})+\sum_{t=1}^{T}[F_t(\boldsymbol{x}_t)-F_{t+1}(\boldsymbol{x}_{t+1})+\ell_t(\boldsymbol{x}_t)]+F_{T+1}(\boldsymbol{x}_{T+1})-F_{T+1}(\boldsymbol{u}) \ .$$

证明. 鉴于 $\ell_t(\boldsymbol{x}_t)$ 出现在等式两边的情况, 我们只需验证

$$-\sum_{t=1}^{T}\ell_t(\boldsymbol{u}) = \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x}\in V} \ \psi_1(\boldsymbol{x}) + \sum_{t=1}^{T}[F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1})] + F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u}) \ .$$

记得 $F_1(\boldsymbol{x}_1) = \min_{\boldsymbol{x}\in V} \ \psi_1(\boldsymbol{x})$ 以及 F_t 的总和是微小的, 有

$$-\sum_{t=1}^{T}\ell_t(\boldsymbol{u}) = \psi_{T+1}(\boldsymbol{u})-F_1(\boldsymbol{x}_1)+F_1(\boldsymbol{x}_1)-F_{T+1}(\boldsymbol{x}_{T+1})+F_{T+1}(\boldsymbol{x}_{T+1})-F_{T+1}(\boldsymbol{u}) = \psi_{T+1}(\boldsymbol{u})-F_{T+1}(\boldsymbol{u}),$$

根据定义 F_{T+1}, 这是正确的. □

附注 7.2. *请注意, 我们基本上没有在 ℓ_t 或者 ψ_t 上假设任何东西, 即使对于非凸损失和正则化函数, 上述等式也是成立的. 然而, 解决每一步的最小化问题在计算上可能是不可行的.*

附注 7.3. *请注意, 定理中等式的左侧不依赖于 ψ_{T+1}, 因此如果需要, 我们可以将其设置为 ψ_T.*

附注 7.4. *由于 FTRL 算法对加到正则化子上的任何正数都是不变的, 所以我们总是可以用 $\psi_t(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi_t(\boldsymbol{x})$ 而不是 $\psi_t(\boldsymbol{u})$. 来表示后悔保证. 但是, 为了清楚起见, 有时我们会显式选择正则化函数, 使其最小值为 0.*

然而, 尽管令人惊讶, 但上述平等还不是后悔的界限, 因为它是“隐含的”. 事实上, 平等的两边都出现了损失.

让我们更仔细地看看这种等式. 如果 $\boldsymbol{u} \in \operatorname{dom} F_{t+1}$, 我们得到 r.h.s 上最后两项之和. 是负的. 另一方面, r.h.s. 得前两个条款, 与我们在 OMD 中发现的相似. 有趣的部分是项 $F_t(\boldsymbol{x}_t)-F_{t+1}(\boldsymbol{x}_{t+1})+\ell_t(\boldsymbol{x}_t)$. 的和. 为了给出一个正在发生的事情的直觉, 让我们考虑正则化随时间变化的情况, 即, $\psi_t = \psi$. 因此, 总和中的术语可以重写为

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) = \psi(\boldsymbol{x}_t) + \sum_{i=1}^{t}\ell_i(\boldsymbol{x}_t) - \left(\psi(\boldsymbol{x}_{t+1}) + \sum_{i=1}^{t}\ell_i(\boldsymbol{x}_{t+1})\right) \ .$$

因此, 我们在算法的两个连续预测中测量正则化损失的最小化 (具有两个不同的正则化) 之间的距离. 粗略地说, 如果 $\boldsymbol{x}_t \approx \boldsymbol{x}_{t+1}$ 和损失 + 正规化是“不错的”, 这一项就会很小. 这应该会提醒你 OMD 的更新, 我们约束 $\boldsymbol{x}_{t+1}$ 接近 $\boldsymbol{x}_{t+1}$. 相反, 如果到时间 t 的正则化损失的最小化接近到时间 $t+1$. 的损失的最小化, 那么这里的两个预测将彼此接近. 所以, 就像在 OMD 中一样, 如果损失没有足够的曲率, 这里的正则化将起到稳定预测的关键作用. 因此, 就像在 OMD 中一样, 如果损失没有足够的曲率, 这里的正则化将起到稳定预测的关键作用, 如果到时间 t 的正则化损失的最小化接近到时间 t+1 的损失的最小化.

在下面, 我们将看到从引理7.1得到显式上界的不同方法.

7.2 FTRL 利用强凸性确定悔值界限

当损失加正则化是强凸的时候, 很容易得到 FTRL 的后悔上界. 事实上, 强凸性保证了预测的稳定性. 为了量化这种直觉, 我们需要一个强凸函数的性质.

7.2.1 Convex Analysis Bits: 强凸函数的性质

我们将对强凸函数使用下面的引理.

引理 7.5. *设 $f:\mathbb{R}^d \to (-\infty,+\infty]$ μ 关于范数 $\|\cdot\|$ 是强凸的. 则, 对于所有的 $\boldsymbol{x},\boldsymbol{y} \in \operatorname{dom} f$, $\boldsymbol{g} \in \partial f(\boldsymbol{y})$, 和 $\boldsymbol{g}' \in \partial f(\boldsymbol{x})$, 有*

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \le \langle \boldsymbol{g}, \boldsymbol{x}-\boldsymbol{y}\rangle + \frac{1}{2\mu}\|\boldsymbol{g}-\boldsymbol{g}'\|_*^2 \ .$$

证明. 定义 $\phi(\boldsymbol{z}) = f(\boldsymbol{z}) - \langle \boldsymbol{g}, \boldsymbol{z}\rangle$. 发现 $\mathbf{0} \in \partial\phi(\boldsymbol{y})$, 因此 $\boldsymbol{y}$ 是 $\phi(\boldsymbol{z})$ 的最小值. 同样, 注意 $\boldsymbol{g}'-\boldsymbol{g} \in \partial\phi(\boldsymbol{x})$. 因此, 有

$$\begin{aligned}
\phi(\boldsymbol{y}) &= \min_{\boldsymbol{z}\in\operatorname{dom}\phi} \ \phi(\boldsymbol{z}) \\
&\ge \min_{\boldsymbol{z}\in\operatorname{dom}\phi} \ \left(\phi(\boldsymbol{x}) + \langle \boldsymbol{g}'-\boldsymbol{g}, \boldsymbol{z}-\boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2\right) \\
&\ge \inf_{\boldsymbol{z}\in\mathbb{R}^d} \ \left(\phi(\boldsymbol{x}) + \langle \boldsymbol{g}'-\boldsymbol{g}, \boldsymbol{z}-\boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2\right) \\
&= \phi(\boldsymbol{x}) - \frac{1}{2\mu}\|\boldsymbol{g}'-\boldsymbol{g}\|_*^2,
\end{aligned}$$

其中最后一步来自平方范数的共轭函数, 请参见示例5.7. □

推论 7.6. *设 $f:\mathbb{R}^d \to (-\infty,+\infty]$ μ 是关于 $\|\cdot\|$ 强凸的. 设 $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}} \ f(\boldsymbol{x})$. 则对于 $\boldsymbol{x} \in \operatorname{dom} f$, 和 $\boldsymbol{g} \in \partial f(\boldsymbol{x})$, 有*

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) \le \frac{1}{2\mu}\|\boldsymbol{g}\|_*^2 \ .$$

换言之, 上面的引理说明次优间隔的上界与次梯度的平方范数成正比.

7.2.2 明确的后悔界限

我们现在陈述一个引理, 它量化了对预测的“稳定性”的直觉.

引理 7.7. *利用引理7.1的记号和假设, 设 F_t 是真的, 且 λ_t 是强凸的, w.r.t. $\|\cdot\|$, ℓ_t 为真且凸, 设 $\partial\ell_t(\boldsymbol{x}_t)$ 非空, 有*

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) \le \frac{\|\boldsymbol{g}_t\|_*^2}{2\lambda_t} + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}),$$

对于所有 $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$.

证明. 有

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t)) - (F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_{t+1})) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\le (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t)) - (F_t(\boldsymbol{x}_t^\star) + \ell_t(\boldsymbol{x}_t^\star)) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\le \frac{\|\boldsymbol{g}_t'\|_*^2}{2\lambda_t} + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}),
\end{aligned}$$

在第二个不等式中, 我们使用推论7.6中的定义 $\boldsymbol{x}_t^\star := \operatorname{argmin}_{\boldsymbol{x}\in V}\ F_t(\boldsymbol{x}) + \ell_t(\boldsymbol{x})$, 以及 $\boldsymbol{g}_t' \in \partial(F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t))$. 观察到 $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}\in V}\ F_t(\boldsymbol{x})$, 根据定理6.10, 有 $\mathbf{0} \in \partial(F_t(\boldsymbol{x}_t) + i_V(\boldsymbol{x}_t))$. 因此, 利用定理2.19, 我们选择 $\boldsymbol{g}_t'$ 则得到 $\boldsymbol{g}_t' \in \partial\ell_t(\boldsymbol{x}_t)$. □

让我们来看看 FTRL 的一些直接应用.

推论 7.8. *设 ℓ_t 为凸损失函数序列. 设 $\psi : V \to \mathbb{R}$ 是 μ 强凸函数 w.r.t. $\|\cdot\|$. 将正则化序列设为 $\psi_t(\boldsymbol{x}) = \frac{1}{\eta_{t-1}}(\psi(\boldsymbol{x}) - \min_{\boldsymbol{z}} \psi(\boldsymbol{z}))$, 其中 $\eta_{t+1} \le \eta_t$, $t = 1, \ldots, T$. 则 FTRL 保证*

$$\sum_{t=1}^T \ell(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}\in V} \psi(\boldsymbol{x})}{\eta_{T-1}} + \frac{1}{2\mu}\sum_{t=1}^T \eta_{t-1}\|\boldsymbol{g}_t\|_\star^2,$$

对于所有的 $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$. 此外, 如果函数 ℓ_t 是 L-Lipschitz, 则设置 $\eta_{t-1} = \frac{\alpha\sqrt{\mu}}{L\sqrt{t}}$ 得到

$$\sum_{t=1}^T \ell(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \left(\frac{\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x})}{\alpha} + \alpha\right)\frac{L\sqrt{T}}{\sqrt{\mu}} .$$

证明. 推论直接来自引理7.1, 引理7.7, 以及根据假设有 $\psi_t(\boldsymbol{x}) - \psi_{t+1}(\boldsymbol{x}) \le 0$, $\forall \boldsymbol{x}$. 设 $\psi_{T+1} = \psi_T$, 这要归功于注释7.3. □

这看起来像是 OMD 的后悔保证, 但是这里有一个非常重要的区别: 最后一项包含一个时变元素 (η_t) 但是域不一定是有界的! 另外, 我使用了正则化的 $\frac{1}{\eta_{t-1}}\psi(\boldsymbol{x})$ 和 not $\frac{1}{\eta_t}\psi(\boldsymbol{x})$ 来提醒你另一个重要的区别: 在 OMD 中, 学习速率 η_t 是在接收次梯度 $\boldsymbol{g}_t$ 之后选择的, 而在这里, 你必须在接收它之前选择它!

7.3 FTRL 具有线性化损失的情况

OMD 和 FTRL 之间的一个重要区别是, 这里的更新规则似乎比 OMD 要昂贵得多, 因为我们需要在每一步都解决一个优化问题. 然而, 在计算复杂度与 OMD 相同的情况下, 我们可以对线性化损失使用 FTRL, 并得到相同的界.

考虑损失是线性的情况, $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle$, $t = 1, \ldots, T$, 我们已经知道 FTRL 的预测是

$$\boldsymbol{x}_{t+1} \in \operatorname*{argmin}_{\boldsymbol{x}\in V}\ \psi_{t+1}(\boldsymbol{x}) + \sum_{i=1}^t \langle \boldsymbol{g}_i, \boldsymbol{x}\rangle = \operatorname*{argmax}_{\boldsymbol{x}\in V} \left\langle -\sum_{i=1}^t \boldsymbol{g}_i, \boldsymbol{x}\right\rangle - \psi_{t+1}(\boldsymbol{x}) .$$

用 $\psi_{V,t}(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + i_V(\boldsymbol{x})$ 表示. 如果假设 $\psi_{V,t}$ 为真, 凸, 闭, 利用定理5.5, 有 $\boldsymbol{x}_{t+1} \in \partial\psi_{V,t+1}^\star(-\sum_{i=1}^t \boldsymbol{g}_i)$. 再者, 如果 $\psi_{V,t+1}$ 是强凸的, 我们知道 $\psi_{V,t+1}^\star$ 是可微的, 得到

$$\boldsymbol{x}_{t+1} = \nabla\psi_{V,t+1}^\star\left(-\sum_{i=1}^t \boldsymbol{g}_i\right) . \tag{7.1}$$

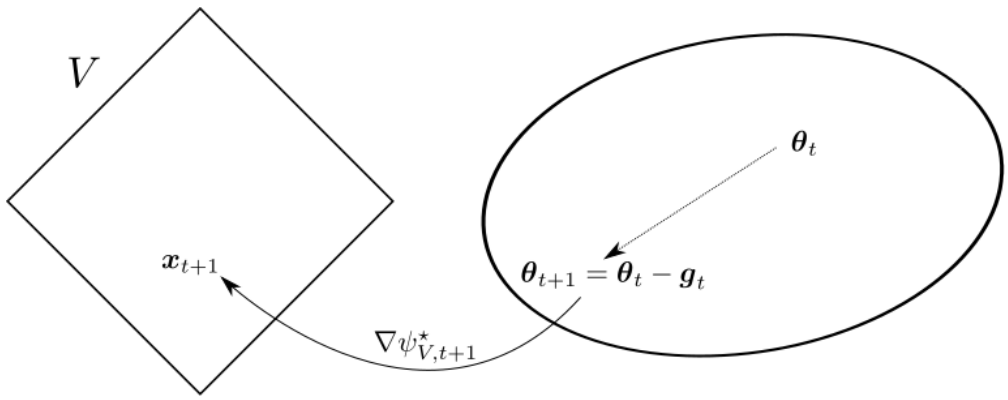

图 7.1: 具有线性损失的 FTRL 双重映射.

反过来, 可以按以下方式编写此更新

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \boldsymbol{g}_t,\\ \boldsymbol{x}_{t+1} &= \nabla\psi^\star_{V,t+1}(\boldsymbol{\theta}_{t+1})\ .\end{aligned}$$

这与图7.1中所示相对应.

将其与以类似方式重写的 OMD 的镜像更新进行比较:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \nabla\psi(\boldsymbol{x}_t) - \eta_t\boldsymbol{g}_t,\\ \boldsymbol{x}_{t+1} &= \nabla\psi^\star_V(\boldsymbol{\theta}_{t+1})\ .\end{aligned}$$

它们非常相似, 但有重要的区别:

- 在 OMD 中, 状态保存在 $\boldsymbol{x}_t$ 中, 因此我们需要在进行更新之前将其转换为对偶变量, 然后再转换回原始变量.
- 在具有线性损失的 FTRL 中, 状态直接保持在对偶空间中, 然后更新, 然后转换为原始变量. 原始变量仅用于预测, 而不是直接在更新中使用.
- 在 OMD 中, 样本由通常递减的学习率加权.
- 在线性损失的 FTRL 中, 所有子梯度具有相同的权重, 但正则化通常随着时间的推移而增加.

此外, 我们不会丢失任何捆绑的东西！事实上, 我们可以对线性化损失 $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle$ 运行 FTRL, 其中 $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$, 保证对损失 ℓ_t 完全相同的后悔. 该过程的算法在算法7.2中.

事实上, 使用次梯度的定义和推论7.8的假设, 有

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T(\ell_t(\boldsymbol{x}_t)-\ell_t(\boldsymbol{u})) \le \sum_{t=1}^T(\tilde{\ell}_t(\boldsymbol{x}_t)-\tilde{\ell}_t(\boldsymbol{u})) \le \frac{\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}\in V}\psi(\boldsymbol{x})}{\eta_{T-1}} + \frac{1}{2\mu}\sum_{t=1}^T \eta_{t-1}\|\boldsymbol{g}_t\|^2_\star,\ \forall \boldsymbol{u}\in V\ .$$

与推论7.8的唯一不同的是, 这里的 $\boldsymbol{g}_t$ 是我们在算法中使用的具体值, 而在推论 7.8 中, 该语句适用于 $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$ 的任何选择.

在下一个示例中, 我们可以看到 FTRL 和 OMD 的不同行为.

Algorithm 7.2 Follow-the-Regularized-Leader on Linearized Losses

Require: Convex, closed, and non-empty set $V \subseteq \mathbb{R}^d$, a sequence of regularizers $\psi_1, \ldots, \psi_T : \mathbb{R}^d \to (-\infty, +\infty]$

1: **for** $t = 1$ **to** T **do**
2: Output $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle$
3: Receive $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ and pay $\ell_t(\boldsymbol{x}_t)$
4: Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
5: **end for**

例 7.9. *考虑 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \le 1\}$. 对于学习率为 $\eta_t = \frac{1}{\sqrt{t}}$ 和 $\boldsymbol{x}_1 = \mathbf{0}$ 的在线次梯度下降 OSD, 更新为*

$$\begin{aligned}
\tilde{\boldsymbol{x}}_{t+1} &= \boldsymbol{x}_t - \frac{1}{\sqrt{t}} \boldsymbol{g}_t, \\
\boldsymbol{x}_{t+1} &= \tilde{\boldsymbol{x}}_{t+1} \min\left(\frac{1}{\|\tilde{\boldsymbol{x}}_{t+1}\|_2}, 1 \right) .
\end{aligned}$$

另一方面, 在具有线性化损失的 FTRL 中, 我们可以使用 $\psi_t(\boldsymbol{x}) = \frac{\sqrt{t}}{2}\|\boldsymbol{x}\|_2^2 + i_V(\boldsymbol{x})$ 并且很容易验证 (7.1)中的更新为

$$\begin{aligned}
\tilde{\boldsymbol{x}}_{t+1} &= \frac{-\sum_{i=1}^{t} \boldsymbol{g}_i}{\sqrt{t}}, \\
\boldsymbol{x}_{t+1} &= \tilde{\boldsymbol{x}}_{t+1} \min\left(\frac{1}{\|\tilde{\boldsymbol{x}}_{t+1}\|_2}, 1 \right) .
\end{aligned}$$

虽然这两个更新的后悔保证是相同的, 但从直观的角度来看, 由于投影和我们只记住投影迭代的事实,OMD 似乎正在丢失大量潜在信息.

7.3.1 具有线性化损失的 FTRL 可以等同于 OMD

首先, 我们看到, 尽管 FTRL 和 OMD 看起来非常不同, 但在某些情况下, 它们是等效的. 例如, 考虑 $V = X = \operatorname{dom} \psi$ 的情况. OMD 的输出为

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}} \ \langle \eta \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) .$$

假设对于所有的 $t = 1, \ldots, T$, 有 $\boldsymbol{x}_{t+1} \in \operatorname{int} \operatorname{dom} \psi$, 这意味着 $\eta \boldsymbol{g}_t + \nabla \psi(\boldsymbol{x}_{t+1}) - \nabla \psi(\boldsymbol{x}_t) = \mathbf{0}$, 即 $\nabla \psi(\boldsymbol{x}_{t+1}) = \nabla \psi(\boldsymbol{x}_t) - \eta \boldsymbol{g}_t$. 设 $\boldsymbol{x}_1 = \min_{\boldsymbol{x} \in V} \psi(\boldsymbol{x})$, 有

$$\nabla \psi(\boldsymbol{x}_{t+1}) = -\eta \sum_{i=1}^{t} \boldsymbol{g}_i .$$

另一方面, 考虑具有正则化因子 $\psi_t = \frac{1}{\eta}\psi$ 的线性损失 FTRL, 则

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}} \ \frac{1}{\eta}\psi(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle = \operatorname*{argmin}_{\boldsymbol{x}} \ \psi(\boldsymbol{x}) + \eta \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle .$$

假设 $\boldsymbol{x}_{t+1} \in \operatorname{int}\operatorname{dom}\psi$, 表明 $\nabla\psi(\boldsymbol{x}_{t+1}) = -\eta\sum_{i=1}^{t}\boldsymbol{g}_i$. 此外, 假设 $\nabla\psi$ 是可逆的, 则意味着 FTRL 和 OMD 的预测是相同的.

这种等价性立即让我们直观地了解了 ψ 在这两种算法中的作用: 同一个函数正在诱导 Bregman, 这是我们的相似性度量, 也是 FTRL 中的正则化. 此外,FTRL 的正则化增长率的倒数在 OMD 中起着学习率的作用.

例 7.10. *考虑 $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ 和 $V = \mathbb{R}^d$, 然后满足上述条件, 使 OMD 的预测值与 FTRL 的预测值相等.*

7.4 基于局部范数的 FTRL 后悔界限

在引理7.7中, 强凸性基本上告诉我们损失加正则化在所有方向上都有一些最小曲率. 然而, 就像在 OMD 的案例中一样, 事实证明, 我们可以再次使用局部规范的概念来后悔上界.

引理 7.11. *在与引理7.1相同的假设下, 假设 $\psi_1,\ldots,\psi_T$ 二次可微并且在其定义域内 Hessian 正定. 且 $\psi_{t+1}(\boldsymbol{x}) \geq \psi_t(\boldsymbol{x}), \forall \boldsymbol{x} \in V, t = 1,\ldots,T$. 同理假设 $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle$, $t = 1,\ldots,T$, 对于任意向量 $\boldsymbol{g}_t$. 定义 $\|\boldsymbol{x}\|_A := \sqrt{\boldsymbol{x}^\top A\boldsymbol{x}}$. 则, 存在 $\boldsymbol{z}_t$ 处于 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+1}$ 之间的连线上, 且 $\boldsymbol{z}_t'$ 处于 $\boldsymbol{x}_t$ 和 $\tilde{\boldsymbol{x}}_{t+1} := \operatorname{argmin}_{\boldsymbol{x}\in X}\ \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + B_{\psi_t}(\boldsymbol{x};\boldsymbol{x}_t)$ 之间的连线上, 使得下面的不等式对于任意 $\boldsymbol{u} \in V$ 成立.*

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle \leq \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x}\in V}\ \psi_1(\boldsymbol{x}) + \frac{1}{2}\sum_{t=1}^{T}\min\left(\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}_t))^{-1}}, \|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}_t'))^{-1}}\right) .$$

证明. 首先, 观察 ψ_t 是严格凸的, 因为 Hessians 是 PD. 因此, 他们可以用来定义 Bregman 散度.

从 $\boldsymbol{x}_t$ 的最优性条件来看, 有

$$\langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{v} - \boldsymbol{x}_t\rangle \geq 0,\ \forall \boldsymbol{v} \in V .$$

因此, 特别的, 有

$$\langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle \geq 0,\ \forall \boldsymbol{v} \in V .$$

利用该不等式, 有

$$B_{F_t}(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) = F_t(\boldsymbol{x}_{t+1}) - F_t(\boldsymbol{x}_t) - \langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle \leq F_t(\boldsymbol{x}_{t+1}) - F_t(\boldsymbol{x}_t) .$$

最后一个不等式表明

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= F_t(\boldsymbol{x}_t) - F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\leq F_t(\boldsymbol{x}_t) - F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) \\
&= F_t(\boldsymbol{x}_t) - F_t(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle \\
&\leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_{F_t}(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) .
\end{aligned}$$

Bregman 散度与线性项无关, 故 $B_{F_t}(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) = B_{\psi_t}(\boldsymbol{x}_{t+1};\boldsymbol{x}_t)$. 根据泰勒定理, 有 $B_{\psi_t}(\boldsymbol{x}_{t+1};\boldsymbol{x}_t) = \frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^\top\nabla^2\psi_t(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)$, 其中 $\boldsymbol{z}_t$ 是处在 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+1}$ 之间的连线上. 观察得到 $\frac{1}{2}\|\boldsymbol{x}_{t+1} -$

$\boldsymbol{x}_t\|^2_{\nabla^2\psi_t(\boldsymbol{z}_t)}$ 这确实是一个范数, 因为我们假设 ψ_t 的 Hessian 是 PD. 因此, 通过 Fenchel-Young 不等式, 我们有

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &\le \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \qquad (7.2)\\
&\le \frac{1}{2}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}_t))^{-1}} + \frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^\top \nabla^2\psi_t(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) - B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t)\\
&= \frac{1}{2}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}_t))^{-1}},
\end{aligned}$$

这至少给出了最小的第一项.

对于最小的第二项, 我们从(7.2) 开始

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &\le \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \le \max_{\boldsymbol{x}\in\mathbb{R}^d} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}\rangle - B_{\psi_t}(\boldsymbol{x}; \boldsymbol{x}_t)\\
&= \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_{t+1}\rangle - B_{\psi_t}(\tilde{\boldsymbol{x}}_{t+1}; \boldsymbol{x}_t)\ .
\end{aligned}$$

然后, 我们继续第一个边界. □

使用 ψ_t 的距离生成功能定义 $\boldsymbol{z}'_t$ 为一系列 OMD 更新, 同样, 与 OMD 的局部规范不同, $\boldsymbol{z}'_t$ 不以任何方式出现在 FTRL 算法中.

7.5 FTRL 示例: T 未知情况下的指数梯度

正如我们在 OMD 的 $sec: omd_eg$ 中所作的那样, 让我们 看看 FTRL 实例化的示例, 以线性化损失具有指数渐变的 FTRL 版本.

设 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \ge 0\}$ 以及损失函数 $\ell_t : V \to \mathbb{R}$ 序列是凸的, 以及 L_∞-Lipschitz w.r.t. the L-infinity norm. 设 $\psi : V \to \mathbb{R}^d$ 定义为 $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, 其中 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \ge 0\}$ 且定义 $0\ln 0 = 0$. 设 $\psi_t(\boldsymbol{x}) = \alpha L_\infty\sqrt{t}\psi(\boldsymbol{x})$, 即 $\alpha L_\infty\sqrt{t}$ 强凸的, w.r.t. 范数 L_1 , 其中 $\alpha > 0$ 是算法的一个参数.

考虑到正则化是强凸的, 我们知道

$$\boldsymbol{x}_t = \nabla\psi_t^\star\left(-\sum_{i=1}^{t-1}\boldsymbol{g}_i\right)\ .$$

我们已经知道 $\psi^\star(\boldsymbol{\theta}) = \ln\left(\sum_{i=1}^d \exp(\theta_i)\right)$, 这表明 $\psi_t^\star(\boldsymbol{\theta}) = \alpha L_\infty\sqrt{t}\ln\left(\sum_{i=1}^d \exp\left(\frac{\theta_i}{\alpha L_\infty\sqrt{t}}\right)\right)$. 所以, 在线性损失的情况下运行 FTRL, 有

$$x_{t,j} = \frac{\exp\left(-\frac{\sum_{k=1}^{t-1} g_{k,j}}{\alpha L_\infty\sqrt{t}}\right)}{\sum_{i=1}^d \exp\left(-\frac{\sum_{k=1}^{t-1} g_{k,i}}{\alpha L_\infty\sqrt{t}}\right)},\ j = 1, \ldots, d,$$

其中 $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$. 注意, 这是完全相同的更新 EG 基于 OMD, 但这里我们有效地使用时变学习率.

我们也知道后悔保证是

$$
\begin{aligned}
\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell(\boldsymbol{u}) &\le L_\infty \sqrt{T}\alpha \left(\sum_{i=1}^{d} u_i \ln u_i + \ln d \right) + \frac{1}{2\alpha L_\infty} \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\infty^2}{\sqrt{t}} \\
&\le L_\infty \sqrt{T} \left(\alpha \left(\sum_{i=1}^{d} u_i \ln u_i + \ln d \right) + \frac{1}{\alpha} \right) \\
&\le L_\infty \sqrt{T} \left(\alpha \ln d + \frac{1}{\alpha} \right), \ \forall \boldsymbol{u} \in V,
\end{aligned} \tag{7.3}
$$

我们利用 $\psi_t = \alpha L_\infty \sqrt{t} \psi(\boldsymbol{x})$ 和 $\psi_t = \alpha L_\infty \sqrt{t}(\psi(\boldsymbol{x}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x}))$ 等价的事实. α 的最小化悔值上界的最优选择是 $\frac{1}{\sqrt{\ln d}}$. 这种后悔保证与我们证明的 OMD 的保证相似, 但有一个重要的区别: 我们不需要事先知道 T 的轮数. 在 OMD 中, 类似的边界是空的, 因为它依赖于无穷大的 $\max_{\boldsymbol{u},\boldsymbol{x} \in V} B_\psi(\boldsymbol{u}; \boldsymbol{x})$.

正如我们在 OMD 情况中所做的那样, 我们也可以使用局部规范得到一个界. 让我们使用额外的假设: $g_{t,i} \ge 0$, 对于 $t = 1, \ldots, T$ 和 $i = 1, \ldots, d$. 利用定理7.11, 对于所有的 $\boldsymbol{u} \in V$ 有

$$
\begin{aligned}
\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell(\boldsymbol{u}) &\le L_\infty \sqrt{T}\alpha \left(\sum_{i=1}^{d} u_i \ln u_i + \ln d \right) + \frac{1}{2} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}'_t))^{-1}} \\
&= L_\infty \sqrt{T}\alpha \left(\sum_{i=1}^{d} u_i \ln u_i + \ln d \right) + \frac{1}{2\alpha L_\infty} \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi(\boldsymbol{z}'_t))^{-1}}}{\sqrt{t}},
\end{aligned}
$$

其中 $\boldsymbol{z}'_t$ 处于 $\boldsymbol{x}_t$ 和 $\tilde{\boldsymbol{x}}_{t+1}$ 之间的线段上. 在这种情况下, 对于 $i = 1, \ldots, d$, 很容易计算出 $\tilde{x}_{t+1,i}$ 为 $x_{t,i} \exp(-\frac{g_{t,i}}{\alpha L_\infty \sqrt{t}})$, 而且, $\nabla^2 \psi(\boldsymbol{z}'_t)$ 是一个对角线矩阵, 其对角线上的元素为 $\frac{1}{z'_{t,i}}, i = 1, \ldots, d$. 因此, 有

$$
\|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi(\boldsymbol{z}'_t))^{-1}} = \sum_{i=1}^{d} g_{t,i}^2 z'_{t,i} \le \sum_{i=1}^{d} g_{t,i}^2 x_{t,i},
$$

它小于或等于 $\|\boldsymbol{g}_t\|_\infty^2$, 我们已经在 (7.3)中得到.

7.6 FTRL 的例子: AdaHedge*

在本节中, 我们将解释 EG/Hedge 算法的一种变体, 称为 AdaHedge. 基本思想是设计一种算法, 它自适应于损失的平方 L_∞ 范数的和, 而不需要任何关于损失范围的先验信息.

首先, 考虑用负熵 $\psi_t(\boldsymbol{x}) = \lambda \sum_{i=1}^{d} x_i \ln x_i$ 作为常数正则化的情况, 其中 $\lambda > 0$ 将在下文中确定, 且 V 是 $\mathbb{R}^d$ 中单一的. 利用带有线性损失的正则化 FTRL, 我们立即得到

$$
\begin{aligned}
\mathrm{Regret}_T(\boldsymbol{u}) &\le \lambda(\ln d + \sum_{i=1}^{d} u_i \ln u_i) + \sum_{t=1}^{T} \left(F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right) \\
&\le \lambda \ln d + \sum_{t=1}^{T} \left(F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right),
\end{aligned}
$$

我们设定 $\boldsymbol{u}$ 的负熵的上界为 0. 利用正则化 w.r.t. L_1 范数和引理7.7的强凸性, 我们将进一步设置上界为

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \lambda \ln d + \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda} \ .$$

这表明, 最优 λ 应该为 $\lambda = \sqrt{\frac{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2}{2\ln d}}$. 然而, 正如我们在 4.2中看到的, 这种算法的任何参数的选择都是不可行的. 因此, 正如我们在4.2中所做的那样, 我们可以考虑使用该选项的在线版本

$$\psi_t(\boldsymbol{x}) = \lambda_t \sum_{i=1}^{d} x_i \ln x_i \quad \text{where} \quad \lambda_t = \frac{1}{\alpha}\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}, \tag{7.4}$$

其中 $\alpha > 0$ 是一个常数, 将在以后确定. 这种选择的一个重要性质是, 它产生了一种无标度的算法, 即它的预测 $\boldsymbol{x}_t$ 不随损失的任何常数因子的标度变化而变化. 这很容易看出来, 因为

$$x_{t,j} \propto \exp\left(-\frac{\alpha \sum_{i=1}^{t-1} g_{i,j}}{\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}}\right), \ \forall i = 1, \ldots, d \ .$$

注意, 这种选择使正则化器不随时间减少, 并立即为我们提供

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \lambda_T \ln d + \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda_t} = \frac{\ln d}{\alpha}\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2} + \alpha \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}} \ .$$

在这一点上, 我们可能想使用引理4.13 来给上界的和上界, 但不幸的是, 我们不能这样做! 的确, 分母中不包含 $\|\boldsymbol{g}_t\|_\infty^2$ 这一项. 我们可以在 λ_t 中加上一个常数, 但这会破坏算法的无标性. 然而, 事实证明, 我们仍然可以证明我们的界, 而不需要改变正则化器. 关键的观察是, 我们可以以两种不同的方式绑定术语 $F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$ 一种方法是使用引理7.7. 另一个是

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle &\leq F_t(\boldsymbol{x}_{t+1}) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \\
&= \psi_t(\boldsymbol{x}_{t+1}) + \sum_{i}^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1} \rangle - \psi_{t+1}(\boldsymbol{x}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1} \rangle \\
&\leq -\langle \boldsymbol{g}_t, \boldsymbol{x}_{t+1} \rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \\
&\leq 2\|\boldsymbol{g}_t\|_\infty,
\end{aligned}$$

我们使用了 $\boldsymbol{x}_{t+1}$ 的定义以及正则化函数不随时间减少的事实. 所以, 有

$$
\begin{aligned}
\sum_{t=1}^{T} F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle &\leq \sum_{t=1}^{T} \min\left(\frac{\alpha \|\boldsymbol{g}_t\|_\infty^2}{2\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}}, 2\|\boldsymbol{g}_t\|_\infty \right) \\
&= 2\sum_{t=1}^{T} \sqrt{\min\left(\frac{\alpha^2 \|\boldsymbol{g}_t\|_\infty^4}{16\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}, \|\boldsymbol{g}_t\|_\infty^2 \right)} \\
&\leq 2\sum_{t=1}^{T} \sqrt{\frac{2}{\frac{16\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}{\alpha^2 \|\boldsymbol{g}_t\|_\infty^4} + \frac{1}{\|\boldsymbol{g}_t\|_\infty^2}}} \\
&= 2\sum_{t=1}^{T} \sqrt{2} \frac{\alpha \|\boldsymbol{g}_t\|_\infty^2}{\sqrt{\alpha^2 \|\boldsymbol{g}_t\|_\infty^2 + 16\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}},
\end{aligned}
$$

我们使用了两个数之间的最小值小于它们的调和平均值的事实. 假设 $\alpha \geq 4$ 利用引理4.13, 有

$$
\sum_{t=1}^{T} F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \frac{\sqrt{2}}{2} \sum_{t=1}^{T} \frac{\alpha \|\boldsymbol{g}_t\|_\infty^2}{\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|_\infty^2}} \leq \sqrt{2\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2}
$$

且

$$
\mathrm{Regret}_T(\boldsymbol{u}) \leq \left(\frac{\ln d}{\alpha} + \alpha\sqrt{2} \right) \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2} \,. \tag{7.5}
$$

α 的边界, 设 $\alpha = \max(4, 2^{-1/4}\sqrt{\ln d})$. 综上所述, 我们得到了一个具有后悔界 $O(\sqrt{\ln d \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2})$ 无标度算法.

我们可以认为自己很快乐, 但在上述算法中有一个明显的问题: 时变正则器 λ_t 的选择严格依赖于我们的上界. 因此, 一个松散的边界将导致一个糟糕的正规化选择! 一般来说, 每次我们在设计一个算法时使用一部分证明, 我们不能期望一个令人兴奋的经验性能, 除非我们的上界真的很紧. 那么, 我们能设计一个更好的正规化器吗? 我们需要一个更好的上界!

设泛型正则化函数 $\psi_t(\boldsymbol{x}) = \lambda_t \psi(\boldsymbol{x})$ 其对应的正则化函数 FTRL 具有线性损失, 且有上界

$$
\mathrm{Regret}_T(\boldsymbol{u}) \leq \lambda_T(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x}\in V} \psi(\boldsymbol{x})) + \sum_{t=1}^{T} \left(F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right),
$$

其中, 假设 λ_t 在时间上不递减.

现在, 注意到这类算法的和不太可能消失, 所以我们可以尝试使 $\lambda_T(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x}\in V} \psi(\boldsymbol{x}))$ 项具有相同的和阶. 因此, 我们想设置 $\sum_{i=1}^{t} \left(F_i(\boldsymbol{x}_i) - F_{i+1}(\boldsymbol{x}_{i+1}) + \langle \boldsymbol{g}_i, \boldsymbol{x}_i \rangle \right)$ 的 λ_t 有相同的顺序. 然而, 这种方法

会导致令人讨厌的重复. 所以, 利用 λ_t 是非递减的这一事实, 我们来计算和的很小的上界为:

$$
\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle &= F_t(\boldsymbol{x}_t) - \lambda_{t+1}\psi(\boldsymbol{x}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1} \rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \\
&\leq F_t(\boldsymbol{x}_t) - \lambda_t\psi(\boldsymbol{x}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1} \rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \\
&\leq F_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in V} \left(\lambda_t \psi(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle \right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle := \delta_t \ .
\end{aligned}
$$

对于 $t \geq 2$, 设 $\lambda_t = \frac{1}{\alpha^2} \sum_{i=1}^{t-1} \delta_i$, $\lambda_1 = 0$, 和 $\boldsymbol{x}_1 = \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi(\boldsymbol{x})$. 即表明

$$
\operatorname{Regret}_T(\boldsymbol{u}) \leq \left(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x} \in V} \psi(\boldsymbol{x}) + \alpha^2 \right) \lambda_{T+1} \ .
$$

设 ψ 等于负的熵, 我们得到一个叫做 AdaHedge 算法.

有了这种正则化的选择, 我们可以简化一点 δ_t 的表达式. 对于 $t = 1$, 有 $\delta_1 = \langle \boldsymbol{g}_1, \boldsymbol{x}_1 \rangle - \min_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_1, \boldsymbol{x} \rangle$. 相反, 对于 $t \geq 2$, 利用 Fenchel 共轭函数的性质, 有

$$
\delta_t = \lambda_t \ln \frac{\sum_{j=1}^{d} \exp\left(\theta_{t+1,j}/\lambda_t\right)}{\sum_{j=1}^{d} \exp\left(\theta_{t,j}/\lambda_t\right)} + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle = \lambda_t \ln \left(\sum_{j=1}^{d} x_{t,j} \exp\left(-g_{t,j}/\lambda_t\right) \right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \ .
$$

总的来说, 我们得到算法 7.3中的 AdaHedge 伪代码.

Algorithm 7.3 AdaHedge

Require: $\alpha > 0$

$\lambda_1 = 0$

$\boldsymbol{x}_1 = [1/d, \ldots, 1/d] \in \mathbb{R}^d$

$\boldsymbol{\theta}_1 = \boldsymbol{0} \in \mathbb{R}^d$

for $t = 1$ **to** T **do**

  Output $\boldsymbol{x}_t$

  Receive $\boldsymbol{g}_t \in \mathbb{R}^d$ and pay $\langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$

  Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \boldsymbol{g}_t$

  Set $\delta_t = \begin{cases} \langle \boldsymbol{g}_1, \boldsymbol{x}_1 \rangle - \min_{j=1,\ldots,d} \ g_{t,j}, & t = 1 \\ \lambda_t \ln \left(\sum_{j=1}^{d} x_{t,j} \exp\left(-g_{t,j}/\lambda_t\right) \right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle & \text{otherwise} \end{cases}$

  Update $\lambda_{t+1} = \lambda_t + \frac{1}{\alpha^2} \delta_t$

  Update $x_{t+1,j} \propto \exp\left(\frac{\theta_{t+1,j}}{\lambda_{t+1}} \right), j = 1, \ldots, d$

end for

现在, 我们需要 λ_T 的上界. 观察 $\lambda_{t+1} = \lambda_t + \frac{1}{\alpha^2}\delta_t$. 此外, 正如我们以前所做的, 我们可以用两种不同的方法来计算 δ_t 的上界, 实际上, 由引理7.7, 当 $t \geq 2$ 时, 有 $\delta_t \leq \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda_t}$. 由 $\tilde{\boldsymbol{x}}_{t+1} =$

$\operatorname{argmin}_{\boldsymbol{x}\in V}\ \lambda_t\psi(\boldsymbol{x})+\sum_{i=1}^t\langle \boldsymbol{g}_i,\boldsymbol{x}\rangle$ 可知, 有

$$\begin{aligned}\delta_t &= F_t(\boldsymbol{x}_t)-\min_{\boldsymbol{x}\in V}\left(\lambda_t\psi(\boldsymbol{x})+\sum_{i=1}^t\langle \boldsymbol{g}_i,\boldsymbol{x}\rangle\right)+\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle = F_t(\boldsymbol{x}_t)-\lambda_t\psi(\tilde{\boldsymbol{x}}_{t+1})-\sum_{i=1}^t\langle \boldsymbol{g}_i,\tilde{\boldsymbol{x}}_{t+1}\rangle+\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle\\ &\le F_t(\tilde{\boldsymbol{x}}_{t+1})-\lambda_t\psi(\tilde{\boldsymbol{x}}_{t+1})-\sum_{i=1}^t\langle \boldsymbol{g}_i,\tilde{\boldsymbol{x}}_{t+1}\rangle+\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle = -\langle \boldsymbol{g}_t,\tilde{\boldsymbol{x}}_{t+1}\rangle+\langle \boldsymbol{g}_t,\boldsymbol{x}_t\rangle \le 2\|\boldsymbol{g}_t\|_\infty\ .\end{aligned}$$

因此, 有

$$\begin{aligned}\lambda_1 &= 0,\\ \lambda_2 &\le 2\alpha\|\boldsymbol{g}_1\|_\infty,\\ \lambda_{t+1} &= \lambda_t+\frac{\delta_t}{\alpha^2}\le \lambda_t+\frac{1}{\alpha^2}\min\left(2\|\boldsymbol{g}_t\|_\infty,\frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda_t}\right),\ \forall t\ge 2\ .\end{aligned}$$

我们可以用下面的引理来解这个递归式, 其中 $\Delta_t=\lambda_t$ 和 $a_t=\|\boldsymbol{g}_t\|_\infty$.

引理 7.12. 设 $\{a_t\}_{t=1}^\infty$ 是任意非负实数序列.

设 $\{\Delta_t\}_{t=0}^\infty$ 是任意非负实数序列满足

$$\Delta_0=0 \qquad \textit{and} \qquad \Delta_t\le\Delta_{t-1}+\min\left\{ba_t,\ c\frac{a_t^2}{2\Delta_{t-1}}\right\} \quad \textit{for any } t\ge 1\ .$$

则, 对于任意 $T\ge 0$, $\Delta_T\le\sqrt{(b^2+c)\sum_{t=1}^T a_t^2}$.

证明. 有

$$\Delta_T^2=\sum_{t=1}^T(\Delta_t^2-\Delta_{t-1}^2)=\sum_{t=1}^T\left[(\Delta_t-\Delta_{t-1})^2+2(\Delta_t-\Delta_{t-1})\Delta_{t-1}\right]\ .$$

我们分别把和式中的每一项限定在一起. 在 Δ_t 的定义中, 最小不等式的左项为 $(\Delta_t-\Delta_{t-1})^2\le b^2a_t^2$, 然而右项为 $2(\Delta_t-\Delta_{t-1})$, , $\Delta_{t-1}\le ca_t^2$. $\Delta_T^2\le(b^2+c)\sum_{t=1}^T a_t^2$. □

所以, 我们总结得出

$$\mathrm{Regret}_T(\boldsymbol{u})\le\left(\psi(\boldsymbol{u})-\inf_{\boldsymbol{x}\in V}\psi(\boldsymbol{x})+\alpha^2\right)\lambda_{T+1}\le\left(\frac{\psi(\boldsymbol{u})-\inf_{\boldsymbol{x}\in V}\psi(\boldsymbol{x})}{\alpha^2}+1\right)\sqrt{(4+\alpha^2)\sum_{t=1}^T\|\boldsymbol{g}_t\|_\infty^2},$$

设 $\alpha=\sqrt{\ln d}$, 有

$$\mathrm{Regret}_T(\boldsymbol{u})\le 2\sqrt{(4+\ln d)\sum_{t=1}^T\|\boldsymbol{g}_t\|_\infty^2}\ .$$

注意, 这与 (7.5)中的后悔大致相同, 但非常重要的区别是, 新的后悔界依赖于更紧的 λ_{T+1}, 我们的上界是 $O(\sqrt{\sum_{t=1}^T\|\boldsymbol{g}_t\|_\infty^2})$, 但通常会比它小得多. 例如, λ_{T+1} 可以用更严格的局部规范上界, 见7.5. 相反, 在第一个解中, 后悔总是被 $\sqrt{\sum_{t=1}^T\|\boldsymbol{g}_t\|_\infty^2}$ 所支配, 因为我们在正则化中明确地使用了它.

我们可以从 AdaHedge 中学到一个重要的教训: 后悔并不是全部, 具有相同最坏情况保证的算法可以表现出截然不同的经验行为. 不幸的是, 这一信息很少被听到, 而且有一部分人过于关注最坏情况的保证, 而不是实证表现. 更糟糕的是, 有时人们喜欢具有“更优雅的分析”的算法, 而完全忽略了可能更糟糕的经验性能.

7.7 复合损失

现在让我们看看损失线性化的一种变体: 复合损失的部分线性化.

假设我们接收到的损失由两项组成: 一项是随时间变化的凸函数, 另一项是固定且已知的. 这些损失被称为复合损失. 例如, 我们可以有 $\ell_t(\boldsymbol{x}) = \tilde{\ell}_t(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_1$. 使用线性化, 我们可以取 ℓ_t. 的次梯度. 然而, 在这种特殊情况下, 我们可能会失去 L_1 范数产生稀疏解的能力.

有一个更好的方法来处理这类损失: 将损失的常数部分移到正规化项内. 这样, 这部分就不会被线性化, 而是在更新的 argmin 中精确地使用. 假设 argmin 仍然很容易计算, 那么这种方法总能获得更好的性能. 特别是, 在向损失添加 L_1 范数的情况下, 您将在每个步骤中使用 L_1 正则化优化问题的解决方案进行预测.

实际上, 在上面的例子中, 我们将定义 $\psi_t(\boldsymbol{x}) = L\sqrt{t}(\psi(\boldsymbol{x}) - \min_{\boldsymbol{y}}\ \psi(\boldsymbol{y})) + \lambda t\|\boldsymbol{x}\|_1$, 其中设 ψ to be 1-strongly convex 以及损失函数 $\tilde{\ell}_t$ 是 L-Lipschitz. 注意, 我们在时刻 t 使用 $\lambda t\|\boldsymbol{x}\|_1$ 因为我们假设. 已知 $\psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1}\ell_t(\boldsymbol{x})$ 是 $L\sqrt{t}$ 强凸的, 使用定理7.7, 有

$$
\begin{aligned}
&\sum_{t=1}^{T}\tilde{\ell}_t(\boldsymbol{x}_t) - \sum_{t=1}^{T}\tilde{\ell}_t(\boldsymbol{u}) \\
&\quad\le \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \\
&\quad\le L\sqrt{T}\left(\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}}\ \psi(x)\right) + \lambda T\|\boldsymbol{u}\|_1 - \min_{\boldsymbol{x}}(\lambda\|\boldsymbol{x}\|_1 + \psi(\boldsymbol{x}) - \min_{\boldsymbol{y}}\ \psi(\boldsymbol{y})) + \frac{1}{2}\sum_{t=1}^{T}\frac{\|\boldsymbol{g}_t\|_\star^2}{L\sqrt{t}} - \lambda\sum_{t=1}^{T-1}\|\boldsymbol{x}_{t+1}\|_1 \\
&\quad\le L\sqrt{T}\left(1 + \psi(\boldsymbol{u}) - \min_{\boldsymbol{x}}\ \psi(x)\right) + \lambda T\|\boldsymbol{u}\|_1 - \lambda\|\boldsymbol{x}_1\|_1 - \lambda\sum_{t=1}^{T-1}\|\boldsymbol{x}_{t+1}\|_1 \\
&\quad= L\sqrt{T}\left(1 + \psi(\boldsymbol{u}) - \min_{\boldsymbol{x}}\ \psi(x)\right) + \lambda T\|\boldsymbol{u}\|_1 - \lambda\sum_{t=1}^{T}\|\boldsymbol{x}_t\|_1,
\end{aligned}
$$

其中 $\boldsymbol{g}_t \in \partial\tilde{\ell}_t(\boldsymbol{x}_t)$. 重新排序这些项, 有

$$
\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T}\ell_t(\boldsymbol{u}) = \sum_{t=1}^{T}(\lambda\|\boldsymbol{x}_t\|_1 + \tilde{\ell}_t(\boldsymbol{x}_t)) - \sum_{t=1}^{T}(\lambda\|\boldsymbol{u}\|_1 + \tilde{\ell}_t(\boldsymbol{u})) \le L\sqrt{T}\left(\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}}\psi(\boldsymbol{x}) + 1\right)\ .
$$

例 7.13. 当 $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ 时, L_1 范数的复合损失也会出现更新, 有

$$
\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}}\ \frac{L\sqrt{t}}{2}\|\boldsymbol{x}\|_2^2 + \lambda t\|\boldsymbol{x}\|_1 + \sum_{i=1}^{t-1}\langle \boldsymbol{g}_i, \boldsymbol{x}\rangle\ .
$$

我们可以解决这个问题, 观察到最小值在 $\boldsymbol{x}$ 的每个坐标上分解. 表示为 $\boldsymbol{\theta}_t = \sum_{i=1}^{t-1}\boldsymbol{g}_i$. 因此, 我们从一阶最优条件知道 $x_{t,i}$ 是 i 坐标的解, 如果存在 $v_i \in \partial|x_{t,i}|$ 使得

$$
L\sqrt{t}x_{t,i} + \lambda t v_i + \theta_{t,i} = 0
$$

考虑 *3* 中不同的情况:

- $|\theta_{t,i}| \le \lambda t$, *then* $x_{t,i} = 0$ *and* $v_i = -\frac{\theta}{\lambda t}$.
- $\theta_{t,i} > \lambda t$, *then* $x_{t,i} = -\frac{\theta_{t,i} - \lambda t}{L\sqrt{t}}$ *and* $v_i = -1$.
- $\theta_{t,i} < -\lambda t$, *then* $x_{t,i} = -\frac{\theta_{t,i} + \lambda t}{L\sqrt{t}}$ *and* $v_i = 1$.

所以, 总之有

$$x_{t,i} = -\frac{\operatorname{sign}(\theta_{t,i}) \max(|\theta_{t,i}| - \lambda t, 0)}{L\sqrt{t}} \ .$$

观察这个更新将产生稀疏解, 而仅仅取 L_1 范数的次梯度永远不会产生稀疏预测.

附注 7.14 (Proximal operators)**.** 在上面的例子中, 我们计算了类似于

$$\operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \ \|\boldsymbol{x}\|_1 - \langle \boldsymbol{v}, \boldsymbol{x}\rangle + \frac{1}{2}\|\boldsymbol{x}\|_2^2 = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \ \|\boldsymbol{x}\|_1 - \langle \boldsymbol{v}, \boldsymbol{x}\rangle + \frac{1}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\|\boldsymbol{v}\|_2^2 = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \ \|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2 \ .$$

这个操作在优化文献中被称为 L_1 范数的 Proximal Operator . 一般情况下, 一个凸的、真的、闭的近端操作函数 $f : \mathbb{R}^d \to (-\infty, +\infty]$ 被定义为

$$\operatorname{Prox}_f(\boldsymbol{v}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \ f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2 \ .$$

在优化中使用近端操作符的方式与我们使用它的方式相同: 它们允许最小化整个函数, 而不是它的线性近似. 同时, 近端算子推广了欧几里得投影的概念. 实际上, $\operatorname{Prox}_{i_V}(\boldsymbol{v}) = \Pi_V(\boldsymbol{v})$.

7.8 FTRL Regret Bound with Proximal Regularizers

让我们回到 FTRL 后悔约束我们看看在正则化函数接近的情况下能否加强它, 它满足 $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}} \ \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$.

引理 7.15. 由 $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$ 表示. 假设 $\operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \ F_t(\boldsymbol{x})$ 非空且集合 $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \ F_t(\boldsymbol{x})$. 同理, 假设 F_t 是 λ_t 强凸, *w.r.t.* $\|\cdot\|$, ℓ_t 凸的, 正则化是 $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}} \ \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$. 同样, 假设 $\partial\ell_t(\boldsymbol{x}_t)$ 非空. 有

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) \le \frac{\|\boldsymbol{g}_t\|_*^2}{2\lambda_{t+1}} + \psi_t(\boldsymbol{x}_t) - \psi_{t+1}(\boldsymbol{x}_t), \ \forall \boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t) \ .$$

证明. 有

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t) + \psi_{t+1}(\boldsymbol{x}_t) - \psi_t(\boldsymbol{x}_t)) - F_{t+1}(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_t) + \psi_t(\boldsymbol{x}_t) \\
&= F_{t+1}(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_t) + \psi_t(\boldsymbol{x}_t) \\
&\le \frac{\|\boldsymbol{g}_t'\|_*^2}{2\lambda_{t+1}} - \psi_{t+1}(\boldsymbol{x}_t) + \psi_t(\boldsymbol{x}_t),
\end{aligned}$$

在不等式中, 我们使用推论7.6, 事实是 $\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x}} \ F_{t+1}(\boldsymbol{x})$, 并且 $\boldsymbol{g}_t' \in \partial F_{t+1}(\boldsymbol{x}_t)$. 从近端属性来看, 我们有 $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}} \ F_t(\boldsymbol{x}) + \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$, $\mathbf{0} \in \partial(F_t(\boldsymbol{x}_t) + \psi_{t+1}(\boldsymbol{x}_t) - \psi_t(\boldsymbol{x}_t))$. 因此, 使用函数和的次微分定理, 并记住 $F_{t+1}(\boldsymbol{x}) = F_t(\boldsymbol{x}) + \ell_t(\boldsymbol{x}) + \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$, 我们选择 $\boldsymbol{g}_t'$ 则, 有 $\boldsymbol{g}_t' \in \partial\ell_t(\boldsymbol{x}_t)$. □

附注 7.16. *注意, 常量正则化是近端, 因为任何点都是零函数的最小值. 另一方面, 常数正则化使两个引理相同, 除非损失函数对总强凸性有贡献.*

现在我们将使用上述引理来证明强凸损失的对数后悔界.

推论 7.17. *设对于 $t = 1, \ldots, T$, ℓ_t 是 μ_t 强凸, w.r.t. $\|\cdot\|$, . 将正则化器的序列设置为零. 那么, FTL 保证以下悔值*

$$\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{1}{2} \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\star^2}{\sum_{i=1}^{t} \mu_i}, \ \forall \boldsymbol{u} \in \mathbb{R}^d, \ \forall \boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t) \ .$$

以上的后悔保证与强凸损失的 OMD 相同, 但在这里我们不需要知道损失的强凸性. 事实上, 我们只需要在过去的损失上输出最小值. 然而, 正如我们上次注意到的, 这可能是不可取的, 因为现在每个更新都是一个优化问题.

因此, 我们可以再次使用用一个简单的替代品来替代损失的想法. 在 Lipschitz 的情况下, 使用线性损失是有意义的. 然而, 在这里你可以做得更好, 使用二次损失, 因为损失是强凸性的. 因此, 我们可以在二次损失 $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{\mu_t}{2} \|\boldsymbol{x} - \boldsymbol{x}_t\|^2$ 的情况下运行 FTRL , 其中 $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. 算法如下

Algorithm 7.4 Follow-the-Regularized-Leader on "Quadratized" Losses

Require: A sequence of regularizers $\psi_1, \ldots, \psi_T : \mathbb{R}^d \to (-\infty, +\infty]$

```
for t = 1 to T do
    Output x_t ∈ argmin_{x∈R^d} ψ_t(x) + Σ_{i=1}^{t-1} (⟨g_i, x⟩ + (μ_t/2)‖x − x_i‖²)
    Receive ℓ_t : R^d → (−∞, +∞] and pay ℓ_t(x_t)
    Set g_t ∈ ∂ℓ_t(x_t)
    Calculate the strong convexity μ_t of ℓ_t
end for
```

为了了解为什么这是一个好主意, 考虑一下损失是 L_2 范数强凸 w.r.t. . 更新变为:

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}} \sum_{i=1}^{t-1} \left(\langle \boldsymbol{g}_i, \boldsymbol{x} \rangle + \frac{\mu_i}{2} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2^2 \right) = \frac{\sum_{i=1}^{t-1} (\mu_i \boldsymbol{x}_i - \boldsymbol{g}_i)}{\sum_{i=1}^{t-1} \mu_i} \ . \tag{7.6}$$

此外, 我们将得到与推论7.17中完全相同的后悔约束, 唯一的区别是, 在这里, 保证适用于 $\boldsymbol{g}_t$ 中的某个特定的选择, 而不适用于 $\partial \ell_t(\boldsymbol{x}_t)$ 中的任何子梯度.

例 7.18. *回到第一章中的例子, 其中 $V = [0, 1]$ 以及 $\ell_t(\boldsymbol{x}) = (x - y_t)^2$ 是强凸的, 我们现在马上看到没有正则化的 FTRL, 也就是跟随前导函数, 给出了对数后悔. 注意, 在这种情况下, 损失只在 V 上定义, 所以最小值在 V 上也保留.*

7.9 Online Newton Step

上次我们看到, 强凸性的概念允许我们构建二次代理损失函数,FTRL 在这个函数上有较小的后悔. 我们能不能找到一个更普遍的强凸性的概念让我们在更大的函数类中得到一个小小的后悔? 我们可以

从强凸性入手, 试着推广它. 所以, 我们不要求函数是强凸的, 而是要求在特定点上具有强凸性, 一个依赖于点本身的模.

特别地, 我们可以得到, 对于每一个损失 ℓ_t , 以及对于所有的 $\boldsymbol{x} \in \operatorname{dom} \ell_t$, 以下成立

$$\forall \boldsymbol{x} \in V, \boldsymbol{g} \in \partial \ell_t(\boldsymbol{x}), \exists A_t \in \mathbb{R}^{d \times d} \ : \ A_t \succeq 0, \ \ell_t(\boldsymbol{y}) \geq \ell_t(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_{A_t}^2, \forall \boldsymbol{y} \in V,$$

其中 $\|\boldsymbol{x}\|_{A_t}$ 被定义为 $\sqrt{\boldsymbol{x}^\top A_t \boldsymbol{x}}$. 请注意, 这是一个比强凸性弱的性质, 因为 A_t 依赖于 $\boldsymbol{x}$. 另一方面, 在强凸性的定义中, 我们希望最后一项在空间的任何地方都是相同的范数 (或更普遍的 Bregman 散度). 这个新定义的基本原理是, 它仍然允许我们建立替代损失函数, 但不需要在整个空间具有强凸性. 因此, 我们可以考虑在替代损失上使用 FTRL

$$\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$$

以及近端正规化器 $\psi_t(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|^2 + \frac{1}{2}\sum_{i=1}^{t-1} \|\boldsymbol{x}_i - \boldsymbol{x}\|_{A_i}^2$, 其中 $\lambda > 0$. 我们用 $S_t = \lambda I_d + \sum_{i=1}^{t} A_i$.

附注 7.19. *注意 $\|\boldsymbol{x}\|_{S_t} := \sqrt{\boldsymbol{x}^\top S_t \boldsymbol{x}}$ 是一个范数, 因为 S_t 是正定的 (PD) 且 $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top S_t \boldsymbol{x}$ 是 1 是强凸的 w.r.t. $\|\cdot\|_{S_t}$ 被定义为 $\|\boldsymbol{x}\|_{S_t} = \sqrt{\boldsymbol{x}^\top S_t \boldsymbol{x}}$ (因为 Hessian 是 S_t 以及 $\boldsymbol{x}^\top \nabla^2 f(\boldsymbol{y}) \boldsymbol{x} = \|\boldsymbol{x}\|_{A_t}^2$). 同时, $\|\cdot\|_{S_t}$ 的双重标准是 $\|\cdot\|_{S_t^{-1}}$.*

由上述知, 正则化器 $\psi_t(\boldsymbol{x})$ 是 1-strongly convex w.r.t $\|\cdot\|_{S_{t-1}}$. 因此, 使用引理7.1 和引理 7.15 中关于近端正则化器的 FTRL 后悔等式, 我们立即得到以下保证

$$\begin{aligned}
\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle &= \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{u}) \\
&\leq \psi_{T+1}(\boldsymbol{u}) + \frac{1}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_{S_t^{-1}}^2 + \sum_{t=1}^{T} \left(\psi_t(\boldsymbol{x}_t) - \psi_{t+1}(\boldsymbol{x}_t)\right) \\
&= \frac{\lambda}{2}\|\boldsymbol{u}\|^2 + \frac{1}{2}\sum_{t=1}^{T} \|\boldsymbol{x}_t - \boldsymbol{u}\|_{A_t}^2 + \frac{1}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_{S_t^{-1}}^2 \ .
\end{aligned}$$

所以, 重新排序这些项

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^{T} \left(\langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle - \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}_t\|_{A_t}^2 \right) \leq \frac{\lambda}{2}\|\boldsymbol{u}\|^2 + \frac{1}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_{S_t^{-1}}^2 \ . \tag{7.7}$$

请注意证明和算法是如何反映我们在7.8中对带有强凸损失的 FTRL 所做的事情的.

附注 7.20. *我们有可能将我们关于近端正则化 FTRL 的引理推广到这个强凸的广义概念中. 这将允许在满足正则化 $\psi(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2$ 的情况下超出原始损失运行 FTRL 得到完全相同的极限.*

现在让我们来看看这个想法的一个实例. 考虑我们接收到的损失函数序列满足的情况

$$\ell_t(\boldsymbol{x}) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{u} \rangle - \frac{\mu}{2}(\langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{u} \rangle)^2, \ \forall \boldsymbol{x}, \boldsymbol{u} \in V, \boldsymbol{g} \in \partial \ell_t(\boldsymbol{x}) \ . \tag{7.8}$$

换句话说, 我们假设有一类函数的上界是依赖于当前次梯度的二次函数. 特别地, 这些函数只在 (任何) 次梯度的方向上具有曲率. 用 $A_t = \mu \boldsymbol{g}_t \boldsymbol{g}_t^\top$ 表示, 我们可以使用上面的思想

$$\psi_t(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|^2 + \frac{1}{2}\sum_{i=1}^{t-1}\|\boldsymbol{x}_i - \boldsymbol{x}\|_{A_i}^2 = \frac{\lambda}{2}\|\boldsymbol{x}\|^2 + \frac{\mu}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{g}_i, \boldsymbol{x} - \boldsymbol{x}_i\rangle)^2 \ .$$

因此, 跟新的规则为

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}\in V} \sum_{i=1}^{t-1}\langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{g}_i, \boldsymbol{x} - \boldsymbol{x}_i\rangle)^2 \ .$$

我们得到了如下算法, 称为 Online Newton Step (ONS).

Algorithm 7.5 Online Newton Step

Require: $V \subset \mathbb{R}^d$ closed non-empty convex set, $\lambda, \mu > 0$

for $t = 1$ **to** T **do**
  Set $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}\in V} \sum_{i=1}^{t-1}\langle \boldsymbol{g}_i, \boldsymbol{x}\rangle + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{g}_i, \boldsymbol{x} - \boldsymbol{x}_i\rangle)^2$
  Receive ℓ_t and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial\ell(\boldsymbol{x}_t)$
end for

用 $S_t = \lambda I_d + \sum_{i=1}^t A_i$ 表示, 且利用 (7.7), 有

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|_{S_t^{-1}}^2 \ .$$

为了限定最后一项, 我们将使用下面的引理

引理 7.21 ([20, Lemma 11.11 and Theorem 11.7])**.** *设 $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T$ 是 $\mathbb{R}^d$ 和 $\lambda > 0$ 上的向量序列. 定义 $H_t = \lambda I_d + \sum_{i=1}^t \boldsymbol{z}_i\boldsymbol{z}_i^\top$. 则, 下式成立*

$$\sum_{t=1}^T \boldsymbol{z}_t^\top H_t^{-1}\boldsymbol{z}_t \le \sum_{i=1}^d \ln\left(1 + \frac{\lambda_i}{\lambda}\right),$$

其中 $\lambda_1, \ldots, \lambda_d$ 是 $H_T - \lambda I_d$ 的特征值.

把所有的东西放在一起, 然后假设 $\|\boldsymbol{g}_t\|_2 \le L$ 且 (7.8)对于损失成立, 则 ONS 满足以下悔值

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2\mu}\sum_{i=1}^d \ln\left(1 + \frac{\lambda_i}{\lambda}\right) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{d}{2\mu}\ln\left(1 + \frac{\mu T L^2}{d\lambda}\right),$$

在第二个不等式中, 我们使用算术和几何均值不等式 $(\prod_{i=1}^d x_i)^{1/d} \le \frac{1}{d}\sum_{i=1}^d x_i$, 事实是 $\sum_{i=1}^d \lambda_i \le \mu T L^2$.

因此, 如果损失满足 (7.8), 我们可以保证一个对数的后悔. 然而, 与强凸情况不同的是, 这里更新的复杂度至少是维数的二次. 而且, 后悔也线性地取决于维度的数量.

附注 7.22. *尽管名字叫 ONS 算法, 但不要把它和牛顿算法混淆了. 它们在精神上是相似的, 因为它们都构造了函数的二次近似, 但牛顿算法使用精确的 Hessian, 而 ONS 使用的近似只适用于一类受限的函数. 从这个角度来看,ONS 算法更类似于准牛顿方法. 然而, 理解 ONS 的最佳视角仍然是通过广义的强凸性概念.*

现在让我们看一个满足 (7.8)的函数示例.

例 7.23 (Exp-Concave Losses)**.** *定义 $X \subseteq \mathbb{R}^d$ 是凸的, 函数 $f : X \to \mathbb{R}$ 是α-**exp-concave** 如果 $\exp(-\alpha f(\boldsymbol{x}))$ 是凹的.*

选择 $\beta \le \frac{\alpha}{2}$ 使得对于所有的 $\boldsymbol{x}, \boldsymbol{y} \in X$ 和 $\boldsymbol{g} \in \partial f(\boldsymbol{x})$, 有 $|\beta\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle| \le \frac{1}{2}$, 注意, 我们需要一个有界域 β 的存在. 那么, 这类函数满足性质 (7.8). 实际上, 已知 f 是 α-exp-concave , 则 2β-exp-concave. 因此, 根据定义有

$$\exp(-2\beta f(\boldsymbol{y})) \le \exp(-2\beta f(\boldsymbol{x})) - 2\beta \exp(-2\beta f(\boldsymbol{x}))\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle,$$

即

$$\exp(-2\beta f(\boldsymbol{y}) + 2\beta f(\boldsymbol{x})) \le 1 - 2\beta\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle,$$

说明

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \le \frac{1}{2\beta}\ln\left(1 + 2\beta\langle \boldsymbol{g}, \boldsymbol{x}-\boldsymbol{y}\rangle\right) \le \langle \boldsymbol{g}, \boldsymbol{x}-\boldsymbol{y}\rangle - \frac{\beta}{2}(\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle)^2,$$

其中, 我们用的初等不等式为 $\ln(1+x) \le x - x^2/4$, for $|x| \le 1$.

例 7.24. *设 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \le U\}$. 线性预测的逻辑逻辑损失为 $\ell(\boldsymbol{x}) = \ln(1+\exp(-\langle \boldsymbol{z}, \boldsymbol{x}\rangle))$, 其中 $\|\boldsymbol{z}\|_2 \le 1$ is $\exp(-2U)/2$-exp-concave.*

7.10 Online Regression: Vovk-Azoury-Warmuth Forecaster

现在, 让我们考虑 $\ell_t(\boldsymbol{x}) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle - y_t)^2$ 和 $V = \mathbb{R}^d$ 的特定情况, 这是无约束的带有平方损失的在线线性回归之一. 这些损失不是强凸的 w.r.t. $\boldsymbol{x}$, 但当域有界时, 他们是成指数凹的. 我们可以使用 ONS 算法, 但它不适用于无界情况. 另一种可能是运行 ftl, 但损失不是很强的凸, 我们只会得到一个 $O(\sqrt{T})$ 悔值.

事实证明, 我们仍然可以得到一个对数的后悔, 如果我们做额外的假设! 在预测 $\boldsymbol{x}_t$ 之前, 我们将假设能够访问 $\boldsymbol{z}_t$, 请注意, 在大多数有趣的应用程序中, 这是一个温和的假设. 然后, 该算法将只对过去的损失加上接收到的标签为 0 的 $\boldsymbol{z}_t$ 损失进行 FTRL 具体细节见算法7.6.

Algorithm 7.6 Vovk-Azoury-Warmuth Forecaster

Require: $\lambda > 0$

for $t = 1$ **to** T **do**
  Receive $\boldsymbol{z}_t \in \mathbb{R}^d$
  Set $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}} \ \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle - y_i)^2 + \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2$
  Receive $y_t \in \mathbb{R}$ and pay $\ell(\boldsymbol{x}_t) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2$
end for

正如我们处理复合损失时所做的那样, 我们仔细查看损失函数, 看看是否有我们可以在正则化器内移动的条款. 其动机将与综合损失案例相同: 界限将仅取决于正则化之外部分损失的次梯度.

故, 观察得

$$\ell_t(\boldsymbol{x}) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2 - y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle + \frac{1}{2}y_t^2 \ .$$

从以上, 可以看到, 我们可以考虑在正则化器中移动 $\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle)^2$ 项, 且在损失中留下线性项: $\tilde{\ell}_t(\boldsymbol{x}) = -y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle$. 因此, 我们利用:

$$\psi_t(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \left(\sum_{i=1}^t \boldsymbol{z}_i\boldsymbol{z}_i^\top\right)\boldsymbol{x} + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 \ .$$

注意, t 时刻的正则化器,contains the $\boldsymbol{z}_t$ that is revealed to the algorithm before it makes its prediction. For simplicity of notation, denote by $S_t = \lambda I_d + \sum_{i=1}^t \boldsymbol{z}_i\boldsymbol{z}_i^\top$.

使用这样的程序, 预测可以以封闭的形式写成:

$$\begin{aligned}
\boldsymbol{x}_t &= \operatorname*{argmin}_{\boldsymbol{x}} \ \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle - y_i)^2 + \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2 \\
&= \left(\lambda I_d + \sum_{i=1}^t \boldsymbol{z}_i\boldsymbol{z}_i^\top\right)^{-1}\sum_{i=1}^{t-1} y_i\boldsymbol{z}_i \ .
\end{aligned}$$

因此, 利用我们证明的关于带强凸正则化的 FTRL 和 $\psi_{T+1} = \psi_T$, 有以下保证

$$\begin{aligned}
\sum_{t=1}^T \left(\tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{u})\right) &= \sum_{t=1}^T (-y_t\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle + y_t\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle) \\
&\le \psi_T(\boldsymbol{u}) - \min_{\boldsymbol{x}} \ \psi_T(\boldsymbol{x}) + \frac{1}{2}\sum_{t=1}^T y_t^2\boldsymbol{z}_t^\top S_t^{-1}\boldsymbol{z}_t + \sum_{t=1}^{T-1}(\psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1})) \\
&= \frac{1}{2}\boldsymbol{u}^\top S_T\boldsymbol{u} + \frac{1}{2}\sum_{t=1}^T y_t^2\boldsymbol{z}_t^\top S_t^{-1}\boldsymbol{z}_t - \frac{1}{2}\sum_{t=1}^{T-1}(\langle \boldsymbol{z}_{t+1}, \boldsymbol{x}_{t+1}\rangle)^2 \ .
\end{aligned}$$

注意 $\boldsymbol{x}_1 = \boldsymbol{0}$ 并且重新排序, 有

$$\begin{aligned}
\sum_{t=1}^T \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2 - \sum_{t=1}^T \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle - y_t)^2 &= \frac{1}{2}\sum_{t=1}^T(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle)^2 + \sum_{t=1}^T(-y_t\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle + y_t\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle) - \frac{1}{2}\sum_{t=1}^T(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle)^2 \\
&\le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\sum_{t=1}^T y_t^2\boldsymbol{z}_t^\top S_t^{-1}\boldsymbol{z}_t \ .
\end{aligned}$$

附注 7.25. *请注意, 与 ONS 算法不同的是, 这里的正则化函数不是近似值. 然而, 由于当前的样本 $\boldsymbol{z}_t$ 用于正则化, 所以我们得到 S_t^{-1} 在界内, 在不知道 $\boldsymbol{z}_t$ 的情况下, 最后一项将是 $\frac{1}{2}\sum_{t=1}^T y_t^2\boldsymbol{z}_t^\top S_{t-1}^{-1}\boldsymbol{z}_t$, 如果没有对 $\boldsymbol{z}_t$ 的额外假设, 则无法控制.*

故, 再次利用7.21 和假设 $|y_t| \le Y,\ t = 1, \ldots, T$, 有

$$\sum_{t=1}^T \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2 - \sum_{t=1}^T \frac{1}{2}\sum_{t=1}^T(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle - y_t)^2 \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{Y^2}{2}\sum_{i=1}^d \ln\left(1 + \frac{\lambda_i}{\lambda}\right),$$

其中 $\lambda_1, \ldots, \lambda_d$ 是 $\sum_{i=1}^T \boldsymbol{z}_i\boldsymbol{z}_i^\top$ 的特征值.

若假设 $\|\boldsymbol{z}_t\|_2 \le R$, 我们可以像在 ONS 中的做的类似项一样进行推理, 得到

$$\sum_{i=1}^{d} \ln\left(1+\frac{\lambda_i}{\lambda}\right) \le d\ln\left(1+\frac{R^2T}{\lambda d}\right) .$$

综上, 有以下定理

定理 7.26. *对于 $t=1,\dots,T$, 假设 $\|\boldsymbol{z}_t\|_2 \le R$ 和 $|y_t| \le Y$. 而后, 在算法7.6中国使用预测策略, 有*

$$\sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2 - \sum_{t=1}^{T}\sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle - y_t)^2 \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{dY^2}{2}\ln\left(1+\frac{R^2T}{\lambda d}\right) .$$

附注 7.27. *有可能表明 Vovk-Azoury-Warmuth 预测器的后悔相对于乘法因子 [20, Theorem 11.9] 是最佳的.*

7.11 FTRL 优化

在本书中, 我们把对抗性模型作为我们的环境模型. 这让我们能够设计出既适用于这种设置, 也适用于其他良性设置的算法. 然而, 这个世界从来不会完全是敌对的. 因此, 我们可能会试图以某种方式对环境建模, 但这将使我们的算法容易受到攻击. 另一种选择是, 把数据看作是由一些可预测的过程加上敌对噪声产生的. 在这种观点下, 尝试对可预测部分建模可能是有益的, 而不损害对抗性噪声的鲁棒性.

在本节中, 我们将通过一个特定版本的跟踪正则化 Follow-The-Regularized-Leader (FTRL) 来探索这种可能性, 在这个版本中我们将预测下一个损失. 从非常直观的角度来看, 如果我们预测的损失是正确的, 我们可以预期后悔会减少. 然而, 如果我们的预测是错误的, 我们仍然希望恢复最坏情况的保证. 这种算法称为乐观 FTRL.

Algorithm 7.7 Optimistic Follow-the-Regularized-Leader

Require: Closed and non-empty set $V \subseteq \mathbb{R}^d$, a sequence of regularizers $\psi_1,\dots,\psi_T : \mathbb{R}^d \to (-\infty,+\infty]$

```
for t = 1 to T do
    Predict next loss ℓ̃_t : V → ℝ
    Output x_t ∈ argmin_{x∈V}  ψ_t(x) + ℓ̃_t(x) + Σ_{i=1}^{t-1} ℓ_i(x)
    Receive ℓ_t : ℝ^d → (−∞, +∞] and pay ℓ_t(x_t)
end for
```

乐观 FTRL 的核心思想是预测下一个损失, 并将其用于更新规则, 如算法7.7所述. 请注意, 为了便于分析, 如何生成预测并不重要. 它甚至可以由另一个在线学习程序生成!

我们来看看为什么这是个好主意. 记住,FTRL 仅仅是用先前损失的最小值加上一个时变正则化器进行预测. 让我们暂时假设我们拥有预测未来的天赋, 这样我们就能提前知道下一次的损失. 然后, 我们可以预测它的最小值, 并遭受负面的后悔. 然而, 也许我们的预见能力并没有那么强, 所以我们对下一次损失的预测可能并不准确. 在这种情况下, 一个更好的主意可能是将我们的预测损失添加到先前的损失中, 并最小化正则化的和. 如果我们对未来损失的预测是准确的, 我们期望后悔保证会改善. 与此同时,

如果预测是错误的, 我们预计其影响将是有限的, 因为我们将它与所有过去的损失一起使用. 所有这些直观都可以用下面的定理来表示.

定理 7.28. *根据算法7.7, 设 $V \subseteq \mathbb{R}^d$ 是凸的, 闭的, 以及非空. 由 $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$ 表明. 假设 $t = 1, \ldots, T$, F_t 为真, 且 λ_t 是强凸的 w.r.t. $\|\cdot\|$, ℓ_t 和 $\tilde{\ell}_t$ 为真且凸, 且 $\operatorname{int}\operatorname{dom} F_t \cap V \neq \{\}$. 同理, 假设 $\partial \ell_t(\boldsymbol{x}_t)$ 和 $\partial \tilde{\ell}_t(\boldsymbol{x}_t)$ 非空. 则, 对于 $t = 1, \ldots, T$, 存在 $\tilde{\boldsymbol{g}}_t \in \partial \tilde{\ell}_t(\boldsymbol{x}_t)$ 有*

$$
\begin{aligned}
&\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \\
&\quad \leq \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) + \sum_{t=1}^{T} \left[\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \right] \\
&\quad \leq \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) + \sum_{t=1}^{T} \left[\frac{\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star^2}{2\lambda_t} + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \right],
\end{aligned}
$$

对于所有的 $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$.

证明. 我们可以将 Optimistic-FTRL 解释为带有正则化器 $\tilde{\psi}_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \tilde{\ell}_t(\boldsymbol{x})$ 的 FTRL. 同样, 注意 $\tilde{\ell}_{T+1}(\boldsymbol{x})$ 对算法没有影响, 所以我们可以设它为空函数.

因此, 从 FTRL 的后悔相等, 我们立即得到

$$
\begin{aligned}
&\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \\
&\quad \leq \tilde{\ell}_{T+1}(\boldsymbol{u}) + \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \left(\tilde{\ell}_1(\boldsymbol{x}) + \psi_1(\boldsymbol{x}) \right) + \sum_{t=1}^{T} [F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) + \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_{t+1}(\boldsymbol{x}_{t+1})] \\
&\quad = \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) + \sum_{t=1}^{T} [F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t)],
\end{aligned}
$$

其中 $\tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_{t+1}(\boldsymbol{x}_{t+1})$ 项来自于一个很小的和. 现在关注 $F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t)$ 项. 得到 $F_t(\boldsymbol{x}) + \ell_t(\boldsymbol{x}) + i_V(\boldsymbol{x})$ 是 λ_t 强凸的 w.r.t. $\|\cdot\|$, 因此, 有

$$
\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t)) - (F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_{t+1})) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\leq \langle \boldsymbol{g}_t', \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}),
\end{aligned}
$$

其中 $\boldsymbol{g}_t' \in \partial(F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t) + i_V(\boldsymbol{x}_t))$. 观察 $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x} \in V} \; F_t(\boldsymbol{x}) + \tilde{\ell}_t(\boldsymbol{x})$, 有 $\boldsymbol{0} \in \partial(F_t(\boldsymbol{x}_t) + \tilde{\ell}_t(\boldsymbol{x}_t) + i_V(\boldsymbol{x}_t))$. 因此, 假定我们的假设保证了这个和的次微分等于这些次微分的和, 存在 $\tilde{\boldsymbol{g}}_t \in \partial \tilde{\ell}_t(\boldsymbol{x}_t)$ 使得 $\boldsymbol{g}_t' = \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t$. 故, 我们有

$$
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) \leq \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \ .
$$

根据双重规范的定义, 我们也有

$$
\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \leq \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\| - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \leq \frac{\lambda_t}{2} \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star^2 \ . \ \square
$$

我们来看看这个定理的第二界限. 与有着类似的边界的 FTRL 相比, 我们现在有 $\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star^2$ 项替代 $\|\boldsymbol{g}_t\|_\star^2$. 所以, 如果对下一次损失的预测是正确的, 那么这个期限可能会变得更短, 甚至可能为零! 另一方面, 如果预测不好, 对于利普希茨的损失我们只损失一个常数因子. 总的来说, 在最好的情况下我们会得到很多, 在最坏的情况下我们不会失去那么多.

尽管算法和分析都很简单, 但这一原理有很多应用. 我们只描述其中的几个. 最近, 这一思想甚至被用于恢复 Nesterov 的加速算法, 并证明了在重复博弈中更快的收敛.

7.11.1 后悔取决于次梯度的方差

考虑在线性损失 $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle$ 上运行 Optimistic-FTRL . 如果我们可以预测下一个 $\boldsymbol{g}_t$, 那么我们可以从 Optimistic-FTRL 和 plain FTRL 对比中获得一些东西. 一种简单的可能性是预测过去值的平均值, $\bar{\boldsymbol{g}}_t = \frac{1}{t-1}\sum_{i=1}^{t-1} \boldsymbol{g}_i$. 事实上, 从第一章我们就知道这种策略本身就是一种在线学习程序! 具体来说, 它对应于损失 $\ell_t(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{g}_t\|_2^2$ 上的 Follow-The-Leader 算法. 因此, 从这个损失的强凸性, 我们知道

$$\sum_{t=1}^T \|\bar{\boldsymbol{x}}_t - \boldsymbol{g}_t\|_2^2 - \sum_{t=1}^T \|\boldsymbol{u} - \boldsymbol{g}_t\|_2^2 \le 4(1+\ln T),\ \forall \boldsymbol{u} \in \mathbb{R}^d\ .$$

表明

$$\sum_{t=1}^T \|\bar{\boldsymbol{x}}_t - \boldsymbol{g}_t\|_2^2 \le 4(1+\ln T) + \min_{\boldsymbol{u}} \sum_{t=1}^T \|\boldsymbol{u} - \boldsymbol{g}_t\|_2^2\ .$$

立即得出最小值是 $\boldsymbol{u} = \frac{1}{T}\sum_{t=1}^T \boldsymbol{g}_t$, 结果是 T 乘以次梯度的经验方差. 将其带入 Optimistic-FTRL 后悔中, 满足 $\psi_t = \psi$, 有

$$\sum_{t=1}^T \ell(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \psi(\boldsymbol{u}) - \psi(\boldsymbol{x}_1) + 4 + 4\ln T + \min_{\boldsymbol{g}} \sum_{t=1}^T \frac{\|\boldsymbol{g}_t - \boldsymbol{g}\|_*^2}{2\lambda}\ .$$

附注 7.29. *我们不需要使用过去次梯度的均值, 我们可以使用其他策略甚至是不同策略的组合. 例如, 假设子梯度有界, 我们可以使用一种算法来解决带专家学习的问题, 其中每个专家是一种策略. 然后, 我们会得到一个边界, 这个边界取决于对最佳策略的预测, 加上专家算法的后悔.*

7.11.2 渐进变化的在线凸优化

在本节中, 我们考虑这样一种情况, 即我们接收到的损失随着时间有微小的变化. 我们将证明, 在这种情况下, 在损失相等的情况下, 人们可能会不断地后悔.

在这种情况下, 我们可以使用简单策略来预测下一个子梯度是使用之前的子梯度, 即对于 $t \ge 2$ 和 $\tilde{\ell}_1(\boldsymbol{x}) = 0$, 有 $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_{t-1}, \boldsymbol{x}\rangle$

推论 7.30. *在定理7.28得假设下, 对于 $t \ge 2$ and $\tilde{\ell}_1(\boldsymbol{x}) = 0$, 定义 $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_{t-1}, \boldsymbol{x}\rangle$, 设 $\psi_t(\boldsymbol{x}) = \lambda_t \psi(\boldsymbol{x})$ 其中 ψ 是 1-strongly convex w.r.t. $\|\cdot\|$ 且对于 $t = 2, \ldots, T$, λ_t 满足 $\lambda_t \lambda_{t-1} \ge 8M^2$, 其中 M 是损失 ℓ_t 的平滑常数. 那么, $\forall \boldsymbol{u} \in V$, 有*

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \le \lambda_t \psi(\boldsymbol{u}) - \lambda_1 \psi(\boldsymbol{x}_1) + \frac{1}{\lambda_1}\|\nabla \ell_1(\boldsymbol{x}_1)\|_\star^2 + \sum_{t=2}^T \frac{2}{\lambda_t}\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2\ .$$

假设对于所有的 $\boldsymbol{x} \in V$, 有 $\|\nabla \ell_t(\boldsymbol{x})\|_\star \le L$, 设 $\lambda_t = \sqrt{\max(8M^2, 4L^2) + \sum_{i=2}^{t-1} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}$, 有

$$\mathrm{Regret}_T(\boldsymbol{u}) \le \left(\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x}) + 4\right) \sqrt{\max(8M^2, 4L^2) + \sum_{t=2}^{T} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2} + \frac{1}{4} \ .$$

证明. 由一个固定正则化器的 Optimistic-FTRL 边界, 立即得到

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \le \lambda_T \psi(\boldsymbol{u}) - \lambda_1 \psi(\boldsymbol{x}_1) + \sum_{t=1}^{T} \left[\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \right] \ .$$

现在, 考虑到损失 ℓ_t 是 M-smooth. 故, 对于任意的 $\alpha_t > 0$, 有

$$\begin{aligned}
\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 &= \langle \nabla \ell_t(\boldsymbol{x}_t) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1}), \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\le \frac{\alpha_t}{2} \|\nabla \ell_t(\boldsymbol{x}_t) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 + \frac{1}{2\alpha_t} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \ .
\end{aligned}$$

关注第一项, 对于 $t = 2, \ldots, T$, 有

$$\begin{aligned}
\frac{\alpha_t}{2} \|\nabla \ell_t(\boldsymbol{x}_t) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 &= \frac{\alpha_t}{2} \|\nabla \ell_t(\boldsymbol{x}_t) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1}) - \nabla \ell_t(\boldsymbol{x}_{t-1}) + \nabla \ell_t(\boldsymbol{x}_{t-1})\|_\star^2 \\
&\le \alpha_t \|\nabla \ell_t(\boldsymbol{x}_t) - \nabla \ell_t(\boldsymbol{x}_{t-1})\|_\star^2 + \alpha_t \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \\
&\le \alpha_t M^2 \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 + \alpha_t \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \ .
\end{aligned}$$

选择 $\alpha_t = \frac{2}{\lambda_t}$. 有 $t = 2, \ldots, T$

$$\begin{aligned}
&\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\quad \le \frac{2M^2}{\lambda_t} \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 + \frac{2}{\lambda_t} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 - \frac{\lambda_t}{4} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \ .
\end{aligned}$$

对于 $t = 1$, 有

$$\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \le \frac{1}{\lambda_t} \|\nabla \ell_t(\boldsymbol{x}_t) - \tilde{\boldsymbol{g}}_t\|_\star^2 - \frac{\lambda_t}{4} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \ .$$

通过观察假设 λ_t , 对于 $t = 2, \ldots, T$, 表明 $\frac{2M^2}{\lambda_t} \le \frac{\lambda_{t-1}}{4}$. 故, 对于 $t = 1, \ldots, T$, 有

$$\sum_{t=1}^{T} \left(\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \right) \le \frac{1}{\lambda_1} \|\nabla \ell_1(\boldsymbol{x}_1)\|_\star^2 + \sum_{t=2}^{T} \frac{2}{\lambda_t} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \ .$$

把它们放在一起, 我们有了第一个规定的边界.

第二个是通过观察得到的

$$
\begin{aligned}
\sum_{t=2}^{T} \frac{\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}{\lambda_t} &= \sum_{t=2}^{T} \frac{\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}{\sqrt{\max(8M^2, 4L^2) + \sum_{i=2}^{t-1} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}} \\
&\leq \sum_{t=2}^{T} \frac{\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}{\sqrt{\sum_{i=2}^{t} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}} \\
&\leq 2\sqrt{\sum_{i=2}^{T} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2} \,. \qquad \square
\end{aligned}
$$

注意, 如果所有的损失都是相同的, 那么后悔就会持续不断! 这并不奇怪, 因为对下一个损失的预测是对前一个损失的线性近似. 事实上, 回顾这个证明, 关键的思想是利用平滑性来论证, 即使过去的子梯度与当前的子梯度在不同的点, 它仍然是当前子梯度的一个很好的预测.

附注 7.31. *注意, 平滑的假设是必要的. 实际上, 始终传递相同的函数并使用 online-to-batch 转换, 这将导致 Lipschitz 函数的收敛速度为 $O(1/T)$, 这是不可能的.*

7.12 历史片段

在 Abernethy et al. [2]中引入了 Regularized Leader, 在每一步中, 预测计算为正则化项的最小值加上过去所有回合的损失之和. 然而, 早在 [75, 77] 之前,FTRL 的关键思想, 特别是通过 dual 的分析, 被 Shai Shalev-Shwartz 和 Yoram Singer 提出. 后来, Shai Shalev-Shwartz [74] 的博士论文包含了对 FTRL 的最精确的双重分析, 但是他将它称为在线镜像下降, 因为 FTRL 这个名字是后来发明的. 后来, 我致力于时变正则化器与线性损失“广义在线镜像下降” [68] 的 FTRL 的混淆命名的一般性分析, 所以, 现在我正在尝试设置录制:

在此之后, 优化社区也在使用线性损失开始使用 FTRL , 称之为双平均 [61]. 请注意, 这篇文章是 Nesterov 在 2005 年写的,[1], 文章中, 他指出这些想法来自 2001-2002 年, 但他决定一段时间不发布他们, 因为他相信“黑匣子方法在凸优化中的重要性将是不可逆转的会消失, 最后, 这种方法将完全被基于巧妙使用问题结构（例如: 内部点方法, 平滑等）替代, 因此, Nesterov 是第一个发明线性损失的离线 FTRL .

相同的想法在 Xiao [87]中被转化为在线学习和随机优化, 基本上重新发现了 Shalev-Shwartz 的框架并将其重新命名进行了正规化的双重验证 (RDA). 最后,McMahan [56]给出了我这里呈现的优雅平等结果（具有次要改进）, 该结果使用于一般损失函数和常规程序. 设置 $\boldsymbol{u} = \boldsymbol{x}_{T+1}$ 且专用于线性损失, 在 Abernethy et al. [1]等人中得到了类似的平等证明, 基本上与 Orabona et al. [68]等人的不平等相匹配. 请注意, 双平均源于线性损失的 FTRL 的双解释器, 但是引理7.1强调 FTRL 实际上更通用的事实. 人们常常使用 be-the-leader-follow-the-leader（见引理1.1 和练习**??**）来证明悔值边界. 然而, 在固定的常规方案 [56] 例子中的 2 个因素下, 证明失败了. 此外, 它似乎在通用强凸增加正则化的情况下失败, 而它适用于近端正则化 McMahan [56]的特定情况.

[1] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=912637

另一个困惑的来源来自于这样一个事实: 一些人区分了“懒惰”和“贪婪”版本的 OMD. 在实际中, 正如 McMahan [56]所证明的那样, 懒惰算法只是带有线性损失的 FTRL, 贪婪算法只是 OMD. Zinkevich [91]引入了懒惰在线镜像下降的概念, 首次引入了 FTRL 在 $\psi_t(\boldsymbol{x}) = \frac{\eta}{2}\|\boldsymbol{x}\|_2^2$ 特殊情况下的线性损失.

引理7.11的第二个结果是对 Zimmert and Seldin [89]中使用的推理的升华.

(7.4)中使用的 FTRL 和正则化器是在 Orabona and Pál [64]中提出来的, 我提供了一个不需要 Fenchel 共轭的证明的简答版本. van Erven et al. [83] 中介绍了 AdaHedge 算法且在 de Rooij et al. [31]中被重新定义. 报告中的分析来自于 Orabona and Pál [64], 即将 AdaHedge 推广至 AdaFTRL 中的任意正规化. de Rooij et al. [31]证明了随机情况下 AdaHedge 的附加特性.

Xiao [87]首先分析了具有复合损失的 FTRL,. 本文用 $\psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1})$ 负项来简单地证明复合损失的 FTRL 的后悔界是 Orabona et al. [68]提出来的.

对于强凸损失的 FTRL 的第一个证明是在 Shalev-Shwartz and Singer [76] (即使他们不称它为 FTRL). 在同一篇文章中, 他们还定义了关于 Bregman 散度的强凸性, 不仅仅是规范, 并证明了相同的对数后悔界. 类似地, 人们后来定义了关于 Bregman 散度 [16] 的平滑概念, 这个概念在 Bauschke et al. [12]等人的著作中重新发现, 后来这两个概念都被重新命名为“相对平滑”和“相对强凸”[[53].

关于 FTRL-Proximal [55] 有一点很有趣: FTRL-Proximal 是使用特定的近端正则化器的 FTRL 的实例化. 谷歌在一篇非常有影响力的论文中披露他们正在使用 FTRL-Proximal 训练用于点击预测 [59] 的分类器时, 它在互联网公司中变得非常有名. 这产生了更多的混乱, 因为许多人开始将术语 FTRL-Proximal (一个特定的算法) 和 FTRL (一个完整的算法家族) 混为一谈. 不幸的是, 这种混乱在这些日子里仍然存在.

Hazan et al. [39]中引入了在线牛顿步进算法, 并描述了损失函数为 exp-concave 的特殊情况. Kivinen and Warmuth [44]中研究了 Exp-concave 函数. 这里, 我描述了满足(7.8), 的任何函数序列的简单概括, 在我看来, 它更好地展示了 FTRL 与强凸函数和 ONS 之间的并行性. 请注意, 请注意,Hazan et al. [39] 等人还描述了基于 OMD 的 ONS 的一个变体, 但我发现, 从说教的角度来看, 它的分析不那么有趣. 这里通过近端正则化器的特性提出的证明可能是新的, 我不确定.

Vovk-Azoury-Warmuth forecaster 预测器是由 Azoury and Warmuth [10] 和 Vovk [84]独立提出来的. 这里提出的证明来自 Orabona et al. [68].

乐观在线镜像下降算法由 Chiang et al. [25]提出, 并在 Rakhlin and Sridharan [69] 中使用任意 “hallucinated” losses 进行了扩展. Rakhlin and Sridharan [69]提出了乐观的 FTRL 版本, 并在 Steinhardt and Liang [80]中重新发现, 尽管它被称为在线镜像下降, 因为我们已经解释过的错误命名问题. 我在这里提出的定理7.28 的证明是新的.

推论7.30 由 Chiang et al. [25] 对乐观的 OMD 进行了证明, 且在 Joulani et al. [42]中提出了对乐观的 FTRL 类似的形式, 但这些都是针对有界域的.

Chapter 8

在线线性分类

在这一章中, 我们将考虑在线线性分类的问题. 我们考虑以下设置:

- 在每个时间步, 我们接收到一个样本 $\boldsymbol{z}_t$
- 我们输出 $\boldsymbol{z}_t$ 的二进制标签 $y_t \in \{-1, 1\}$
- 我们接收到真实的标签, 我们看看我们是否犯了错误
- 我们更新我们的在线分类器

在线算法的目的是尽量减少与某些最佳固定分类器相比的错误数量. 我们将关注线性分类器, 它用向量 $\boldsymbol{x}_t$ 和输入特征 $\boldsymbol{z}_t$ 之间的内积的符号来预测. 因此, $\tilde{y}_t = \operatorname{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle)$. 这个问题可以再次写成后悔最小化问题:

$$\operatorname{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}),$$

其中 $\ell_t(\boldsymbol{x}) = \mathbf{1}[\operatorname{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle) \neq y_t]$. It should be clear that these losses are non-convex. 很明显, 这些损失是非凸性的. 因此, 我们需要另一种方法来处理它们. 下面, 我们将看到解决这个问题的两种可能的方法.

8.1 在线随机分类器

正如我们在专家建议框架中所做的那样, 我们可能会考虑使用随机化来消除损失. 因此, 在每一轮中, 我们可以预测 $q_t \in [-1, 1]$ 中的一个数字, 并根据概率 $\frac{1+q_t}{2}$ 输出标签 $+1$, 根据概率 $\frac{1-q_t}{2}$ 输出 -1 . 故, 定义随机变量.

$$\tilde{y}(p) = \begin{cases} +1, & \text{with probability } p, \\ -1, & \text{with probability } 1-p \ . \end{cases}$$

现在观察到 $\mathbb{E}_{\tilde{y}_t}[\tilde{y}(\frac{1+q_t}{2}) \neq y_t] = \frac{1}{2}|q_t - y_t|$. 如果我们考虑线性预测因子, 我们可以认为 $q_t = \langle \boldsymbol{z}_t, \boldsymbol{x}_t$, 且对于竞争对手 $q'_t = \langle \boldsymbol{z}_t, \boldsymbol{u} \rangle$ 也类似. 对于 $t = 1, \ldots, T$, 约束算法和竞争对手的向量空间为 $|\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle| \leq 1$,

我们可以写到

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left[\tilde{y}(\langle \boldsymbol{z}_t,\boldsymbol{x}_t\rangle)\neq y_t\right]-\sum_{t=1}^{T}\mathbf{1}\left[\tilde{y}(\langle \boldsymbol{z}_t,\boldsymbol{u}\rangle)\neq y_t\right]\right]=\mathbb{E}\left[\sum_{t=1}^{T}\frac{1}{2}\left|\langle \boldsymbol{z}_t,\boldsymbol{x}_t\rangle-y_t\right|-\sum_{t=1}^{T}\frac{1}{2}\left|\langle \boldsymbol{z}_t,\boldsymbol{u}\rangle-y_t\right|\right] .$$

因此, 代理凸损失成为 $\tilde{\ell}_t(\boldsymbol{x})=\frac{1}{2}|\langle \boldsymbol{z}_t,\boldsymbol{x}\rangle-y_t|$, 可行集是任意凸集, 对于 $t=1,\ldots,T$, 我们具有属性 $|\langle \boldsymbol{z}_t,\boldsymbol{x}\rangle|\leq 1$.

考虑到这个问题是凸的, 假设 $\boldsymbol{z}_t$ to be bounded w.r.t. some norm, 我们可以使用到目前为止我们看到的几乎任何算法, 从在线镜像下降到 Follow-The-Regularized-Leader. 所有这些都有 $O(\sqrt{T})$ 的后悔上界, 假设 $\boldsymbol{z}_t$ 在某个范数上有界. 唯一的警告是在 $[-1,1]$ 中约束 $\langle \boldsymbol{z}_t,\boldsymbol{x}_t\rangle$. 一种可以考虑的假设是 $\|\boldsymbol{z}_t\|_\star\leq R$, 并选择可行集 $V=\{\boldsymbol{x}\in\mathbb{R}^d:\|\boldsymbol{x}\|\leq\frac{1}{R}\}$.

综上所述, 例如, 我们可以使用带有正则化器的 FTRL 来实现以下策略 $\psi_t(\boldsymbol{x})=\frac{\sqrt{2t}}{4}\|\boldsymbol{x}\|_2^2$.

Algorithm 8.1 Randomized Online Linear Classifier through FTRL

Require: $R>0$ such that $\|\boldsymbol{z}_t\|_2\leq R$ for $t=1,\ldots,T$

Set $\boldsymbol{\theta}_1=\mathbf{0}\in\mathbb{R}^d$
for $t=1$ **to** T **do**
  $\boldsymbol{x}_t=\eta_t\theta_t\min(\frac{1}{R\eta_t\|\theta_t\|_2},1)$
  Receive $\boldsymbol{z}_t\in\mathbb{R}^d$
  Predict $\tilde{y}_t=\begin{cases}1, & \text{with probability } \frac{\langle \boldsymbol{z}_t,\boldsymbol{x}_t\rangle+1}{2}\\ -1, & \text{otherwise}\end{cases}$
  Receive y_t and pay $\mathbf{1}[y_t\neq\tilde{y}_t]$
  $\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\frac{1}{2}\mathrm{sign}(\langle \boldsymbol{z}_t,\boldsymbol{x}_t\rangle-y_t)\boldsymbol{z}_t$ where $\mathrm{sign}(0):=0$
end for

定理 8.1. 设 $(\boldsymbol{z}_t,y_t)_{t=1}^T$ 是一个样本/标签对的任意序列, 其中 $(\boldsymbol{z}_t,y_t)\in X\times\{-1,1\}$ 和 $X\subset\mathbb{R}^d$. 假设对于 $t=1,\ldots,T$, 有 $\|\boldsymbol{z}_t\|_2\leq R$. 因此, 运行随机在线线性分类器算法满足 $\eta_t=\frac{\sqrt{2}}{\sqrt{t}}$, 其中 $\alpha>0$, 对于任意的 $\boldsymbol{u}\in\{\boldsymbol{x}\in\mathbb{R}^d:\|\boldsymbol{x}\|_2\leq\frac{1}{R}\}$ 有以下保证

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}[\tilde{y}(\langle \boldsymbol{z}_t,\boldsymbol{x}_t\rangle)\neq y_t]\right]-\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}[\tilde{y}(\langle \boldsymbol{z}_t,\boldsymbol{u}\rangle)\neq y_t]\right]\leq\sqrt{2T} .$$

证明. 从选择的递增正规化器的 FTRL 后悔约束得到的证明是直接的. □

8.2 感知器算法

上述策略的缺点是将可行向量限制在一个可能非常小的集合中, 这反过来会使竞争对手的性能较低. 反过来, 在线算法的性能只接近于竞争对手的一个. 另一种处理非凸性的方法是比较算法的错误数量与竞争对手的凸累积损失. 也就是说, 我们可以试着证明一个较弱的后悔保证:

$$\sum_{t=1}^{T}\mathbf{1}[y_t\neq\tilde{y}_t]-\sum_{t=1}^{T}\ell(\langle \boldsymbol{z}_t,\boldsymbol{u}\rangle,y_t)=O\left(\sqrt{T}\right) . \tag{8.1}$$

具体来说, 我们考虑的凸损失是 Hinge 损失的方幂: $\ell_q(\tilde{y}, y) = \max(1 - \tilde{y}y, 0)^q$. hinge 损失是 0/1 损失的凸上界, 当预测符号正确且内积的大小足够大时,hinge 损失达到 0. 此外, 对其取幂, 我们得到一系列函数, 这些函数在错误分类样本的损失与正确分类样本的损失之间进行权衡, 但是满足 $|\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle| \leq 1$, 见图8.1.

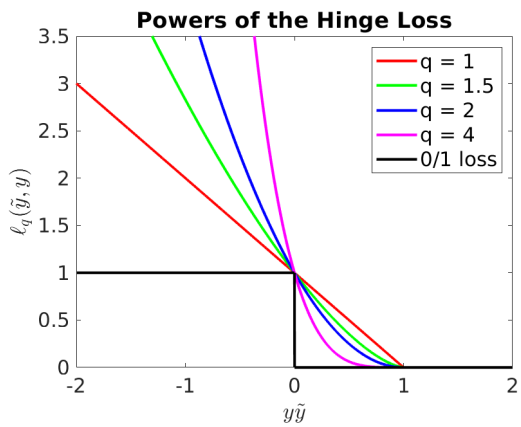

图 8.1: Hinge 损失的功率.

我们需要最小化 (8.1)中的修改过的最古老的算法是算法 8.2中的感知器算法.

Algorithm 8.2 Perceptron Algorithm

1: Set $\boldsymbol{x}_1 = \mathbf{0} \in \mathbb{R}^d$
2: **for** $t = 1$ **to** T **do**
3: Receive $\boldsymbol{z}_t \in \mathbb{R}^d$
4: Predict $\tilde{y}_t = \text{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle)$
5: Receive y_t and pay $\mathbf{1}[y_t \neq \tilde{y}_t]$
6: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t$
7: **end for**

感知机算法更新当前的预测 $\boldsymbol{x}_t$ 沿着当前样本乘以标签的方向移动. 我们来看看为什么这是个好主意. 假设 $y_t = 1$ 算法出现错误. 然后, 更新后的预测 $\boldsymbol{x}_{t+1}$ 将在同一样品 $\boldsymbol{z}_t$ 上预测一个更正数. 事实上, 有

$$\langle \boldsymbol{z}_t, \boldsymbol{x}_{t+1}\rangle = \langle \boldsymbol{z}_t, \boldsymbol{x}_t + y_t \boldsymbol{z}_t\rangle = \langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle + \|\boldsymbol{z}_t\|_2^2 \ .$$

同样, 当 $y_t = -1$ 且算法出现错误时, 更新会导致对同一样本的预测更为负面. 对于感知器算法, 我们可以证明以下保证.

定理 8.2. *设 $(\boldsymbol{z}_t, y_t)_{t=1}^T$ 是一个样本/标签对的任意序列, 其中 $(\boldsymbol{z}_t, y_t) \in X \times \{-1, 1\}$ 且 $X \subset \mathbb{R}^d$. 对于 $t = 1, \ldots, T$, 设 $\|\boldsymbol{z}_t\|_2 \leq R$. 然后, 运行感知器算法我们有以下保证*

$$\sum_{t=1}^{T}\mathbf{1}[y_t \neq \tilde{y}_t] - \sum_{t=1}^{T}\ell(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle, y_t) \leq \frac{q^2R^2\|\boldsymbol{u}\|_2^2}{2} + q\|\boldsymbol{u}\|\sqrt{\frac{q^2R^2\|\boldsymbol{u}\|_2^2}{4} + \sum_{t=1}^{T}\ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle, y_t)},\ \forall \boldsymbol{u} \in \mathbb{R}^d, q \geq 1\ .$$

在证明这个定理之前, 让我们先看看它的意义. 如果存在一个 $\boldsymbol{u} \in \mathbb{R}^d$ 使得 $\sum_{t=1}^{T}\ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle, y_t) = 0$, 则感知器算法产生的错误数量有限, 上限为 $R^2\|\boldsymbol{u}\|_2^2$. 如果有很多 $\boldsymbol{u}$ 达到 $\sum_{t=1}^{T}\ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle, y_t) = 0$. 我们有有限的错误数目是其中最小 $\boldsymbol{u}$ 的范数的界. 这个量是什么意思?

记住, 由其法向量 $\boldsymbol{u}$ 表示的超平面将空间分割成两个半空间: 一个点 $\boldsymbol{z}$ 为内积 $\langle \boldsymbol{z}, \boldsymbol{u}\rangle$ 提供正值, 另一个内积为负数. 现在, 我们知道样本 $\boldsymbol{z}_t$ 到法向为 $\boldsymbol{u}$ 的超平面的距离是

$$\frac{|\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle|}{\|\boldsymbol{u}\|_2}\ .$$

同时, 考虑到累积铰链损失为零的 $\boldsymbol{u}$ 值, 我们知道这个量至少为 $\frac{1}{\|\boldsymbol{u}\|_2}$. 因此, 累积 hinge 损失等于零的最小 $\boldsymbol{u}$ 的范数与点与分离超平面之间的最小距离成反比. 这个距离称为样本 $(\boldsymbol{z}_t, y_t)_{t=1}^{T}$ 的边距. 所以, 如果边界很小, 感知器算法会比边界很大时出错更多.

如果问题不能线性可分, 感知器满足 $O(\sqrt{L^\star})$ 的后悔, 其中 $L^\star$ 是竞争对手的损失. 此外, 我们用一组损失函数来衡量竞争对手, 并与用最佳损失衡量的最佳 $\boldsymbol{u}$ 进行竞争. 这种适应性是通过两个基本成分实现的:

- 感知器通过假设的学习速率 η 来独立于更新的缩放, 也就是说感知器所犯的错误与缩放无关. 也就是说, 我们可以使用 $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\mathbf{1}[y_t \neq \tilde{y}_t]y_t\boldsymbol{z}_t$ 进行更新, 并且有相同的错误和更新, 因为它们只依赖于 $\tilde{y}_t$. 因此, 我们可以认为它总是使用最好的学习速率 η. “后悔”的弱化定义允许考虑一系列的损失函数, 因为感知器在更新中没有使用它们中的任何一个.

现在让我们来证明后悔的保证. 为了证明, 我们将需要以下两个技术引理.

引理 8.3. *[26, Lemma 10.17] Let $p, q > 1$ be such that $\frac{1}{p} + \frac{1}{q} = 1$. Then*

$$ab \leq \frac{1}{q}a^q + \frac{1}{p}b^p,\ \forall a, b\ .$$

引理 8.4. 设 $a, c > 0$, $b \geq 0$, 且 $x \geq 0$ 使得 $x - a\sqrt{x} \leq c$. 则, $x \leq c + \frac{a^2}{2} + a\sqrt{\frac{a^2}{4} + c}$.

证明. 设 $y = \sqrt{x}$, 有 $y^2 - ay - c \leq 0$. 求解 y, 有 $y \leq \frac{a+\sqrt{a^2+4c}}{2}$. 因此, $x = y^2 \leq \frac{a^2+2a\sqrt{a^2+4c}+a^2+4c}{4} = \frac{a^2}{2} + \frac{a}{2}\sqrt{a^2+4c} + c$. □

引理??. 用感知器算法的总错误数表示为 $M = \sum_{t=1}^{T}\mathbf{1}[y_t \neq \tilde{y}_t]$.

首先, 请注意, 感知器算法可以被认为是在线运行的子梯度下降 (OSD), 对于 $V = \mathbb{R}^d$ 上的 $\tilde{\ell}_t(\boldsymbol{x}) = -\langle \mathbf{1}[y_t \neq \tilde{y}_t]y_t\boldsymbol{z}_t, \boldsymbol{x}\rangle$, 其步长固定为 η . 事实上,OSD 针对这些损失将更新为

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\mathbf{1}[y_t \neq \tilde{y}_t]y_t\boldsymbol{z}_t\ . \tag{8.2}$$

现在, 如上文所述, η 不会以任何方式影响预测的符号, 因此感知器算法可以运行 (8.2) 它的预测将完全相同. 因此, 我们有

$$\sum_{t=1}^{T} -\langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{x}_t \rangle + \sum_{t=1}^{T} \langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle \leq \frac{\|\boldsymbol{u}\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathbf{1}[y_t \neq \tilde{y}_t] \|\boldsymbol{z}_t\|_2^2, \ \forall \eta > 0 \ .$$

既然这个不等式对任意 η 值都成立, 我们就可以选择使 r.h.s. 最小的值, 有

$$\sum_{t=1}^{T} -\langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{x}_t \rangle + \sum_{t=1}^{T} \langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle \leq \|\boldsymbol{u}\| \sqrt{\sum_{t=1}^{T} \mathbf{1}[y_t \neq \tilde{y}_t] \|\boldsymbol{z}_t\|_2^2} \leq \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \mathbf{1}[y_t \neq \tilde{y}_t]} \ . \tag{8.3}$$

注意 $-\mathbf{1}[y_t \neq \tilde{y}_t] y_t \langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle \geq 0$. 同样, 有

$$\langle y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle = 1 - (1 - \langle y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle) \geq 1 - \max(1 - \langle y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle, 0) = 1 - \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) \ .$$

故, 用 $\tau_t = \mathbf{1}[y_t \neq \tilde{y}_t]$ 表示, 可以将 (8.3) 写成

$$\begin{aligned}
\sum_{t=1}^{T} \tau_t &\leq \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \tau_t} + \sum_{t=1}^{T} \tau_t \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) \\
&\leq \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \tau_t} + \left(\sum_{t=1}^{T} \tau_t^p \right)^{1/p} \left(\sum_{t=1}^{T} \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)^q \right)^{1/q} \\
&= \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \tau_t} + \left(\sum_{t=1}^{T} \tau_t \right)^{1/p} \left(\sum_{t=1}^{T} \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)^q \right)^{1/q} ,
\end{aligned}$$

我们使用 Holder 不等式, 且 $\frac{1}{p} + \frac{1}{q} = 1$.

给出 $M = \sum_{t=1}^{T} \tau_t$, 且用 $L_q = \sum_{t=1}^{T} \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)$ 表示, 有

$$M \leq \|\boldsymbol{u}\| R \sqrt{M} + M^{1/p} L_q^{1/q} \ .$$

考虑两个例子, 对于 $q = 1$, 我们可以使用引理8.4 , 得到给定的边界. 实际上, 对于 $q > 1$, 使用引理8.3 有

$$M \leq \|\boldsymbol{u}\| R \sqrt{M} + M^{1/p} L_q^{1/q} \leq \|\boldsymbol{u}\| R \sqrt{M} + \frac{1}{p} M + \frac{1}{q} L_q,$$

表明

$$M \left(1 - \frac{1}{p} \right) \leq \|\boldsymbol{u}\| R \sqrt{M} + \frac{1}{q} L_q \ .$$

使用事实 $1 - \frac{1}{p} = \frac{1}{q}$, 有

$$M \leq q \|\boldsymbol{u}\| R \sqrt{M} + L_q \ .$$

最后, 使用引理8.4, 我们有给定的边界. □

8.3 历史片段

感知器是由 Rosenblatt Rosenblatt [73]提出的. 对于 $q = 1$ 在不可分情况下的收敛性证明是由 Gentile [35] 证明的, 对于 $q = 2$, 则是由 Freund and Schapire [34]证明的. 这里给出的证明是基于 Beygelzimer et al. [15]等人的证明.

Chapter 9

参数无关在线线性优化

在前面的章节中, 我们已经展示了在线镜像下降 (OMD) 和 Follow-The-Regularized-Leader (FTRL) 实现凸 Lipschitz 损失的后悔 $O(\sqrt{T})$. 我们还证明了对于有界域, 这些边界在常数乘因子下是最优的. 然而, 在无界的情况下, 我们得到的边界是次优的 w.r.t. 依赖于竞争对手. 更特别的是, 让我们考虑一个在线次梯度下降的例子,$V = \mathbb{R}^d$ over 1-Lipschitz 损失和学习速率 $\eta = \frac{\alpha}{\sqrt{T}}$. 我们得到以下后悔保证

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{\|\boldsymbol{u}\|_2^2}{2\eta} + \frac{\eta T}{2} = \frac{1}{2}\sqrt{T}\left(\frac{\|\boldsymbol{u}\|_2^2}{\alpha} + \alpha\right) .$$

因此, 要想得到最好的保证, 就必须知道 $\|\boldsymbol{u}\|_2$ 并设 $\alpha = \|\boldsymbol{u}\|_2$. 就像我们说的, 这个策略是行不通的, 一是我们不知道 $\|\boldsymbol{u}\|_2$, 二是如果我们猜中了 $\|\boldsymbol{u}\|_2$ 的价值, 对手很容易改变损失, 使这个价值完全错误. 这不是一个技术性问题, 而是一个重要的问题, 如下面的示例所示.

例 9.1. *考虑到我们想要使用 OSD with online-to-batch 转换来最小化一个 1-Lipschitz 函数. 利用学习效率 $\eta = \frac{\alpha}{\sqrt{T}}$ 使转化效率为 $O((\frac{\|\boldsymbol{u}^\star\|_2^2}{\alpha} + \alpha)\sqrt{T})$. 假设 $\|\boldsymbol{u}^\star\|_2 = 100$, 指定 $\alpha = 1$ 的收敛速度比指定最佳选择 $\alpha = 100$ 慢 100 倍. 请注意, 这是一个真实的效果, 而不是人为的证明. 事实上, 最优学习率应该与算法选取的初始点与最优解之间的距离成正比, 这是很直观的.*

如果我们能以最优的方式调整学习速率, 我们将会感到后悔

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \|\boldsymbol{u}\|_2\sqrt{T} .$$

然而, 这也是不可能的, 因为我们对 $\Omega(\|\boldsymbol{u}\|_2\sqrt{T\ln(\|\boldsymbol{u}\|_2 + 1)})$ 的后悔证明了一个下限.

在下面, 我们将证明将任何在线凸优化 (OCO) 游戏减少到投注非随机硬币是可能的. 这将允许我们使用一种完全不同的方式来设计 OCO 算法, 它将享受最优的后悔, 并且不需要调整任何参数 (例如学习速率、正则化权重). 我们称之为无参数算法.

9.1 Coin-Betting 游戏

想象以下重复性试验:

- 设 ϵ 的初始值为: $\text{Wealth}_0 = \epsilon$.
- 每一轮次 $t = 1, \ldots, T$
  - 你下注 $|x_t|$ 钱, 押硬币的一面, 记为 $\text{sign}(x_t)$; 你不可能押比你现有的钱更多的钱.
  - 对手揭示硬币的结果 $c_t \in \{-1, 1\}$.
  - 你获得 $x_t c_t$, 即 $\text{Wealth}_t = \text{Wealth}_{t-1} + c_t x_t = \epsilon + \sum_{i=1}^{t} c_i x_i$.

假设我们不能借钱, 我们可以将 x_t 映射为 $\beta_t \in [-1, 1]$ 上的 $\beta_t \text{Wealth}_{t-1}$. 因此, $|\beta_t|$ 是要下注的钱的分数, $\text{sign}(\beta_t)$ 也是我们下注的硬币那面的标签 $\text{sign}(\beta_t)$.

游戏的目标是尽可能多地赚钱. 像往常一样, 考虑到游戏的对抗性, 我们不能指望总是赢钱. 相反, 我们试图在整个游戏中获得与 $\beta^\star \in [-1, 1]$ 打赌固定数量一样多的钱.

注意

$$\text{Wealth}_t = \text{Wealth}_{t-1} + c_t x_t = \text{Wealth}_{t-1} + \beta_t \text{Wealth}_{t-1} c_t = \text{Wealth}_{t-1}(1 + \beta_t c_t) = \epsilon \prod_{i=1}^{t-1} (1 + \beta_i c_i) \ .$$

所以, 考虑到财富的乘数性质, 对财富的比率取对数的算法和财富的最优投注分数也是有用的. 因此, 我们要尽量减少以下后悔

$$\begin{aligned}
\ln \max_{\beta \in [-1,1]} \epsilon \prod_{t=1}^{T} (1 + \beta c_t) - \ln \text{Wealth}_T &= \ln \max_{\beta \in [-1,1]} \epsilon \prod_{t=1}^{T} (1 + \beta c_t) - \ln \left(\epsilon \prod_{t=1}^{T} (1 + \beta_t c_t) \right) \\
&= \max_{\beta \in [-1,1]} \sum_{t=1}^{T} \ln(1 + \beta c_t) - \sum_{t=1}^{T} \ln(1 + \beta_t c_t) \ .
\end{aligned}$$

换句话说, 这只是 OCO 游戏的后悔, 损失是 $\ell_t(x) = -\ln(1 + x c_t)$ 和 $V = [-1, 1]$. 我们也可以扩展一下允许“连续币”的公式, 其中 $c_t \in [-1, 1]$ 而不是 $\{-1, 1\}$.

附注 9.2. *注意, 打赌在 -1 和 1 之间的分数的约束并不是严格必须的, 我们可以让算法在现有的基础上押更多的钱, 每轮借给它一些钱. 然而, 这种限制使分析更容易, 因为它允许上述转化为 OCO 问题, 使用 $1 + \beta_t c_t$ 的非负性.*

我们可以只使用 OMD 或 FTRL, 特别注意函数的 non-Lipschitzness 性质, 但事实证明, 有一个更好的策略专门针对这个问题. 解决上述投币游戏有一个非常简单的策略, 叫做 **Krichevsky-Trofimov (KT) bettor**. 它简单地说, 在每个时间步 t 上, 你打赌 $\beta_t = \frac{\sum_{i=1}^{t-1} c_i}{t}$. 故, 算法是这样的

为此, 我们能够证明以下定理:

定理 9.3 ([20, Theorem 9.4])**.** *对于 $t = 1, \ldots, T$, 设 $c_t \in \{-1, 1\}$. 算法9.1中的 KT bettor 可以保证*

$$\ln \text{Wealth}_T \geq \ln \max_{\beta \in [-1,1]} \prod_{t=1}^{T} (1 + \beta c_t) - \frac{1}{2} \ln T - K,$$

其中 K 是一个普遍的常数.

Algorithm 9.1 Krichevsky-Trofimov bettor

Require: Initial money $\text{Wealth}_0 = \epsilon > 0$

for $t = 1$ **to** T **do**
  Calculate the betting fraction $\beta_t = \frac{\sum_{i=1}^{t-1} c_i}{t}$
  Bet $x_t = \beta_t \text{Wealth}_{t-1}$, that is $|x_t|$ money on the side $\text{sign}(x_t) = \text{sign}(\beta_t)$
  Receive the coin outcome $c_t \in [-1, 1]$
  Win/loose $x_t c_t$, that is $\text{Wealth}_t = \text{Wealth}_{t-1} + c_t x_t$
end for

请注意, 如果硬币的结果偏向一边, 最优投注分数将获得指数数量的钱, 正如下一个引理所证明的那样.

引理 9.4. *设 $g_t \in \{-1, 1\}, t = 1, \ldots, T$. 有*

$$\max_{\beta \in [-1,1]} \exp\left(\sum_{t=1}^{T} \ln(1 - \beta g_t)\right) \geq \exp\left(\sum_{t=1}^{T} \frac{\left(\sum_{t=1}^{T} g_t\right)^2}{4T}\right) .$$

证明.

$$\begin{aligned}\max_{\beta \in [-1,1]} \exp\left(\sum_{t=1}^{T} \ln(1 - \beta g_t)\right) &\geq \max_{\beta \in [-1/2,1/2]} \exp\left(\sum_{t=1}^{T} \ln(1 - \beta g_t)\right) \geq \max_{\beta \in [-1/2,1/2]} \exp\left(\beta \sum_{t=1}^{T} g_t - \beta^2 \sum_{t=1}^{T} g_t^2\right) \\ &= \max_{\beta \in [-1/2,1/2]} \exp\left(\beta \sum_{t=1}^{T} g_t - \beta^2 T\right) = \exp\left(\frac{\left(\sum_{t=1}^{T} g_t\right)^2}{4T}\right) .\end{aligned}$$

对于 $x \in [-1/2, 1/2]$, 使用基本不等式 $\ln(1 + x) \geq x - x^2$. □

因此,KT 保证了一个指数数量的资金, 只支付 $\sqrt{T}$ 的罚款. 有可能证明 KT 算法的上述保证对常数加性因子是最优的. 此外, 注意 KT 策略不需要设置任何参数: 没有学习率, 也没有正则化器. 也就是说 KT 是无参数的. 同样, 我们可以将 KT 算法的保证扩展到硬币“连续的情况, 即 $c_t \in [-1, 1]$. 有以下定理.

定理 9.5 ([65, Lemma 14])**.** *对于 $t = 1, \ldots, T$, 设 $c_t \in [-1, 1]$. 算法9.1的 KT bettor 保证*

$$\ln \text{Wealth}_T \geq \sum_{t=1}^{T} \frac{\left(\sum_{t=1}^{T} c_t\right)^2}{4T} - \frac{1}{2} \ln T - K,$$

其中 K 是一个普遍函数.

因此, 我们引入投币游戏, 并将其扩展到连续投币, 提出了一种简单的无参数最优策略. 在下一节中, 我们将展示如何使用 KT bettor 作为无参数的 1-d OCO 算法.

9.2 Parameter-free 1d OCO through Coin-Betting

故, 定理9.3 告诉我们, 在每一步中, 通过最优固定比例的投注, 我们可以赢得几乎和策略一样多的钱. 我们只支付对数财富的对数价格, 它对应于实际财富的 $\frac{1}{\sqrt{T}}$.

现在, 让我们看看为什么这个问题在 OCO 中很有趣. 结果表明, 用连续硬币解决投币游戏等价于解决一个一维无约束在线线性优化问题. 也就是说, 投币算法等价于设计一个在线学习算法, 产生一个 $x_t \in \mathbb{R}$ 的序列, 使一维后悔线性损失最小化:

$$\mathrm{Regret}_T(u) := \sum_{t=1}^{T} g_t x_t - \sum_{t=1}^{T} g_t u,$$

其中 g_t 是对抗的和有界限的. 在不丧失一般性的前提下, 我们假设 $g_t \in [-1, 1]$. 此外, 要记住 OCO 游戏可以简化为在线线性优化 (OLO) 游戏, 这种简化将有效地将 OCO 简化为投币游戏! 此外, 通过在线批量转换, 任何随机的一维问题都可以简化为投币游戏!

允许在 OLO 和投币之间进行转换的关键定理如下.

定理 9.6. *设 $\phi : \mathbb{R}^d \to (-\infty, +\infty]$ 是一个为真的闭凸函数, 设 $\phi^\star : \mathbb{R}^d \to (-\infty, +\infty]$ 是它的 Fenchel 共轭. 一个生成 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T \in \mathbb{R}^d$ 的算法保证*

$$\forall \boldsymbol{g}_1, \ldots, \boldsymbol{g}_T \in \mathbb{R}^d, \quad \epsilon - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \geq \phi\left(-\sum_{t=1}^{T} \boldsymbol{g}_t\right)$$

其中 $\epsilon \in \mathbb{R}$, 如果其仅仅保证

$$\forall \boldsymbol{u} \in \mathbb{R}^d, \quad \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle}_{\mathrm{Regret}_T(\boldsymbol{u})} \leq \phi^\star(\boldsymbol{u}) + \epsilon \ .$$

证明. 我们来证明从左到右的含义.

$$\begin{aligned}
\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle &\leq -\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \phi\left(-\sum_{t=1}^{T} \boldsymbol{g}_t\right) + \epsilon \\
&\leq \sup_{\boldsymbol{\theta} \in \mathbb{R}^d} \langle \boldsymbol{\theta}, \boldsymbol{u} \rangle - \phi(\boldsymbol{\theta}) + \epsilon = \phi^\star(\boldsymbol{u}) + \epsilon \ .
\end{aligned}$$

对于另一个含义, 我们有

$$\begin{aligned}
-\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle &= -\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \\
&= \sup_{\boldsymbol{u} \in \mathbb{R}^d} -\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \\
&\geq \sup_{\boldsymbol{u} \in \mathbb{R}^d} -\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \phi^\star(\boldsymbol{u}) - \epsilon = \phi\left(-\sum_{t=1}^{T} \boldsymbol{g}_t\right) - \epsilon \ .
\end{aligned}$$

□

为了使上述定理有意义, 假设我们正在考虑一个一维问题且 $g_t \in [-1, 1]$. 然后, 保证一个下界

$$\epsilon - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$$

可以通过押注 x_t 在硬币 $c_t = -g_t$ 的投注策略来完成. 因此, 该定理意味着证明投币游戏中财富的奖励下限意味着对应的一维 OLO 游戏的后悔上限. 然而, 证明奖励下限更容易, 因为它不依赖于竞争对手 $\boldsymbol{u}$. 事实上, 不知道竞争对手的规范正是优化 OMD 的学习率困难的原因!

这种考虑立即给了我们 1-d OLO 和投币之间的转换: **硬币的结果是当前预测损失的次梯度的负值.** 事实上, 设置 $c_t = -g_t$, 我们有一个投币算法, 投币 x_t 会给我们

$$\text{Wealth}_T = \epsilon + \sum_{t=1}^{T} x_t c_t = \epsilon - \sum_{t=1}^{T} x_t g_t \ .$$

因此, 财富的下限对应于定理9.6中使用的下限. 为了得到一个后悔的保证, 我们只需要计算奖励函数的 Fenchel 共轭, 假设它可以表示为 $\sum_{t=1}^{T} c_t$ 的函数.

最后一步是将 1-d OCO 减少为 1-d OLO. 但是, 这是一个我们已经做过很多次的简单步骤. 事实上, 我们有

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq \sum_{t=1}^{T} x_t g_t - \sum_{t=1}^{T} g_t u,$$

其中 $g_t \in \partial \ell_t(x_t)$.

因此, 总的来说, 投币博弈财富下限的 Fenchel 共轭成为 OCO 博弈的后悔保证. 在下一节中, 我们将所有这些考虑专门化到 KT 算法.

9.2.1 KT 作为一种一维在线凸优化算法

在这里, 我们希望利用上一节中的考虑因素来使用 KT 作为无参数的 1-d OCO 算法. 首先, 让我们看看这种算法是什么样的. KT 投注 $x_t = \beta_t \text{Wealth}_{t-1}$, 从 ϵ 资金开始. 现在, 设 $c_t = -g_t$, 其中 $g_t \in \partial \ell_t(x_t)$ 且设损失 ℓ_t 1-Lipschitz. 因此, 得到

$$x_t = -\frac{\sum_{i=1}^{t-1} g_i}{t} \left(\epsilon - \sum_{i=1}^{t-1} g_i x_i \right) \ .$$

伪代码在算法中9.2.

现在看看我们会后悔到什么程度. 根据定理9.5, 我们可以得到, 当使用 $c_t = -g_t$ 时,KT 下注者保证了以下财富的下界: :

$$\epsilon - \sum_{t=1}^{T} x_t g_t \geq \frac{\epsilon}{K\sqrt{T}} \exp\left(\frac{(\sum_{t=1}^{T} g_t)^2}{4T} \right) \ .$$

故, 我们发现了 ϕ 函数, 我们需要 $\phi^\star$ 或者它的上界, 可以用下面的引理找到.

Algorithm 9.2 Krichevsky-Trofimov OCO

Require: $\epsilon > 0$ (any number between 1 and $\sqrt{T}$)

1: **for** $t = 1$ **to** T **do**
2: Predict $x_t = -\frac{\sum_{i=1}^{t-1} g_i}{t}\left(\epsilon - \sum_{i=1}^{t-1} g_i x_i\right)$
3: Receive loss ℓ_t and pay $\ell_t(x_t)$
4: Set $g_t \in \partial \ell_t(x_t)$
5: **end for**

引理 9.7. 定义 $f(x) = \beta \exp \frac{x^2}{2\alpha}$, 对于 $\alpha, \beta > 0$, $x \geq 0$. 则

$$f^\star(y) = |y|\sqrt{\alpha W\left(\frac{\alpha y^2}{\beta^2}\right)} - \beta \exp\left(\frac{W\left(\frac{\alpha y^2}{\beta^2}\right)}{2}\right) \leq |y|\sqrt{\alpha W\left(\frac{\alpha y^2}{\beta^2}\right)} - \beta \leq |y|\sqrt{\alpha \ln\left(1 + \frac{\alpha y^2}{\beta^2}\right)} - \beta \ .$$

其中 $W(x)$ 是 *Lambert* 函数, *i.e.* $W : \mathbb{R}_+ \to \mathbb{R}_+$ 定义为满足 $x = W(x)\exp(W(x))$.

证明. 根据 Fenchel dual 的定义, 有

$$f^\star(y) = \max_x \ x\,y - f(x) = \max_x \ x\,y - \beta \exp \frac{x^2}{2\alpha} \leq x^\star\, y - \beta,$$

其中 $x^\star = \operatorname{argmax}_x x\,y - f(x)$. 我们利用 $x^\star$ 满足 $y = f'(x^\star)$ 的事实, 有 $x^\star = \operatorname{sign}(y)\sqrt{\alpha W\left(\frac{\alpha y^2}{\beta^2}\right)}$, 其中 $W(\cdot)$ 是 Lambert 函数. 使用附录中的引理A.2 . 我们得到所规定的界. □

因此, 一维 OLO 算法使用的 KT 的后悔保证的上限为

$$\text{Regret}_T(u) = \sum_{t=1}^T \ell_t(x_t) - \sum_{t=1}^T \ell_t(u) \leq |u|\sqrt{4T \ln\left(\frac{\sqrt{2}|u|KT}{\epsilon} + 1\right)} + \epsilon, \ \forall u \in \mathbb{R},$$

其中唯一的假设是 ℓ_t 的一阶导数（或次阶倒数）的绝对值以 1 为界. 同样, 需要注意的是, 在 $[1, \sqrt{T}]$ 中 ϵ 的任意设置都不会改变渐进率.

为了更好地理解这个后悔, 将这个绑定与学习速率 $\eta = \frac{\alpha}{\sqrt{T}}$ 的 OMD 绑定进行比较:

$$\text{Regret}_T(u) = \sum_{t=1}^T \ell_t(x_t) - \sum_{t=1}^T \ell_t(u) \leq \frac{1}{2}\left(\frac{u^2}{\alpha} + \alpha\right)\sqrt{T}, \ \forall u \in \mathbb{R} \ .$$

因此, 投币方法允许获得几乎最优边界, 而不必猜测正确的学习率! 我们为这个无参数性所付出的代价是 log 因子, 它在我们的下界是最优的.

同样有趣的是, 我们可以看看这个算法在一个简单的问题上能做些什么, 比如 $\ell_t(x) = |x - 10|$. 在图9.1中, 我们展示了 KT 算法和在线次梯度下降 (OSD) 会做的不同预测. 注意 OSD 的收敛速度很大程度上取决于学习速度: 太大不会收敛, 太小会减慢收敛速度. 另一方面,KT 会以指数速度向最小值移动, 然后它会自动后退. 这种指数增长有效地工作, 就像线搜索程序, 允许获得最佳的后悔, 而无需调整学习速率. 在稍后的迭代中,KT 将在最小值附近振荡, 自动收缩它的步骤, 不需要任何参数来调整. 当然, 这是一个简化的例子. 在真正的 OCO 游戏中, 每个时间步骤的损失是不同的, 算法背后的直觉变得更加困难. 然而, 这种最优的后悔让我们确信,KT 战略是正确的战略.

下次, 我们将看到, 我们也可以减少 OCO 在 $\mathbb{R}^d$ 学习与专家投币游戏.

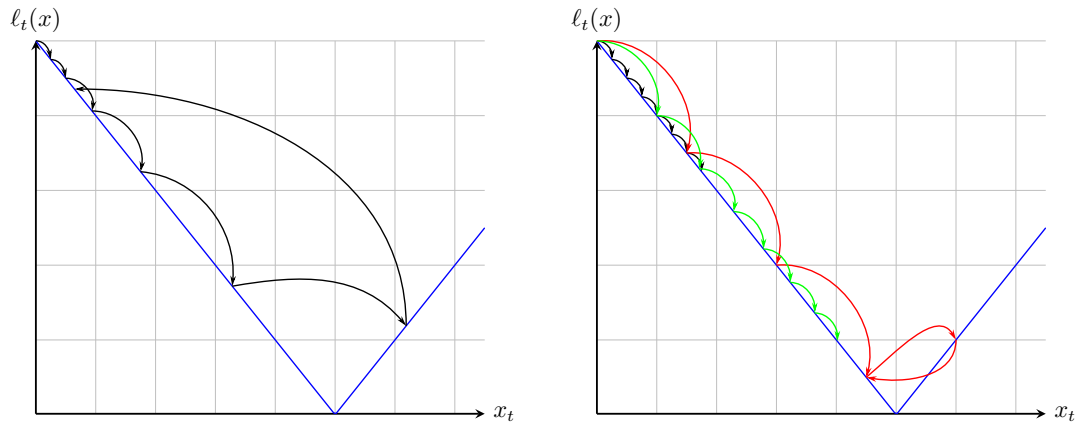

图 9.1: 当 $\ell_t(x) = |x - 10|$ 时, 不同学习速率和相同步数的 KT(左) 和在线梯度下降的行为 (右)

9.3 Coordinate-wise Parameter-free OCO

我们已经看到, 在坐标上分解 OCO 问题并在每个坐标上使用不同的一维在线线性优化算法总是可能的. 特别地, 我们看到

$$\begin{aligned}\mathrm{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \\ &= \sum_{t=1}^T \sum_{i=1}^d g_{t,i}(x_{t,i} - u_i) = \sum_{i=1}^d \sum_{t=1}^T g_{t,i}(x_{t,i} - u_i),\end{aligned}$$

其中 $\sum_{t=1}^T g_{t,i}(x_{t,i} - u_i)$ 正是后悔 w.r.t. 由次梯度的坐标 i 构造的线性损失.

因此, 如果我们有一个一维的 OLO 算法, 我们可以复制它的 d 个副本, 每个副本都由子梯度的坐标 i 提供. 特别地, 我们可以考虑在每个坐标上使用 KT 算法. 这个过程的伪代码在算法9.3中.

Algorithm 9.3 OCO with Coordinate-Wise Krichevsky-Trofimov

Require: $\epsilon > 0$

for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t$ whose coordinates are $x_{t,i} = -\frac{\sum_{j=1}^{t-1} g_{j,i}}{t}\left(\epsilon - \sum_{j=1}^{t-1} g_{j,i} x_{j,i}\right)$ for $i = 1, \dots, d$
  Receive loss ℓ_t and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
end for

我们得到的后悔边界是即时的: 我们只需要在坐标上对后悔求和.

定理 9.8. *根据算法9.3的说明, 设 $\|\boldsymbol{g}_t\|_\infty \le 1$. 则, $\forall \boldsymbol{u} \in \mathbb{R}^d$, 以下后悔边界成立*

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \le \sum_{i=1}^d |u_i| \sqrt{4T \ln\left(1 + \frac{\sqrt{2} u_i^2 K T}{\epsilon}\right)} + d\epsilon \le \|\boldsymbol{u}\|_1 \sqrt{4T \ln\left(1 + \frac{\sqrt{2}\|\boldsymbol{u}\|_\infty^2 K T}{\epsilon}\right)} + d\epsilon,$$

其中 K 是一个普遍常数

注意, 上面的定理表明, 在高维环境下, ϵ 应该和 $\frac{1}{d}$ 成比例.

9.4 无参数的任意范数

上述约简仅适用于有限维空间. 此外, 它对竞争对手有所依赖. L_1 范数也许不合需要, 所以, 这里我们给出了另一个从一维 OCO 到无限维的简单简化.

这种约简需要一种用于一维情况的无约束 OCO 算法和一种用于在 d 维 (或无限维) 球中学习的算法. 对于一维学习者, 我们可以使用 KT 算法, 而对于 d 维的球, 我们可以使用在线镜像下降 (OMD). 有了这两个学习者, 我们把学习向量 $\boldsymbol{x}_t$ 的问题分解为学习方向和大小的问题. 这个程序的后悔结果只是两个学习者后悔的总和. 我们可以用下面的定理来表达这个观点

Algorithm 9.4 Learning Magnitude and Direction Separately

Require: 1d Online learning algorithm $\mathcal{A}_{\text{1d}}$, Online learning algorithm $\mathcal{A}_{\mathcal{B}}$ with feasible set equal to the unit ball $B \subset \mathbb{R}^d$ w.r.t. $\|\cdot\|$

for $t = 1$ **to** T **do**
  Get point $z_t \in \mathbb{R}$ from $\mathcal{A}_{\text{1d}}$
  Get point $\tilde{\boldsymbol{x}}_t \in B$ from $\mathcal{A}_B$
  Play $\boldsymbol{x}_t = z_t \tilde{\boldsymbol{x}}_t \in \mathbb{R}^d$
  Receive $\ell_t : \mathbb{R}^d \to (-\infty, +\infty]$ and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  Set $s_t = \langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle$
  Send $\ell_t^{\mathcal{A}_{\text{1d}}}(x) = s_t x$ as the t-th linear loss to $\mathcal{A}_{\text{1d}}$
  Send $\ell_t^{\mathcal{A}_B}(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$ as the t-th linear loss to $\mathcal{A}_B$
end for

定理 9.9. *Denote by* $\text{Regret}_T^{\mathcal{A}_B}(\boldsymbol{u})$ *the linear regret of algorithm* $\mathcal{A}_B$ *for any* $\boldsymbol{u}$ *in the unit ball w.r.t a norm* $\|\cdot\|$, *and* $\text{Regret}_T^{\mathcal{A}_{1d}}(u)$ *the linear regret of algorithm* $\mathcal{A}_{1d}$ *for any competitor* $u \in \mathbb{R}$. *Then, for any* $\boldsymbol{u} \neq \boldsymbol{0} \in \mathbb{R}^d$, *Algorithm 9.4 guarantees regret*

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle = \text{Regret}_T^{\mathcal{A}_{1d}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \, \text{Regret}_T^{\mathcal{A}_B}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right),$$

和

$$\text{Regret}_T(\boldsymbol{0}) \leq \text{Regret}_T^{\mathcal{A}_{1d}}(0) \ .$$

此外, 次梯度 s_t 发送到 $\mathcal{A}_{1d}$ 满足 $|s_t| \leq \|\boldsymbol{g}_t\|_\star$.

证明. 首先, 观察到对于所有的 t, 有 $|s_t| \le \|\boldsymbol{g}_t\|_\star \|\tilde{\boldsymbol{x}}_t\| \le \|\boldsymbol{g}_t\|_\star$, 因为 $\|\tilde{\boldsymbol{x}}_t\| \le 1$. 现在, 假设 $\boldsymbol{u} \neq \boldsymbol{0}$ 计算:

$$\begin{aligned}
\mathrm{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle = \sum_{t=1}^T \langle \boldsymbol{g}_t, z_t \tilde{\boldsymbol{x}}_t\rangle - \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \\
&= \underbrace{\sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t\rangle z_t - \langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t\rangle \|\boldsymbol{u}\|\right)}_{\text{linear regret of } \mathcal{A}_{1\mathrm{d}} \text{ at } \|\boldsymbol{u}\| \in \mathbb{R}} + \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t\rangle \|\boldsymbol{u}\| - \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle\right) \\
&= \mathrm{Regret}_T^{\mathcal{A}_{1\mathrm{d}}}(\|\boldsymbol{u}\|) + \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t\rangle \|\boldsymbol{u}\| - \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle\right) \\
&= \mathrm{Regret}_T^{\mathcal{A}_{1\mathrm{d}}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t\rangle - \left\langle \boldsymbol{g}_t, \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right\rangle\right) \\
&= \mathrm{Regret}_T^{\mathcal{A}_{1\mathrm{d}}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \,\mathrm{Regret}_T^{\mathcal{A}_B}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right) .
\end{aligned}$$

$\boldsymbol{u} = \boldsymbol{0}$ 的情况类似. □

附注 9.10. *注意, 方向向量没有被限制为范数等于 1, 但这似乎不会影响后悔相等.*

对于一维学习者, 我们可以使用 KT 投注算法和方向学习者的 OMD 算法来实例化上述定理. 观察到, 为了得到 $[-1, 1]$, 中的硬币结果, 我们需要将损失除以 Lipschitz 常数. 我们得到以下示例.

例 9.11. *设 $\mathcal{A}_B$ 是 OSD 满足 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \le 1\}$, 学习率为 $\eta_t = \frac{\sqrt{2}D}{2\sqrt{t}}$. 设 $\mathcal{A}_{1d}$ 是 KT 算法, 且对于 1 维的 OCO 满足 $\epsilon = 1$. 假设损失函数是 1-Lipschitz w.r.t. $\|\cdot\|_2$. 则, 利用算法9.4中的构造, 有*

$$\mathrm{Regret}_T(\boldsymbol{u}) \le O\left(\left(\|\boldsymbol{u}\|_2 \sqrt{\ln(\|\boldsymbol{u}\|_2 T + 1)} + \|\boldsymbol{u}\|_2\right)\sqrt{T} + 1\right), \ \forall \boldsymbol{u} \in \mathbb{R}^d .$$

该算法采用在线到批处理的转换方法, 是一个不需要调整学习速率的随机梯度下降过程.

为了更好地理解这种保证, 让我们来看看 FTRL 的一种 (OSD 只能在学习速率恒定的无界域中使用). 由于正则器的 $\psi_t(\boldsymbol{x}) = \frac{\sqrt{t}}{2\alpha}\|\boldsymbol{x}\|_2$ 和 1-Lipschitz 的损失, 得到了后悔的结果.

$$\mathrm{Regret}_T(\boldsymbol{u}) \le \sqrt{T}\left(\frac{\|\boldsymbol{u}\|_2^2}{2\alpha} + \alpha\right) .$$

因此, 要得到正确的依赖 $\|\boldsymbol{u}\|_2$, 我们需要调整 α, 但我们看到这是不可能的. 另一方面,9.11 例子中的后悔受到了对数因素的影响, 这是不调整参数所付出的代价. 同样地, 我们甚至可以为 L_p norms 提供一个无参数的后悔约束.

例 9.12. *设 $\mathcal{A}_B$ 是 OMD 满足 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_p \le 1\}$, 学习率为 $\eta_t = \frac{\sqrt{2(p-1)}D}{2\sqrt{t}}$. 设 $\mathcal{A}_{1d}$ 是 KT 算法, 且对于 1 维的 OCO 满足 $\epsilon = 1$. 假设损失函数是 1-Lipschitz w.r.t. $\|\cdot\|_q$. 则, 利用算法9.4中的构造, 有*

$$\mathrm{Regret}_T(\boldsymbol{u}) \le O\left(\left(\|\boldsymbol{u}\|_p \sqrt{\ln(\|\boldsymbol{u}\|_p T + 1)} + \frac{\|\boldsymbol{u}\|_p}{\sqrt{p-1}}\right)\sqrt{T} + 1\right), \ \forall \boldsymbol{u} \in \mathbb{R}^d .$$

如果我们想测量竞争者 *w.r.t the* L_1 范数, 我们必须使用和我们看到的 *OMD* 相同的方法: 设 $q = 2\ln d$ 且 p 使得 $1/p + 1/q = 1$. 现在, 假设 $\|\boldsymbol{g}_t\|_\infty \le 1$, 有 $\|\boldsymbol{g}_t\|_q \le d^{1/q}$. 因此, 我们必须将所有损失除以 $d^{1/q}$, 对于所有 $\boldsymbol{u} \in \mathbb{R}^d$, 获得

$$
\begin{aligned}
d^{-1/q}\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\le O\left(\left(\|\boldsymbol{u}\|_p\sqrt{\ln(\|\boldsymbol{u}\|_p T + 1)} + \|\boldsymbol{u}\|_p\sqrt{q-1}\right)\sqrt{T} + 1\right) \\
&\le O\left(\left(\|\boldsymbol{u}\|_1\sqrt{\ln(\|\boldsymbol{u}\|_1 T + 1)} + \|\boldsymbol{u}\|_1\sqrt{\ln d}\right)\sqrt{T} + 1\right) \ .
\end{aligned}
$$

注意, 无参数构造对 $\boldsymbol{u} = \boldsymbol{0}$ 的后悔是恒定的. 重要的是要理解原点并没有什么特别之处: 我们可以通过任意偏移量来转换预测, 并得到一个将偏移量视为不断后悔的点的保证. 下一个命题就说明了这一点.

命题 9.13. *设 $\mathcal{A}$ 是一个 OLO 算法, 预测 $\boldsymbol{x}_t$ 对于任意的 $\boldsymbol{u} \in \mathbb{R}^d$ 保证线性后悔 $\mathrm{Regret}_T^{OLO}(\boldsymbol{u})$. 对于 OCO, 我们得到预测 $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t + \boldsymbol{x}_0$ 的后悔*

$$
\sum_{t=1}^{T}\ell_t(\hat{\boldsymbol{x}}_t) - \sum_{t=1}^{T}\ell_t(\boldsymbol{u}) \le \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t + \boldsymbol{x}_0 - \boldsymbol{u}\rangle = \mathrm{Regret}_T^{OLO}(\boldsymbol{u} - \boldsymbol{x}_0) \ .
$$

9.5 结合在线凸优化算法

最后, 我们现在展示一个无参数 OCO 算法属性的有用应用程序, 它对 $\boldsymbol{u} = \boldsymbol{0}$ 有一个常量的后悔.

定理 9.14. *设 $\mathcal{A}_1$ 和 $\mathcal{A}_2$ 这两个 OLO 算法分别产生预测 $\boldsymbol{x}_{t,1}$ 和 $\boldsymbol{x}_{t,2}$. 然后, 用 $\boldsymbol{x}_t = \boldsymbol{x}_{t,1} + \boldsymbol{x}_{t,2}$ 来预测, 我们有任意 $\boldsymbol{u} \in \mathbb{R}^d$*

$$
\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = \min_{\boldsymbol{u}=\boldsymbol{u}_1+\boldsymbol{u}_2} \ \mathrm{Regret}_T^{\mathcal{A}_1}(\boldsymbol{u}_1) + \mathrm{Regret}_T^{\mathcal{A}_2}(\boldsymbol{u}_2) \ .
$$

而且, 如果这两种算法都保证了对 $\boldsymbol{u} = \boldsymbol{0}$ 有 ϵ 的持续的后悔, 我们对任意 $\boldsymbol{u} \in \mathbb{R}^d$ 有

$$
\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \le \epsilon + \min\left(\mathrm{Regret}_T^{\mathcal{A}_1}(\boldsymbol{u}), \mathrm{Regret}_T^{\mathcal{A}_2}(\boldsymbol{u})\right) \ .
$$

证明. 设 $\boldsymbol{u}_1 + \boldsymbol{u}_2 = \boldsymbol{u}$. 则,

$$
\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_{t,1}\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}_1\rangle + \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_{t,2}\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}_2\rangle \ . \qquad \square
$$

换句话说, 上面的定理让我们可以结合在线学习算法. 如果我们组合的算法对零竞争对手有持续的后悔, 那么我们总是得到两个保证中最好的

例 9.15. *我们可以将两种无参数 OCO 算法结合起来, 一种给出了一个依赖于竞争对手和次梯度的 L_2 范数的界, 另一种专门给出了竞争对手/次梯度的 L_1/L_∞ 范数的界. 上面的定理让我们确信, 我们也会在两者之间得到最好的保证, 在后悔方面只付出一个恒定的因素.*

当然,OCO 后悔的上界与线性后悔的上界一样, 上面的定理也上了 OCO 后悔的上界.

9.6 简化为与专家一起学习

首先, 请记住我们从 OMD(和 FTRL) 得到的后悔是

$$\mathrm{Regret}_T(\boldsymbol{u}) \le O\left(\frac{KL(\boldsymbol{u};\boldsymbol{\pi})}{\eta} + \eta T\right),$$

其中 $\boldsymbol{\pi}$ 为专家上的先验分布, $KL(\cdot;\cdot)$ 是 KL 散度. 正如我们在 OCO 情况下推理的那样, 为了设置学习速率, 我们应该知道 $KL(\boldsymbol{u};\boldsymbol{\pi})$ 的值. 如果我们把 η 设为 $\sqrt{\frac{KL(\boldsymbol{u};\boldsymbol{\pi})}{T}}$, 就会得到 $\sqrt{T\,KL(\boldsymbol{u};\boldsymbol{\pi})}$ 的后悔. 然而, 考虑到游戏的对抗性, 这是不可能的. 因此, 正如我们在 OCO 案例中所做的那样, 我们将证明即使是这个问题也可以简化为对一枚硬币下注, 通过无参数算法获得最佳保证.

附注 9.16. *与专家一起学习的后悔是一种不同的概念. 将所有行为的累积损失从最低到最高排序, 并定义 ϵ-quantile 后悔为学习者的累计损失与排序列表中的 $\lceil \epsilon d \rceil$-th 元素之间的差值. 在公式*

$$\mathrm{Regret}_T(\epsilon) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \sum_{t=1}^{T} g_{t,i_\epsilon},$$

其中 $\epsilon \in [1/d, 1]$ 和 i_ϵ 是 $\lceil \epsilon d \rceil$-th 最佳行为.

当行动的数量非常大且许多行动接近最优时, 这个定义就有意义了. 注意, 随着越来越多的 ϵ. 保证一份小小的后悔变得越来越容易. 所以, 我们想设计一种算法, 它的后悔度与 ϵ 成反比, 而且无论如何都不依赖于 d .

我们现在表明, 依赖于一般竞争对手 $\boldsymbol{u}$ 和 $\boldsymbol{\pi}$ 之间的 KL 散度意味着后悔得到保证. 定义向量 $\boldsymbol{u}_\epsilon$, 其坐标对应于最小的 $\lceil \epsilon d \rceil$ 专家数等于 $\frac{1}{\lceil \epsilon d \rceil}$, 其他坐标为 0 . 假设一个算法, 对于任意损失序列 $\boldsymbol{g}_t \in [0,1]^d$, 保证 $\mathrm{Regret}_T(\boldsymbol{u})$ 后悔上界为 $F_T(KL(\boldsymbol{u};\boldsymbol{\pi}))$, 其中 F 是一个非增长函数. 然后设置 $\boldsymbol{\pi} = [1/d, \ldots, 1/d]$, 有

$$\mathrm{Regret}_T(\epsilon) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \sum_{t=1}^{T} g_{t,i_\epsilon} \le \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u}_\epsilon \rangle \le F_T(KL(\boldsymbol{u}_\epsilon;\boldsymbol{\pi})) = F_T\left(\ln \frac{d}{\lceil \epsilon d \rceil}\right) \le F_T\left(\ln \frac{1}{\epsilon}\right),$$

在第一个不等式中, 我们使用了一组数字的平均值小于该集合中最大的数.

首先, 让我们引入一些符号. 设 $d \ge 2$ 为专家数且 Δ^d 为 d-dimensional probability simplex. 设 $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_d] \in \Delta^d$ 是任意的先验分布. 设 $\mathcal{A}$ 是一个投币算法. 我们将实例化 $\mathcal{A}$ 的 d 个副本.

考虑任意轮次 t. 设 $x_{t,i} \in \mathbb{R}$ 为 $\mathcal{A}$ 的副本的 i-th 投注. 由 LEA 算法计算 $\widehat{\boldsymbol{p}}_t = [\widehat{p}_{t,1}, \widehat{p}_{t,2}, \ldots, \widehat{p}_{t,d}] \in \mathbb{R}_+^d$ 为

$$\widehat{p}_{t,i} = \pi_i \cdot \max(x_{t,i}, 0)\ . \tag{9.1}$$

然后, LEA 算法预测 $\boldsymbol{p}_t = [p_{t,1}, p_{t,2}, \ldots, p_{t,d}] \in \Delta^d$ 为

$$\boldsymbol{p}_t = \begin{cases} \frac{\widehat{\boldsymbol{p}}_t}{\|\widehat{\boldsymbol{p}}_t\|_1}, & \text{if } \widehat{\boldsymbol{p}}_t \ne \boldsymbol{0}, \\ \boldsymbol{\pi}, & \text{otherwise}\ . \end{cases} \tag{9.2}$$

然后, 算法接收到奖励向量 $\boldsymbol{g}_t = [g_{t,1}, g_{t,2}, \ldots, g_{t,d}] \in [0,1]^d$. 最后, 它将奖励给 $\mathcal{A}$ 的每个副本, $\mathcal{A}$ 的第 i 个副本的奖励为 $c_{t,i} \in [-1,1]$ 定义为

$$c_{t,i} = \begin{cases} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i} & \text{if } x_{t,i} > 0, \\ \max\left(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0\right) & \text{if } x_{t,i} \leq 0 \ . \end{cases} \tag{9.3}$$

上面的构造定义了一个由预测 $\boldsymbol{p}_t$ 定义的 LEA 算法, 基于 $\mathcal{A}$ 算法, 我们可以证明它的如下后悔界.

定理 9.17 (专家后悔)**.** *设 $\mathcal{A}$ 是一种投币算法, 对于任意连续的投币结果 $c'_1, \ldots, c't \in [-1,1]$, 将最开始的金额视为 1, 该算法可以保证在 t 轮次之后获得 $\exp(f_t(\sum_{i=1}^t c'_t))$ 的财富, 然后, 在 (9.2)的 $\boldsymbol{p}_t$ 预测每一轮的先验 $\pi \in \Delta^d$ 的 LEA 算法的后悔满足:*

$$\forall T \geq 0, \quad \forall \boldsymbol{u} \in \Delta^d, \qquad \text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{p}_t - \boldsymbol{u} \rangle \leq h\left(KL(\boldsymbol{u}; \boldsymbol{\pi})\right),$$

对于任意 $h : \mathbb{R} \to \mathbb{R}$ 是凹的且非递减的, 使得 $x \leq h(f_T(x))$.

证明. 我们首先证明 $\sum_{i=1}^d \pi_i c_{t,i} x_{t,i} \leq 0$. 事实上,

$$\begin{aligned} \sum_{i=1}^d \pi_i c_{t,i} x_{t,i} &= \sum_{i\,:\,\pi_i x_{t,i} > 0} \pi_i \max(x_{t,i}, 0)(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}) \ + \sum_{i\,:\,\pi_i x_{t,i} \leq 0} \pi_i x_{t,i} \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0) \\ &= \|\widehat{\boldsymbol{p}}_t\|_1 \sum_{i=1}^d p_{t,i}(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \boldsymbol{g}_{t,i}) \ + \sum_{i\,:\,\pi_i x_{t,i} \leq 0} \pi_i x_{t,i} \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0) \\ &= 0 \ + \sum_{i\,:\,\pi_i x_{t,i} \leq 0} \pi_i x_{t,i} \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0) \ \leq 0 \ . \end{aligned}$$

第一个等式来自于 $c_{t,i}$ 的定义. 观察第二个等式, 考虑两种情况: 对于所有的 i, 若 $\pi_i x_{t,i} \leq 0$ 则 $\|\widehat{p}_t\|_1 = 0$, 因此 $\|\widehat{\boldsymbol{p}}_t\|_1 \sum_{i=1}^d p_{t,i}(g_{t,i} - \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle)$ 和 $\sum_{i\,:\,\pi_i x_{t,i} > 0} \pi_i \max(w_{t,i}, 0)(g_{t,i} - \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle)$ 是一个很小的值. 若 $\|\widehat{\boldsymbol{p}}_t\|_1 > 0$ 则对于所有的 i, 有 $\pi_i \max(x_{t,i}, 0) = \widehat{p}_{t,i} = \|\widehat{\boldsymbol{p}}_t\|_1 p_{t,i}$.

根据 $\mathcal{A}$ 的假设, 有, 对于任意序列 $\{c'_t\}_{t=1}^T$ 有 $c'_t \in [-1,1]$ 即:

$$\text{Wealth}_T = 1 + \sum_{t=1}^T c'_t x_t \geq \exp\left(f_T\left(\sum_{t=1}^T c'_t\right)\right) \ . \tag{9.4}$$

故, 不等式 $\sum_{i=1}^d \pi_i c_{t,i} x_{t,i} \leq 0$ 和 (9.4) 表明

$$\sum_{i=1}^d \pi_i \exp\left(f_T\left(\sum_{t=1}^T c_{t,i}\right)\right) \leq 1 + \sum_{i=1}^d \pi_i \sum_{t=1}^T c_{t,i} x_{t,i} \leq 1 \ . \tag{9.5}$$

现在, 对于任意对手 $\boldsymbol{u} \in \Delta^d$,

$$
\begin{aligned}
\mathrm{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t - \boldsymbol{u} \rangle \\
&= \sum_{t=1}^{T} \sum_{i=1}^{d} u_i \left(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i} \right) \\
&\leq \sum_{t=1}^{T} \sum_{i=1}^{N} u_i c_{t,i} \qquad \text{(by definition of } c_{t,i}) \\
&\leq \sum_{i=1}^{N} u_i h \left(f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right) \qquad \text{(definition of the } h(x)) \\
&\leq h \left[\sum_{i=1}^{N} u_i f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right] \qquad \text{(by concavity of } h \text{ and Jensen inequality)} \\
&\leq h \left[KL(\boldsymbol{u}; \boldsymbol{\pi}) + \ln \left(\sum_{i=1}^{N} \pi_i \exp \left(f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right) \right) \right] \qquad \text{(Fenchel-Young inequality)} \\
&\leq h \left(KL(\boldsymbol{u}; \boldsymbol{\pi}) \right) \qquad \text{(by (9.5))} .
\end{aligned}
$$

□

现在, 我们可以用 KT bettor 来证明这个定理. 然而, 我们将得到一个次优的后悔保证. 实际上, 记住 KT 财富的下界, 设 $h(x) = \sqrt{4T(x + \frac{1}{2}\ln T + K)}$ 其中 K 是一个通用常数, 我们就得到了

$$
\mathrm{Regret}_T(\boldsymbol{u}) \leq O(\sqrt{T(KL(\boldsymbol{u}; \boldsymbol{\pi}) + \ln T)}) .
$$

我们可能会认为 $\ln T$ 是我们为了适应未知的竞争对手 $\boldsymbol{u}$ 而必须付出的代价. 然而, 事实证明它是可以被移除的. 在下一节中, 我们将看到如何改变 KT 策略以获得最优保证.

9.6.1 一种赌博策略, 最多输掉一定比例的钱

在之前的简化中, 如果我们使用 KT 投注策略, 我们会在平方根下有一个 $\ln T$ 项. 事实证明, 我们可以避免这个词如果我们事先知道轮数. 那么, 在 T 是未知的情况下, 我们可以使用翻倍的技巧, 只支付一个常数倍的后悔.

后悔中的对数术语来自于财富的下限是

$$
O\left(\frac{1}{\sqrt{T}} \exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{4T} \right) \right) .
$$

注意, 如果序列中正面的个数等于正面的个数, 那么 $\sum_{t=1}^{T} c_t = 0$, 保证财富与 $\frac{1}{\sqrt{T}}$ 成正比. 故, 当 T 趋于无穷时, 赌徒会输掉所有的钱.

相反, 我们需要一个更保守的策略来保证

$$
O\left(\alpha \exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{4T} \right) \right),
$$

当 α 足够小和独立于 T 的在这种情况下, 其打赌策略必须要有节奏, 可能与游戏的持续的知识有关, 因此, 即使是在正面的个数等于尾巴的数量只会失去资金的一小部分. 与此同时, 当硬币结果偏向一边时, 它仍然会获得指数数量的金钱. 我们将证明这是可能的, 设计一个新的投注策略.

对于 $t = 1, \ldots, T$, 由 $S_t = \sum_{i=1}^{t} c_i$ 表示, 定义

$$F_t(x) = \epsilon \exp\left(\frac{x^2}{2(t+T)} - \sum_{i=1}^{t} \frac{1}{2(i+T)}\right), \tag{9.6}$$

$$\beta_t = \frac{F_t(S_{t-1}+1) - F_t(S_{t-1}-1)}{F_t(S_{t-1}+1) + F_t(S_{t-1}-1)} . \tag{9.7}$$

注意, 若

$$(1+\beta_t c_t) F_{t-1}\left(\sum_{i=1}^{t-1} c_i\right) \geq F_t\left(\sum_{i=1}^{t} c_i\right) \tag{9.8}$$

则, 通过归纳, $\text{Wealth}_T \geq F_T\left(\sum_{t=1}^{T} c_t\right)$. 有

$$\text{Wealth}_t = (1+\beta_t c_t)\,\text{Wealth}_{t-1} \geq (1+\beta_t c_t) F_{t-1}\left(\sum_{i=1}^{t-1} c_i\right) \geq F_t\left(\sum_{i=1}^{t} c_i\right) .$$

因此, 我们必须证明 (9.8) 为真, 以保证我们投注策略的最少的财富.

首先, 已知 $\ln(1+\beta_t c) - \frac{(x+c)^2}{2(t+T)}$ 是 c 的凹函数, 有

$$\min_{c\in[-1,1]} \ln(1+\beta_t c) - \frac{(x+c)^2}{2(t+T)} = \min\left(\ln(1+\beta_t) - \frac{(x+1)^2}{2(t+T)}, \ln(1-\beta_t) - \frac{(x-1)^2}{2(t+T)}\right) .$$

同样, 我们的 β_t 选择使得以上两个量等于 $x = S_{t-1}$, 即

$$\ln(1+\beta_t) - \ln F_t(S_{t-1}+1) = \ln(1-\beta_t) - \ln F_t(S_{t-1}-1)$$

对于 β_t, 的其他选项, 两个选项是不同的, 最小的选项总是对手选择的选项. 相反, 让两种选择的最坏结果相等, 我们就可以最小化硬币结果的对抗性选择的损害. 我们得到了这个

$$
\begin{aligned}
\ln(1+\beta_t c_t) - \ln F_t(S_t) &= \ln(1+\beta_t c_t) + F_t(S_{t-1}+c_t) \\
&\geq \min_{c\in[-1,1]} \ln(1+\beta_t c) + \ln F_t(S_{t-1}+c) \\
&= -\ln \frac{F_t(S_{t-1}+1)+F_t(S_{t-1}-1)}{2} \\
&= -\ln\left[\exp\left(\frac{S_{t-1}^2+1}{2(t+T)} - \sum_{i=1}^{t}\frac{1}{2(i+T)}\right)\frac{1}{2}\left(\exp\left(\frac{S_{t-1}}{t+T}\right)+\exp\left(\frac{-S_{t-1}}{t+T}\right)\right)\right] \\
&= -\frac{S_{t-1}^2+1}{2(t+T)} - \ln\cosh\frac{S_{t-1}}{t+T} + \sum_{i=1}^{t}\frac{1}{2(i+T)} \\
&\geq -\frac{S_{t-1}^2}{2(t+T)} - \frac{S_{t-1}^2}{2(t+T)^2} + \sum_{i=1}^{t-1}\frac{1}{2(i+T)} \\
&\geq -\frac{S_{t-1}^2}{2(t+T)} - \frac{S_{t-1}^2}{2(t+T)(t-1+T)} + \sum_{i=1}^{t-1}\frac{1}{2(i+T)} \\
&= -\frac{S_{t-1}^2}{2(t-1+T)} + \sum_{i=1}^{t-1}\frac{1}{2(i+T)} \\
&= -\ln F_{t-1}(S_{t-1}),
\end{aligned}
$$

在第二个等式中我们使用了 β_t 的定义, 在第二个不等式中我们使用了 $\ln\cosh(x) \leq \frac{x^2}{2}, \forall x$.

因此, 若 (9.8)为真, 该策略保证

$$
\text{Wealth}_T \geq \exp\left(\frac{\left(\sum_{t=1}^T c_t\right)^2}{4T} - \sum_{t=1}^{T}\frac{1}{2(i+T)}\right) \geq \exp\left(\frac{\left(\sum_{t=1}^T c_t\right)^2}{4T} - \frac{1}{2}\ln 2\right) = \frac{\sqrt{2}}{2}\exp\left(\frac{\left(\sum_{t=1}^T c_t\right)^2}{2T}\right).
$$

我们现在可以在定理9.17的专家约简中使用这种投注策略, 设 $h(x) = \sqrt{4T(x+\frac{1}{2}\ln 2)}$, 有

$$
\text{Regret}_T(\boldsymbol{u}) \leq \sqrt{4T\left(KL(\boldsymbol{u};\boldsymbol{\pi}) + \frac{1}{2}\ln 2\right)}. \tag{9.9}
$$

注意, 这种投注策略也可以用于 OCO 的减少. 假设我们移除了指数中的对数项, 在一维情况下, 我们会得到一个后悔

$$
\text{Regret}_T(u) \leq O\left(|u|\sqrt{T\ln\left(\frac{|u|\sqrt{T}}{\epsilon}+1\right)+\epsilon}\right),
$$

我们得到的是对数的 $\sqrt{T}$ 项, 而不是 KT 算法的 T 项. 这表明现在我们可以设置 ϵ 为 $\sqrt{T}$, 从而得到 $O(\sqrt{T})$ 的渐进速率, 而不是 $O(\sqrt{T\ln T})$.

9.7 历史片段

Chaudhuri et al. [22]引入了关键字 “无参数” , 用于类似的专家问题学习策略. 现在, 它被用作一个总括的术语, 用于所有在线算法, 这些算法保证最优的后悔统一超过竞争对手类. 一维无参数 OCO

的第一种算法来自 Streeter and McMahan [81], 但其界是次优的. 然后, 该算法被推广到 Orabona [63]中 Hilbert 空间, 仍然有次最优边界. 在 McMahan and Orabona [57]上得到了 Hilbert 空间的最优界.Orabona and Pal [65]引入了使用投币来实现无参数 OCO 的想法. Krichevsky-Trofimov 算法来自于 Krichevsky and Trofimov [46] , 其对 “连续硬币” 的扩展来自于 Orabona and Pal [65]. McMahan and Orabona [57]首次证明了后悔-奖励二元关系. 引理A.2 来自 Orabona and Pal [65].

Streeter and McMahan [81]的第一篇无参数 OLO 论文提出了使用投币算法的坐标版本的方法. 最近, 同样的方法和一个特殊的投币算法也被用于深度神经网络的优化 [67]. 定理9.9来自 Cutkosky and Orabona [30]. 注意, 原来的定理更一般, 因为它甚至适用于 Banach 空间. 结合两种无参数的 OLO 算法以获得这两种保证的最佳效果的想法来自 Cutkosky [29].

Orabona and Pal [65]提出了一种不同的方法, 将投币算法转换为 OCO 算法使其工作在 $\mathbb{R}^d$, 该算法甚至可以在 Hilbert 空间使用. 然而, 这种方法似乎适用于 L_2 范数, 它不是一个 black-box 缩减. 也就是说, Orabona and Pal [65]的减少似乎比定理9.9中有更好的经验表现.

也有一些减少可以将不受约束的 OCO 学习者转变为受约束的 [30]. 他们在域上构造一个 Lipschitz 势垒函数, 并将原始的子梯度和势垒函数的子梯度传递给算法.

专家们的第一个无参数算法来自 Chaudhuri et al. [22], 名为 NormalHedge, 他们引入了 ϵ-quantile 后悔的概念, 得到了 $O(\sqrt{(T+\ln^2 d)(1+\ln\frac{1}{\epsilon})})$ 的界, 但没有对封闭公式进行更新. 然后, Chernov and Vovk [24]删除了对 d 的虚假依赖, 再次使用了一个没有封闭形式的更新. Orabona and Pal [65] 表明, 通过定理9.17中新颖的投币约化, 可以有效地获得这一保证. 后来, 在 Squint 算法 [45] 中, 这些后悔的保证被改进为依赖于损失的平方和而不是时间, 但添加了额外的 $\ln\ln T$ 因子, 值得注意的是, Squint 算法可以准确地解释为投币算法加上定理9.17中的约简.

(9.6) 和 (9.7) 中的投注策略是新的, 来自于 Orabona and Pal [65]中的 shifted-KT potentials . 这种保证是由 shifted-KT potentials 得到的, 但可以在不知道函数性质的情况下进行分析.

Chapter 10

多臂机

多臂机设置类似于专家建议学习 (LEA) 设置: 在每一轮中, 我们选择一个专家 A_t , 不同于全信息设置, 我们只观察到专家 $g_{t,i}$. 的损失, 即 i. 这样做的目的仍然是为了在事后看来, 与累积流失的最优秀专家竞争. 观察到的损失可以是敌对的或随机的, 导致敌对的和随机的多臂机.

10.1 多臂机对抗

就像在学习专家案例中一样, 我们需要随机化以产生次线性的后悔. 事实上, 这是一个比 LEA 更难的问题. 然而, 我们将假设对手是无意识的, 也就是说, 他在游戏开始前决定了所有回合的输掉, 但是他拥有在线算法的知识. 这使得损失成为确定的数量, 当对手是适应性的时, 它避免了我们对后悔的定义的不足 (见 Arora et al. [4]).

这种我们没有得到完整信息的问题, 即我们没有观察到损失向量, 称为 **bandit 问题**. 这个名字来自一个赌徒的问题, 他玩一堆老虎机, 可以被称为“独臂强盗”. 在每一轮中, 赌徒都把赌注押在老虎机上, 他的目标是赢得几乎和他事先知道哪台老虎机会给他最大总回报一样多的钱.

在这个问题中, 我们显然有一个探索-利用的权衡. 事实上, 一方面我们想玩老虎机, 根据之前的回合, 我们相信它会给我们最大的胜利. 另一方面, 我们必须探索老虎机来找到最好的. 在每一轮中, 我们都必须解决这个问题.

考虑到我们不能完全观察到损失, 我们不能使用我们的两个框架: 在线镜像下降 (OMD) 和 Follow-The-Regularized-Leader (FTRL) 都需要损失函数或至少需要它们的下界.

解决这一问题的一种方法是构造未知损失的随机估计. 这是个自然选择假设我们已经知道预测策略是随机的. 在每一轮 t 中, 我们在臂 $\boldsymbol{x}_t$ 上构建一个概率分布我们根据这个概率分布对一个动作 A_t 进行采样. 然后, 我们只观察损失向量 $\boldsymbol{g}_t \in \mathbb{R}^d$ 的坐标 A_t . 一种可能的损失的随机估计是使用一个重要加权估计: 用以下方法构造未知向量 $\boldsymbol{g}_t$ 的估计器 $\tilde{\boldsymbol{g}}_t$:

$$\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{x_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases} .$$

注意, 这个估计量的所有坐标都等于 0, 除了 the coordinate corresponding the arm that was pulled.

这个估计量是无偏的, 为 $\mathbb{E}_{A_t}[\tilde{\boldsymbol{g}}_t] = \boldsymbol{g}_t$. 为了知晓为什么, 注意 $\tilde{g}_{t,i} = \mathbf{1}[A_t = i]\frac{g_{t,i}}{x_{t,i}}$ and $\mathbb{E}_{A_t}[\mathbf{1}[A_t = i]] = x_{t,i}$. 因此, 对于 $i = 1, \ldots, d$, 有

$$\mathbb{E}_{A_t}[\tilde{g}_{t,i}] = \mathbb{E}_{A_t}\left[\mathbf{1}[A_t = i]\frac{g_{t,i}}{x_{t,i}}\right] = \frac{g_{t,i}}{x_{t,i}}\mathbb{E}_{A_t}[\mathbf{1}[A_t = i]] = g_{t,i} \ .$$

让我们也计算这个估计量坐标的 (未居中的) 方差. 我们有

$$\mathbb{E}_{A_t}[\tilde{g}_{t,i}^2] = \mathbb{E}_{A_t}\left[\mathbf{1}[A_t = i]\frac{g_{t,i}^2}{x_{t,i}^2}\right] = \frac{g_{t,i}^2}{x_{t,i}} \ .$$

我们现在可以考虑使用 OMD 与这些估计损失和熵正则器. 因此, 假设 $\|\boldsymbol{g}_t\|_\infty \le L_\infty$ 并将 $\psi : \mathbb{R}_+^d \to \mathbb{R}$ 定义为 $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, 即非标准化负熵. 同样, 设 $\boldsymbol{x}_1 = [1/d, \ldots, 1/d]$. 使用 OMD 分析, 我们有

$$\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u}\rangle \le \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\infty^2 \ .$$

我们可以两边同时取期望, 得到

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^T g_{t,A_t}\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle &= \mathbb{E}\left[\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = \mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u}\rangle\right] \\ &\le \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T \mathbb{E}[\|\tilde{\boldsymbol{g}}_t\|_\infty^2] \le \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T\sum_{i=1}^d \frac{g_{t,i}^2}{x_{t,i}} \ . \end{aligned} \tag{10.1}$$

我们现在有麻烦了, 因为求和尺度中的各项为 $\max_{i=1,\ldots,d} \frac{1}{x_{t,i}}$. 所以, 我们需要一种方法来控制手臂的最小概率.

一种方法是, 取 $\boldsymbol{x}_t$ 和均匀概率的凸组合. 即, 我们可以预测 $\tilde{\boldsymbol{x}}_t = (1-\alpha)\boldsymbol{x}_t + \alpha[1/d, \ldots, 1/d]$, 其中 α 将在下面被选择. 因此, α 可以被看作是我们对算法所需要的最小探索量. 它的价值将通过后悔分析来选择, 以最优地权衡勘探与开发. 得到的算法在算法 p10.1中.

Algorithm 10.1 Exponential Weights with Explicit Exploration for Multi-Armed Bandit

Require: $\eta, \alpha > 0$

Set $\boldsymbol{x}_1 = [1/d, \ldots, 1/d]$
for $t = 1$ **to** T **do**
  Set $\tilde{\boldsymbol{x}}_t = (1-\alpha)\boldsymbol{x}_t + \alpha[1/d, \ldots, 1/d]$
  Draw A_t according to $P(A_t = i) = \tilde{x}_{t,i}$
  Select expert A_t
  Observe *only* the loss of the selected arm $g_{t,A_t} \in [-L_\infty, L_\infty]$ and pay it
  Construct the estimate $\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{\tilde{x}_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases}$
  $x_{t+1,i} \propto x_{t,i}\exp(-\eta\tilde{g}_{t,i}),\ i = 1, \ldots, d$
end for

在估计器中使用相同的概率分布:

$$\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{\tilde{x_{t,i}}}, & i = A_t \\ 0, & \text{otherwise} \end{cases} . \tag{10.2}$$

我们得到 $\frac{1}{\tilde{x}_{t,i}} \le \frac{d}{\alpha}$. 然而, 我们在引入的偏见中付出代价:

$$\sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \tilde{\boldsymbol{x}}_t - \boldsymbol{u} \rangle = \sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, (1-\alpha)\boldsymbol{x}_t - \boldsymbol{u} \rangle + \frac{\alpha}{d} \sum_{t=1}^{T} \sum_{i=1}^{d} \tilde{g}_{t,i} \ .$$

已知 $\mathbb{E}[\sum_{i=1}^{d} \tilde{g}_{t,i}] = \sum_{i=1}^{d} g_{t,i} \le dL_\infty$, 有

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \tilde{\boldsymbol{x}}_t - \boldsymbol{u} \rangle\right] \le \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, (1-\alpha)\boldsymbol{x}_t - \boldsymbol{u} \rangle\right] + \alpha L_\infty T \le \mathbb{E}[\mathrm{Regret}_T(\boldsymbol{u})] + \alpha L_\infty T \ .$$

将 (10.1)中的最后一个不等式和预期后悔的上限放在一起, 我们得到了

$$\mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \le \frac{\ln d}{\eta} + \frac{\eta d^2 L_\infty^2 T}{2\alpha} + \alpha L_\infty T \ .$$

设 $\alpha \propto \sqrt{d^2 L_\infty \eta}$ 且 $\eta \propto \left(\frac{\ln d}{dL_\infty^{3/2} T}\right)^{2/3}$, 得到 $O(L_\infty (dT)^{2/3} \ln^{1/3} d)$ 的悔值.

这比完全信息 $O(\sqrt{T \ln d})$ 的情况要糟糕得多. 然而, 虽然预计强盗情况比完全信息情况更困难, 但事实证明这并不是最优策略.

10.1.1 用于探索和利用的指数权重算法: Exp3

事实证明, 上面的算法实际上是有效的, 即使没有与均匀分布混合! 我们只是对后悔的保证太松了. 我们将分析下面的算法, 这就是所谓的探索和利用指数权算法 (Exp3), 它只不过是带有熵正则化和损失随机估计的 OMD. 注意, 现在我们假设 $g_{t,i} \in [0, L_\infty]$.

Algorithm 10.2 Exp3

Require: $\eta > 0$

$\boldsymbol{x}_1 = [1/d, \ldots, 1/d]$
for $t = 1$ **to** T **do**
  Draw A_t according to $P(A_t = i) = x_{t,i}$
  Select expert A_t
  Observe *only* the loss of the selected arm $g_{t,A_t} \in [0, L_\infty]$ and pay it
  Construct the estimate $\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{x_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases}$
  $x_{t+1,i} \propto x_{t,i} \exp(-\eta \tilde{g}_{t,i}),\ i = 1, \ldots, d$
end for

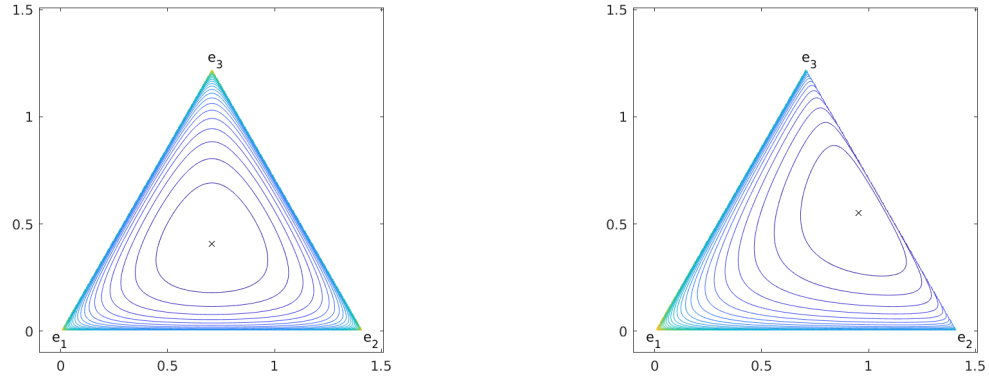

图 10.1: 当 $\boldsymbol{x}_t = [1/3, 1/3, 1/3]$ (左) 和 $\boldsymbol{x}_t = [0.1, 0.45, 0.45]$ (右) 时,KL 散度的三维等值线图.

这一次, 我们将为 OMD 使用本地规范后悔绑定. 这是因为, 当我们使用强凸性时, 我们是正则化器的 Hessian 的最小特征值的倒数和中的项的上界. 然而, 如果我们考虑到当地的规范, 我们可以做得更好. 事实上, 在 $\boldsymbol{x}_t$ 较小的坐标中, 散度的增长较小. 这也可以在图10.1. 中看到. 的确, 对于熵正则化器, 我们有 Hessian 是一个对角矩阵:

$$(\nabla^2 \psi(\boldsymbol{x}))_{ii} = \frac{1}{x_i} \; i = 1, \ldots, d \; .$$

在 $t = 1, \ldots, T$ 的情况下, 引理6.14的不等式的和, 上面的 Hessian 表达式给出了给出了后悔:

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{i=1}^{d} z_{t,i} g_{t,i}^2,$$

其中 $\boldsymbol{z}_t = \alpha_t \boldsymbol{x}_t + (1 - \alpha_t) \boldsymbol{x}_{t+1}$ 且 $\alpha_t \in [0, 1]$. 注意, 对于任意的 $\alpha_t \in [0, 1]$ $\boldsymbol{z}_t$ 是单一的, 所以这个上界总是优于

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2,$$

我们只是利用熵正则化的强凸性推导出来的.

但我们不知道 $\boldsymbol{z}_t$ 的确切值, 只知道它在 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+1}$ 之间的线段上. 然而, 如果你可以说 $z_{t,i} = O(x_{t,i})$, 在 bandit 的情况下我们将获得 $O(\sqrt{dT \ln d})$ 的预期后悔保证, 极大地改进了我们上面证明的界! 换句话说, 有可能得到后悔担保

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \leq \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{i=1}^{d} x_{t,i} g_{t,i}^2, \; \forall \boldsymbol{u} \in V, \tag{10.3}$$

其中 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \geq 0\}$. 幸运的是, 我们有定理6.14 来帮助我们.

附注 10.1. *使用熵正则化器的特定性质而不是引理6.14从第一性原理证明 (10.3) 是可能的. 但是, 我们宁愿不这样做, 因为这样的证明不能说明实际发生的情况.*

为了使用引理6.14, 我们需要一个位于 $\boldsymbol{x}_t$ 和 $\tilde{\boldsymbol{x}}_t = \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}_+^d} \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x};\boldsymbol{x}_t)$ 之间线段上的 $z'_{t,i}$ 来处理. 考虑到我们只需要一个上限, 我们可以只看 $x_{t,i}$ and $\tilde{x}_{t+1,i}$, 看看哪一个更大. 这很容易做到: 使用定义 $\tilde{\boldsymbol{x}}_t$, 我们有

$$\ln(\tilde{x}_{t+1,i}) + 1 = \ln(x_{t,i}) + 1 - \eta g_{t,i},$$

即

$$\tilde{x}_{t+1,i} = x_{t,i}\exp(-\eta g_{t,i}) \ .$$

假设 $g_{t,i} \geq 0$, 有 $\tilde{x}_{t+1,i} \leq x_{t,i}$ 说明 $z_{t,i} \leq x_{t,i}$.

总的来说, 我们有以下改进的后悔保证与专家学习设置正损失.

定理 10.2. *对于 $t = 1,\ldots,T$ 和 $i = 1,\ldots,d$, 假设 $g_{t,i} \geq 0$. 设 $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \geq 0\}$ 且 $\eta > 0$. 利用 OMD 的熵正则器 $\psi : \mathbb{R}_+^d \to \mathbb{R}$ 定义为 $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, 学习率为 η, 且 $\boldsymbol{x}_1 = [1/d,\ldots,1/d]$ 给出以下后悔保证*

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \leq \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^T\sum_{i=1}^d x_{t,i} g_{t,i}^2, \ \forall \boldsymbol{u} \in V \ .$$

有了这个新工具, 我们现在可以再次讨论多臂机问题了.

现在让我们考虑带有熵正则化、学习速率 η, 设 $\tilde{\boldsymbol{g}}_t$ 等于 $\boldsymbol{g}_t$ 的 stochastic 估计, 就像算法10.2中的一样. 运用定理10.2 求期望值, 有

$$\mathbb{E}\left[\sum_{t=1}^T g_{t,A_t}\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = \mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u}\rangle\right] \leq \frac{B_\psi(\boldsymbol{u};\boldsymbol{x}_1)}{\eta} + \mathbb{E}\left[\sum_{t=1}^T\sum_{i=1}^d x_{t,i}\tilde{g}_{t,i}^2\right] \ .$$

现在, 注意 $\mathbb{E}[x_{t,i}\tilde{g}_{t,i}^2]$ 项, 有

$$\mathbb{E}\left[\sum_{i=1}^d x_{t,i}\tilde{g}_{t,i}^2\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^d x_{t,i}\tilde{g}_{t,i}^2 \middle| A_1,\ldots,A_{t-1}\right]\right] = \mathbb{E}\left[\sum_{i=1}^d x_{t,i}\frac{g_{t,i}^2}{x_{t,i}}\right] \leq dL_\infty^2 \ . \tag{10.4}$$

设 $\eta \propto \sqrt{\frac{\ln d}{L_\infty^2 T}}$, 有

$$\mathbb{E}\left[\sum_{t=1}^T g_{t,A_t}\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \leq O\left(L_\infty\sqrt{dT\ln d}\right) \ .$$

因此, 通过更严密的分析, 我们表明, 即使没有明确的探索术语, 带有熵正则化器的 OMD 解决了多臂强盗问题, 只比完整信息情况多出一个因子 $\sqrt{d}$ 然而, 这还不是最理想的后悔!

在下一节中, 我们将看到, 使用相同的分析来更改正则化器, 将删除后悔中的 $\sqrt{\ln d}$ 项.

10.1.2 使用 Tsallis Entropy 的 OMD 的最佳后悔

在本节中, 我们提出了隐式归一化预测器 (INF), 也称为 OMD 与 Tsallis 熵的多臂机.

定义 $\psi_q : \mathbb{R}_+^d \to \mathbb{R}$ 是 $\psi_q(\boldsymbol{x}) = \frac{1}{1-q}\left(1 - \sum_{i=1}^d x_i^q\right)$, 其中 $q \in [0,1]$ 且在 $q = 1$ 中, 我们通过连续性来扩展函数. 这是向量 $\boldsymbol{x}$ 的负的 **Tsallis entropy** . 这是香农熵的严格推广, 因为当 q 趋于 1 时, $\psi_q(\boldsymbol{x})$ 收敛于 $\boldsymbol{x}$ 的负的香农熵.

Algorithm 10.3 INF Algorithm

Require: $\eta > 0$

$\boldsymbol{x}_1 = [1/d, \ldots, 1/d]$

for $t = 1$ **to** T **do**

  Draw A_t according to $P(A_t = i) = x_{t,i}$

  Select expert A_t

  Observe *only* the loss of the selected arm $g_{t,A_t} \in [0, L_\infty]$ and pay it

  Construct the estimate $\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{x_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases}$

  $\tilde{\boldsymbol{x}}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in V} \ \eta \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x} \rangle - \frac{1}{1-q} \sum_{i=1}^d x_i^q + \frac{q}{1-q} \sum_{i=1}^d x_{t,i}^{q-1} x_i$

end for

我们将用这个正则化器为多臂问题实例化 OMD, 就像在算法10.3中那样.

注意 $\operatorname{argmin}_{\boldsymbol{x}} \ \psi_q(\boldsymbol{x}) = \frac{1}{d}$ 和 $\min_{\boldsymbol{x}} \psi_q(\boldsymbol{x}) = \frac{1-d^{1-q}}{1-q}$.

我们将不会使用任何从信息论观点的正规化的解释. 我们将在下面看到, 选择它的唯一原因是它的 Hessian. 事实上, 这个正规化器的 Hessian 线仍然是对角线, 它等于

$$(\nabla^2 \psi_q(\boldsymbol{x}))_{ii} = q \frac{1}{x_i^{2-q}} \ .$$

现在, 我们再用一遍6.14. 故, 对于任意 $\boldsymbol{u} \in V$, 有

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \le \frac{B_{\psi_q}(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^T \boldsymbol{g}_t^\top (\nabla^2 \psi_q(\boldsymbol{z}_t'))^{-1} \boldsymbol{g}_t = \frac{d^{1-q} - \sum_{i=1}^d u_i^q}{\eta(1-q)} + \frac{\eta}{2} \sum_{t=1}^T \sum_{i=1}^d g_{t,i}^2 (z_{t,i}')^{2-q},$$

其中 $\boldsymbol{z}_t' = \alpha_t \boldsymbol{x}_t + (1-\alpha_t) \tilde{\boldsymbol{x}}_{t+1}$, $\alpha_t \in [0,1]$, 和 $\tilde{\boldsymbol{x}}_t = \operatorname{argmin}_{\boldsymbol{x} \in \mathbb{R}_+^d} \ \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_{\psi_q}(\boldsymbol{x}; \boldsymbol{x}_t)$.

就像我们对 Exp3 做的那样, 现在我们需要一个 $z_{t,i}'$ 的上界. 从 $\tilde{\boldsymbol{x}}_t$ 和 ψ 的定义来看, 有

$$-\frac{q}{1-q} \tilde{x}_{t+1,i}^{q-1} = -\frac{q}{1-q} x_{t,i}^{q-1} - \eta g_{t,i},$$

即

$$\tilde{x}_{t+1,i} = \frac{x_{t,i}}{\left(1 + \frac{1-q}{q} \eta g_{t,i} x_{t,i}^{1-q}\right)^{\frac{1}{1-q}}} \ .$$

故, 若 $g_{t,i} \ge 0$, $\tilde{x}_{t+1,i} \le x_{t,i}$, 表明 $z_{t,i}' \le x_{t,i}$.

因此, 把所有这些放在一起, 我们做到了

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \le \frac{d^{1-q} - \sum_{i=1}^d u_i^q}{\eta(1-q)} + \frac{\eta}{2q} \sum_{t=1}^T \sum_{i=1}^d g_{t,i}^2 x_{t,i}^{2-q} \ .$$

我们现在可以专门化上述推理, 考虑 Tsallis 熵中的 $q = 1/2$ 有以下定理.

定理 10.3. *假设 $g_{t,i} \in [0, L_\infty]$. 设 $q = 1/2$ 且 $\boldsymbol{x}_1 = [1/d, \ldots, 1/d]$. 则, 算法10.3*

$$\mathbb{E}\left[\sum_{t=1}^T g_{t,A_t}\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \le \frac{2\sqrt{d}}{\eta} + \eta \sqrt{d} L_\infty^2 T \ .$$

证明. 我们只需要计算这些项

$$\mathbb{E}\left[\sum_{i=1}^{d}\tilde{g}_{t,i}^2 x_{t,i}^{3/2}\right] .$$

就像 (10.4)中的过程一样, 有

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{d}\tilde{g}_{t,i}^2 x_{t,i}^{3/2}\right] &= \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}^{3/2}\tilde{g}_{t,i}^2 \middle| A_1,\ldots,A_{t-1}\right]\right] = \mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}^{3/2}\frac{g_{t,i}^2}{x_{t,i}^2}x_{t,i}\right] = \mathbb{E}\left[\sum_{i=1}^{d} g_{t,i}^2\sqrt{x_{t,i}}\right] \\ &\le \mathbb{E}\left[\sqrt{\sum_{i=1}^{d} g_{t,i}^2}\sqrt{\sum_{i=1}^{d} g_{t,i}^2 x_{t,i}}\right] \le L_\infty\sqrt{d} .\end{aligned}$$

□

选择 $\eta \propto \frac{1}{L_\infty\sqrt{T}}$, 我们最终得到了一个 $O(L_\infty\sqrt{dT})$ 的期望后悔, 它可以被证明是最优的.

还有最后一件事: 我们如何计算这个算法的预测? 在每一步中, 我们都要解决一个约束优化问题. 因此, 我们可以写出相应的拉格朗日量:

$$L(\boldsymbol{x},\beta) = \sum_{i=1}^{d}\left(\eta\tilde{g}_{t,i} + \frac{q}{1-q}x_{t,i}^{q-1}\right)x_i - \frac{1}{1-q}\sum_{i=1}^{d}x_i^q + \beta\left(\sum_{i=1}^{d}x_i - 1\right) .$$

从 KKT 的情况来看, 我们有

$$x_{t+1,i} = \left[\frac{1-q}{q}\left(\beta + \frac{q}{1-q}x_{t,i}^{q-1} + \eta\tilde{g}_{t,i}\right)\right]^{\frac{1}{q-1}} .$$

我们也知道 $\sum_{i=1}^{d} x_{t+1,i} = 1$. 因此, 我们有一个一维的问题在 β 中, 必须在每一轮解决.

10.2 随机臂

今天, 我们将讨论随机强盗设定. 这里, 每个臂都与一个未知的概率分布相关联. 在每个时间步, 算法选择一个手臂 A_t, 它收到一个损失 (或奖励) g_{t,A_t}, 从手臂 A_t 的分布中抽取 i.i.d. 我们关注的是最小化伪后悔, 即对预期中的最佳行动的后悔, 而不是对实现损失顺序的最佳行动:

$$\text{Regret}_T := \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \min_{i=1,\ldots,d}\ \mathbb{E}\left[\sum_{t=1}^{T} g_{t,i}\right] = \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \min_{i=1,\ldots,d}\ \mu_i,$$

其中, 我们用 μ_i 表示与臂 i 相关的分布的期望.

附注 10.4. *在随机臂文献中, 通常的符号是考虑回报而不是损失. 相反, 为了使我们的符号与 OCO 文献保持一致, 我们将考虑损失. 这两项完全等价于 -1 的乘法.*

在给出我们的第一个随机臂算法之前, 我们将介绍一些关于浓度不等式的基本概念, 这些概念对我们的定义和证明是有用的.

10.2.1 Concentration Inequalities Bits

假设 $X_1, X_2, \ldots, X_n$ 是一个独立和同分布的随机变量序列, 其均值为 $\mu = \mathbb{E}[X_1]$, 方差为 $\sigma = \mathrm{Var}[X_1]$. 观察到 $X_1, X_2, \ldots, X_n$ 将估算到共同均值 μ. 最自然的估计是经验均值

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i \ .$$

期望的线性表明 $\mathbb{E}[\hat{\mu}] = \mu$, 意味着 $\hat{\mu}$ 是 μ 的无偏估计. 然而, $\hat{\mu}$ 本身就是一个随机变量. 那么, 我们能量化 $\hat{\mu}$ 和 μ 之间的距离吗?

我们可以用切比雪夫不等式来计算 $\hat{\mu}$ 和 μ 之间距离概率的上界:

$$\mathbb{P}[|\hat{\mu} - \mu| \geq \epsilon] \leq \frac{\mathrm{Var}[\hat{\mu}]}{\epsilon^2} \ .$$

利用事实 $\mathrm{Var}[\hat{\mu}] = \frac{\sigma^2}{n}$, 有

$$\mathbb{P}[|\hat{\mu} - \mu| \geq \epsilon] \leq \frac{\sigma^2}{n\epsilon^2} \ .$$

因此, 我们可以预期" 坏" 估计的概率趋近于 0, 即 1 /经验均值中的样本数. 这是我们能找到的最好的吗? 为了理解我们希望得到什么, 让我们看一下中心极限定理.

我们知道, 定义 $S_n = \sum_{t=1}^{n}(X_i - \mu)$, $\frac{S_n}{\sqrt{n\sigma^2}} \to N(0,1)$ 是标准高斯分布, 当 n 趋于无穷, 这意味着

$$\mathbb{P}[\hat{\mu} - \mu \geq \epsilon] = \mathbb{P}[S_n \geq n\epsilon] = \mathbb{P}\left[\frac{S_n}{\sqrt{2\sigma^2}} \geq \sqrt{\frac{n}{\sigma^2}}\epsilon\right] \approx \int_{\epsilon\sqrt{\frac{n}{\sigma^2}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx,$$

这个近似来自中心极限定理. 这个积分不能用一个封闭的形式计算, 但我们可以很容易地得到它的上界. 的确, 对于 $a > 0$, 有

$$\int_a^{\infty} \exp\left(-\frac{x^2}{2}\right) dx = \int_a^{\infty} \frac{x}{x} \exp\left(-\frac{x^2}{2}\right) dx \leq \frac{1}{a}\int_a^{\infty} x \exp\left(-\frac{x^2}{2}\right) dx = \frac{1}{a}\exp\left(-\frac{a^2}{2}\right) \ .$$

因此, 有

$$\mathbb{P}[\hat{\mu} - \mu \geq \epsilon] \lessapprox \sqrt{\frac{\sigma^2}{2\pi\epsilon^2 n}} \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right) \ . \tag{10.5}$$

这比我们用切比雪夫不等式得到的要好于我们想用相似的渐近速率得到一个精确的界. 为此, 我们将注意力集中在亚高斯随机变量上.

定义 10.5 (Subgaussian Random Variable)**.** *我们说一个随机变量是 σ-**subgaussian** , 若对所有的 $\lambda \in \mathbb{R}$, 有 $\mathbb{E}[\exp(\lambda X)] \leq \exp(\lambda^2\sigma^2/2)$.*

例 10.6. *以下是亚高斯随机变量:*

- *若 X 是均值为 0, 方差为 σ^2 的 Gaussian 函数, 则 X 是 σ 亚高斯函数.*
- *若 X 均值为 0, 且 $X \in [a,b]$ 为真, 则 X 是 $(b-a)/2$ 亚高斯函数.*

对于亚高斯随机变量, 我们有以下性质

引理 10.7 ([51, Lemma 5.4])**.** *设 X_1 和 X_2 是相互独立的, 且 σ_1 和 σ_2 是亚高斯函数, 则*

(a) $\mathbb{E}[X_1] = 0$ *且* $\mathrm{Var}[X_1] \le \sigma_1^2$.

(b) cX_1 *是* $|c|\sigma_1$*-亚高斯函数.*

(c) $X_1 + X_2$ *是* $\sqrt{\sigma_1^2 + \sigma_2^2}$ *亚高斯函数.*

亚高斯随机变量和高斯随机变量一样, 它们的尾部概率的上界是方差为 σ^2 的高斯随机变量的尾部概率. 为了证明它, 我们首先要证明马尔可夫不等式.

定理 10.8 (Markov's inequality)**.** *对于一个非负的随机变量 X 和 $\epsilon > 0$, 有* $\mathbb{P}[X \ge \epsilon] \le \frac{\mathbb{E}[X]}{\epsilon}$.

根据马尔科夫不等式, 现在我们可以把上述关于亚高斯随机变量的陈述形式化.

定理 10.9. *如果一个随机变量是 σ 亚高斯函数, 则* $\mathbb{P}[X \ge \epsilon] \le \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$.

证明. 对于任意的 $\lambda > 0$, 有

$$\mathbb{P}[X \ge \epsilon] = \mathbb{P}[\exp(\lambda X) \ge \exp(\lambda\epsilon)] \le \frac{\mathbb{E}[\exp(\lambda X)]}{\exp(\lambda\epsilon)} \le \exp(\lambda^2\sigma^2/2 - \lambda\epsilon) \ .$$

将不等式 w.r.t. λ 的右边最小化, 得到所述的结果 □

上述定理的一个简单结果是, 次高斯随机变量的经验平均集中在其期望周围, 渐进率与 (10.5)相同.

推论 10.10. *假设 $X_i - \mu$ 是独立的, σ 是亚高斯函数随机变量. 则, 对于任意的 $\epsilon \ge 0$, 有*

$$\mathbb{P}\left[\hat{\mu} \ge \mu + \epsilon\right] \le \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right) \qquad \textit{and} \qquad \mathbb{P}\left[\hat{\mu} \le \mu - \epsilon\right] \le \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right),$$

其中 $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i$.

在 δ 的推论中等于不等式的上界, 我们有等价的表述, 至少有 $1 - 2\delta$ 的概率, 有

$$\mu \in \left[\hat{\mu} - \sqrt{\frac{2\sigma^2 \ln\frac{1}{\delta}}{n}}, \hat{\mu} + \sqrt{\frac{2\sigma^2 \ln\frac{1}{\delta}}{n}}\right] \ .$$

10.2.2 Explore-Then-Commit Algorithm

现在, 我们准备为随机强盗设置呈现最自然的算法, 即探索然后提交 (等等) 算法. 也就是说, 我们首先在 md 探索回合中确定最好的武器, 然后我们就对它做出承诺. 该算法总结在算法10.4中.

在下面, 我们用 $S_{t,i} = \sum_{j=1}^t \mathbf{1}[A_t = i]$ 表示, 这是在第一轮 t 中, 手臂 i 被拉动的次数.

由 $\mu^\star$ 定义以最小的期望失去手臂, 即 $\min_{i=1,\ldots,d} \mu_i$. 我们分析中的关键数量将是差距, $\Delta_i := \mu_i - \mu^\star$ 对于 $i = 1, \ldots, d$, 用来衡量武器和最优武器之间损失的预期差异. 特别地, 我们可以将后悔分解为手臂的预期次数乘以手臂的间隙的总和.

Algorithm 10.4 Explore-Then-Commit Algorithm

Require: $T, m \in \mathbb{N}, 1 \le m \le \frac{T}{d}$

$S_{0,i} = 0, \hat{\mu}_{0,i} = 0,\ i = 1, \dots, d$

for $t = 1$ **to** T **do**

  Choose $A_t = \begin{cases} (t \mod d) + 1, & t \le dm \\ \operatorname{argmin}_i\ \hat{\mu}_{dm,i}, & t > dm \end{cases}$

  Observe g_{t,A_t} and pay it

  $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$

  $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{j,A_j} \mathbf{1}[A_j = i],\ i = 1, \dots, d$

end for

引理 10.11. 对于任何选择手臂的政策, 后悔的上限是

$$\mathrm{Regret}_T = \sum_{i=1}^{d} \mathbb{E}[S_{T,i}]\Delta_i \ .$$

证明. 已知

$$\sum_{t=1}^{T} g_{t,A_t} = \sum_{t=1}^{T}\sum_{i=1}^{d} g_{t,i}\mathbf{1}[A_t = i] \ .$$

因此,

$$\begin{aligned}
\mathrm{Regret}_T &= \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - T\mu^\star = \mathbb{E}\left[\sum_{t=1}^{T}(g_{t,A_t} - \mu^\star)\right] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)] \\
&= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}[\mathbb{E}[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)|A_t]] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}[\mathbf{1}[A_t = i]\mathbb{E}[g_{t,i} - \mu^\star|A_t]] \\
&= \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}[\mathbf{1}[A_t = i](\mu_{A_t} - \mu^\star)] = \sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}[\mathbf{1}[A_t = i]](\mu_i - \mu^\star) \ .
\end{aligned}$$

□

上面的引理量化了这样一种直觉: 为了有一点后悔, 我们必须选择次优臂的次数少于最佳臂的次数. 我们现在准备证明 ETC 算法的后悔保证.

定理 10.12. 假设臂的损失减去它们的期望是 1-*subgaussian*, 且 $1 \le m \le T/d$. 则, *ETC* 保证一个后悔

$$\mathrm{Regret}_T \le m\sum_{i=1}^{d}\Delta_i + (T - md)\sum_{i=1}^{d}\Delta_i \exp\left(-\frac{m\Delta_i^2}{4}\right) \ .$$

证明. 让我们不失一般性地假设最优臂是第一个.

故, 对于 $i \ne 1$, 有

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}[\mathbf{1}[A_t = i]] &= m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,i} \le \min_{j \ge i}\ \hat{\mu}_{md,j}\right] \\
&\le m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,i} \le \hat{\mu}_{md,1}\right] \\
&= m + (T - md)\mathbb{P}\left[\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \ge \Delta_i\right] \ .
\end{aligned}$$

根据引理10.7, 有 $\hat{\mu}_{md,i} - \mu_i - (\hat{\mu}_{md,1} - \mu_1)$ 是 $\sqrt{2/m}$-subgaussian. 故, 根据引理10.9, 有

$$\mathbb{P}\left[\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \geq \Delta_i\right] \leq \exp\left(-\frac{m\Delta_i^2}{4}\right) . \qquad \square$$

边界显示了探索和利用之间的权衡: 如果 m 太大, 我们在勘探阶段 (边界中的第一项) 付出的代价太高. 另一方面, 如果 m 较小, 选择次优臂的概率增大 (界中的第二项). 已知所有的间隙 Δ_i, 有可能选择使边界最小的 m . 例如, 在 $d = 2$ 的情况下, 后悔的上限是

$$m\Delta + (T - 2m)\Delta \exp\left(-m\frac{\Delta^2}{4}\right) \leq m\Delta + T\Delta \exp\left(-m\frac{\Delta^2}{4}\right),$$

它被最小化了

$$m = \frac{4}{\Delta^2} \ln \frac{T\Delta^2}{4} .$$

记住 m 一定是一个我们可以选择的自然数

$$m = \max\left(\left\lceil \frac{4}{\Delta^2} \ln \frac{T\Delta^2}{4} \right\rceil, 1\right) .$$

当 $\frac{T\Delta^2}{4} \leq 1$, 我们选择 $m = 1$. 故, 有 $\Delta + (T - 2)\Delta \leq T\Delta \leq \frac{4}{\Delta}$. 因此, 后悔上界为

$$\min\left(\Delta T, \frac{4}{\Delta}\left(1 + \max\left(\ln \frac{T\Delta^2}{4}, 0\right)\right) + \Delta\right) .$$

该算法的主要缺点是其最优调整依赖于缺口 δ_i. 假设有了差距的知识, 就可以使随机强盗问题完全不重要. 然而, 其调整后的后悔界限为我们提供了比较其他 bandit 算法的基线. 特别地, 在下一节中, 我们将提出一种算法, 在不知道任何差距的情况下实现相同的渐近后悔.

10.2.3 上置信界算法

Algorithm 10.5 上置信界算法

Require: $\alpha > 2, T \in \mathbb{N}$

$S_{0,i} = 0, \hat{\mu}_{0,i} = 0,\ i = 1, \ldots, d$

for $t = 1$ **to** T **do**

  Choose $A_t = \operatorname{argmin}_{i=1,\ldots,d} \begin{cases} \mu_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}}, & \text{if } S_{t-1,i} \neq 0 \\ -\infty, & \text{otherwise} \end{cases}$

  Observe g_{t,A_t} and pay it

  $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$

  $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{t,A_t} \mathbf{1}[A_t = i],\ i = 1, \ldots, d$

end for

ETC 算法的缺点是需要了解裂缝信息来调整勘探阶段. 此外, 它还以一种笨拙的方式解决了探索与开发之间的权衡. 最好有一种算法, 它能以依赖数据的方式从一个阶段平稳地过渡到另一个阶段. 因此,

我们现在描述一个最优的和自适应的策略称为上置信界 (UCB) 算法. 它在不确定的情况下运用乐观主义的原则, 在每一个回合中选择有潜力成为最好的一个.

UCB 的工作是保持对每只手臂的预期损失的估计, 以及在一定概率下的置信区间. 粗略地说, 我们得到的概率至少是 $1-\delta$

$$\mu_i \in \left[\hat{\mu}_i - \sqrt{\frac{2\ln\frac{1}{\delta}}{S_{t-1,i}}}, \hat{\mu}_i\right],$$

这里的“大致”来自于 $S_{t-1,i}$ 本身是一个随机变量. 然后,UCB 将查询具有最小下限的臂, 也就是可能具有最小预期损失的臂.

附注 10.13. *置信上限这个名称来自于这样一个事实: 传统上随机强盗是根据回报而不是损失来定义的. 所以, 在我们的例子中, 我们实际上在算法中使用了较低的置信界. 但是, 为了避免与文献混淆, 我们仍然称其为置信上限算法.*

证明中的关键点是如何选择正确的置信水平 δ 和如何解决依赖问题.

该算法总结在算法10.5 中, 我们可以证明以下后悔界.

定理 10.14. *假设臂的奖励是 1-subgaussian 且 $\alpha > 2$ 并设 $\alpha > 2$. 则, UCB 保证一个后悔*

$$\mathrm{Regret}_T \leq \frac{\alpha}{\alpha-2}\sum_{i=1}^{d}\Delta_i + \sum_{i:\Delta_i>0}\frac{8\alpha\ln T}{\Delta_i}\ .$$

证明. 我们当时只分析一只手臂. 同时, 在不失一般性的前提下, 假设最优臂是第一个. 对于臂 i, 我们要证明 $\mathbb{E}[S_{T,i}] \leq \frac{8\alpha\ln T}{\Delta_i^2} + \frac{\alpha}{\alpha-2}$.

这个证明是基于这样一个事实: 一旦我对一个手臂进行了足够多的采样, 取次优手臂的概率很小.

设 $t^\star$ 是最大的时间索引使得 $S_{t^\star-1,i} \leq \frac{8\alpha\ln T}{\Delta_i^2}$. 若 $t^\star = T$, 则上述说法是正确的. 因此, 我们可以安全地假设 $t^\star < T$. 现在, 对于 $t > t^\star$, 有

$$S_{t-1,i} > \frac{8\alpha\ln T}{\Delta_i^2}\ . \tag{10.6}$$

考虑 $t > t^\star$, 使得 $A_t = i$, 则我们可以确定以下两个等式中至少有一个为真:

$$\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha\ln t}{S_{t-1,1}}} \geq \mu_1, \tag{10.7}$$

$$\hat{\mu}_{t-1,i} + \sqrt{\frac{2\alpha\ln t}{S_{t-1,i}}} < \mu_i\ . \tag{10.8}$$

如果第一个为真, 那么我们对最优臂期望的估计的置信区间不包含 μ_1. 另一方面, 如果第二项为真, 则我们对期望 μ_i 的估计不包含 μ_i 的置信区间. 所以, 我们声称如果 $t > t^\star$ 并且我们选择了一个次优的手臂, 那么这两件糟糕的事情中至少有一件发生了.

让我们来证明这个论断: 如果上面的不等式都是错误的, 有 $t > t^\star$, 和 $A_t = i$, 有

$$\begin{aligned}
\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} &< \mu_1 \qquad \text{((10.7) false)} \\
&= \mu_i - \Delta_i \\
&< \mu_i - 2\sqrt{\frac{2\alpha \ln T}{S_{t-1,i}}} \qquad \text{(for (10.6))} \\
&\le \mu_i - 2\sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} \\
&\le \hat{\mu}_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} \qquad \text{((10.8) false)},
\end{aligned}$$

根据算法的选择策略, 这意味着 $A_t \neq i$.

注意 $S_{t^\star,i} \le \frac{8\alpha \ln T}{\Delta_i^2} + 1$. 因此, 有

$$\begin{aligned}
\mathbb{E}\left[S_{T,i}\right] &= \mathbb{E}[S_{t^\star,i}] + \sum_{t=t^\star+1}^{T} \mathbb{E}[\mathbf{1}[A_t = i, \text{(10.7) or (10.8) true}]] \\
&\le \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=t^\star+1}^{T} \mathbb{E}[\mathbf{1}[\text{(10.7) or (10.8) true}]] \\
&\le \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=t^\star+1}^{T} \left(\mathbb{P}[\text{(10.7) true}] + \mathbb{P}[\text{(10.8) true}]\right) .
\end{aligned}$$

现在, 我们来上界求和的概率. 考虑到臂上的损失是身份证和使用联合边界, 我们有

$$\begin{aligned}
\mathbb{P}\left[\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} \ge \mu_1\right] &\le \mathbb{P}\left[\max_{s=1,\ldots,t-1} \frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \ge \mu_1\right] \\
&= \mathbb{P}\left[\bigcup_{s=1}^{t-1}\left\{\frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \ge \mu_1\right\}\right] .
\end{aligned}$$

因此, 有

$$\begin{aligned}
\mathbb{P}[\text{(10.7) true}] &\le \sum_{s=1}^{t-1} \mathbb{P}\left[\frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \ge \mu_1\right] \qquad \text{(union bound)} \\
&\le \sum_{s=1}^{t-1} t^{-\alpha} = (t-1)t^{-\alpha} .
\end{aligned}$$

已知对于 $\mathbb{P}[\text{(10.8) true}]$, 有相同的边界, 有

$$\begin{aligned}
\mathbb{E}\left[S_{T,i}\right] &\le \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty} 2(t-1)t^{-\alpha} \le \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=2}^{\infty} 2t^{1-\alpha} \le \frac{8\alpha \ln T}{\Delta_i^2} + 1 + 2\int_1^\infty x^{1-\alpha} \\
&= \frac{8\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha - 2} .
\end{aligned}$$

利用我们上次证明的后悔分解, $\text{Regret}_T = \sum_{i=1}^{d} \Delta_i \mathbb{E}[S_{T,i}]$, 得到了给定的界. □

如果间隙太小, 上面的边界就没有意义了. 这里我们证明了另一个不依赖于逆的边界.

定理 10.15. *假设臂的报酬减去期望是 1-subgaussian 并设 $\alpha > 2$. 那么,UCB 保证一个后悔*

$$\mathrm{Regret}_T \leq 4\sqrt{2\alpha d T \ln T} + \frac{\alpha}{\alpha - 2}\sum_{i=1}^{d} \Delta_i \ .$$

证明. 设 $\Delta > 0$ 是一些随后要调整的值, 回想一下定理10.14 的证明, 对于每个次优臂 i 我们可以界定

$$\mathbb{E}[S_{T,i}] \leq \frac{\alpha}{\alpha - 2} + \frac{8\alpha \ln T}{\Delta_i^2} \ .$$

因此, 使用我们上次证明的后悔分解, 我们得到了

$$\begin{aligned}
\mathrm{Regret}_T &= \sum_{i:\Delta_i < \Delta} \Delta_i \mathbb{E}[S_{T,i}] + \sum_{i:\Delta_i \geq \Delta} \Delta_i \mathbb{E}[S_{T,i}] \\
&\leq T\Delta + \sum_{i:\Delta_i \geq \Delta} \Delta_i \mathbb{E}[S_{T,i}] \\
&\leq T\Delta + \sum_{i:\Delta_i \geq \Delta} \left(\Delta_i \frac{\alpha}{\alpha - 2} + \frac{8\alpha \ln T}{\Delta_i} \right) \\
&\leq T\Delta + \frac{\alpha}{\alpha - 2}\sum_{i=1}^{d} \Delta_i + \frac{8\alpha d \ln T}{\Delta} \ .
\end{aligned}$$

选择 $\Delta = \sqrt{\frac{8\alpha d \ln T}{T}}$, 我们得到给定的边界. □

附注 10.16. *注意, 虽然 UCB 算法被认为是无参数的, 但我们仍然需要知道臂的亚高斯性. 虽然这很容易成为有界支撑的随机臂的上界, 但在没有任何关于臂分布的先验知识的情况下, 我们不清楚如何做到这一点.*

在以下定理的意义上, 可以证明 UCB 算法是渐近最优的.

定理 10.17 ([19, Theorem 2.2])**.** *考虑一个策略, 满足 $\mathbb{E}[S_{T,i}] = o(T^a)$ 对于任何伯努利奖励分布集, 任何臂 i 具有 $\Delta_i > 0$ 和任意 $a > 0$. 那么, 对于任意一组伯努利奖励分布, 下面的结论成立*

$$\liminf_{T \to +\infty} \frac{\mathrm{Regret}_T}{\ln T} \geq \sum_{i:\Delta_i} \frac{1}{2\Delta_i} \ .$$

10.3 历史片段

算法10.1来自于 Cesa-Bianchi and Lugosi [20, Theorem 6.9]. Auer et al. [8]提出了 Exp3 算法.

INF 算法由 Audibert and Bubeck [5] 提出, 并在 Audibert et al. [6]将其重新转换为 OMD 过程. 与 Tsallis 熵的联系在 Abernethy et al. [3]进行了研究. 这里提出的具体证明是新的, 它建立在 Abernethy et al. [3]的证明之上. 注意, Abernethy et al. [3] 在损失的随机估计 (他们称之为基于梯度的预测算法) 上证明了 FTRL 过程的相同后悔, 而在这里, 我们使用 OMD 过程证明了它.

ETC 算法可以追溯到 Robbins [71], 即使 Robbins 提出了现在被称为 epoch-greedy [50] 的算法. 要了解更多关于 ETC 的历史, 请查看 Lattimore and Szepesvári [51]的第六章. 这里提供的证明也来自 Lattimore and Szepesvári [51].

信心界限和乐观主义的概念最早出现在 Lai and Robbins [49]. 第一个版本的 UCB 是由 Lai [48]提出. 我提出的 UCB 版本是 Auer et al. [7]命名为 UCB1. 注意, 并不考虑 1-subgaussian 环境, Auer et al. [7] 考虑奖励被限制在 $[0,1]$ 的 bandits . 10.14的证明是定理 2.1 在 Bubeck and Cesa-Bianchi [19]中的一个小的变化, 推广到了亚高斯设置, 定理10.15 来自于 Bubeck and Cesa-Bianchi [19].

附录 A

附录

A.1 Lambert 函数及其应用

Lambert 函数 $W(x):\mathbb{R}_+\to\mathbb{R}_+$ 由以下等式定义

$$x = W(x)\exp\left(W(x)\right) \qquad \text{for } x\geq 0\ . \tag{A.1}$$

因此, 从定义中我们得到了 $\exp(W(x)/2)=\sqrt{\frac{x}{W(x)}}$.

定理 A.1 ([41, Theorem 2.3])**.**

$$W(x)\leq \ln\frac{x+C}{1+\ln(C)},\quad \forall x>-\frac{1}{e},\ C>\frac{1}{e}.$$

下面引理给出了 $W(x)$ 的上界和下界.

定理 A.2. *Lambert 函数 $W(x)$ 满足*

$$0.6321\ln(x+1)\leq W(x)\leq \ln(x+1),\ \forall x\geq 0\ .$$

证明. 不等式满足 $x=0$, 因此我们在下面假设 $x>0$. 我们首先证明下界. 从 (A.1) 有

$$W(x)=\ln\left(\frac{x}{W(x)}\right)\ . \tag{A.2}$$

从这个等式出发, 利用初等不等式 $\ln(x)\leq\frac{a}{e}x^{\frac{1}{a}}$ 对于任意 $a>0$, 有

$$W(x)\leq\frac{1}{a\,e}\left(\frac{x}{W(x)}\right)^{a}\quad \forall a>0,$$

即

$$W(x)\leq\left(\frac{1}{a\,e}\right)^{\frac{1}{1+a}}x^{\frac{a}{1+a}}\quad \forall a>0. \tag{A.3}$$

用(A.2)中的 (A.3) , 有

$$W(x) \geq \ln\left(\frac{x}{\left(\frac{1}{a\,e}\right)^{\frac{1}{1+a}} x^{\frac{a}{1+a}}}\right) = \frac{1}{1+a}\ln(a\,e\,x) \quad \forall a > 0\ .$$

现在考虑函数 $g(x) = \frac{x}{x+1} - \frac{b}{\ln(1+b)(b+1)}\ln(x+1)$ 定义在 $[0,b]$ 中, 其中 b 是一个正数, 将在下面决定. 该函数在 $x^\star = (1+\frac{1}{b})\ln(1+b) - 1$ 中有最大值, 在 $[0,x^\star]$ 中导数为正, 在 $[x^\star,b]$ 中为负. 因此, 在 $x=0$ 和 $x=b$ 中, 最小值都是 0. 利用刚才在 g 上证明的性质, 设 $a = \frac{1}{x}$, 有

$$W(x) \geq \frac{x}{x+1} \geq \frac{b}{\ln(1+b)(b+1)}\ln(x+1) \quad \forall x \leq b\ .$$

对于 $x > b$, 设 $a = \frac{x+1}{ex}$, 有

$$W(x) \geq \frac{e\,x}{(e+1)x+1}\ln(x+1) \geq \frac{e\,b}{(e+1)b+1}\ln(x+1) \tag{A.4}$$

因此, 设 b 导致

$$\frac{e\,b}{(e+1)b+1} = \frac{b}{\ln(1+b)(b+1)}$$

数字上, $b = 1.71825...$, 故

$$W(x) \geq 0.6321\ln(x+1)\ .$$

对于上界, 我们用定理A.1 且设 $C = 1$. □

定理 A.3. *设 $a, b > 0$. 则, $f(x) = b\exp(x^2/(2a))$ 的 Fenchel 共轭是*

$$f^\star(\theta) = \sqrt{a}|\theta|\sqrt{W(a\theta^2/b^2)} - b\exp\left(\frac{W(a\theta^2/b^2)}{2}\right) = \sqrt{a}|\theta|\left(\sqrt{W(a\theta^2/b^2)} - \frac{1}{\sqrt{W(a\theta^2/b^2)}}\right)\ .$$

而且,

$$f^\star(\theta) \leq \sqrt{a}|\theta|\sqrt{\ln(a\theta^2/b^2+1)} - b\ .$$

证明. 首先, 发现到

$$\max_x\ \theta x - b\exp(x^2/(2a)) = \max_y\ b\left(\frac{\sqrt{a}\theta}{b}y - \exp(y^2/2)\right)\ .$$

同样, 根据 Lambert 函数的定义, 有

$$\operatorname*{argmax}_y\ uy - \exp(y^2/2) = \operatorname{sign}(u)\sqrt{W(y^2)},$$

对于 $u = 0$, 其中 $\operatorname{sign}(u) = 0$. 因此 $\operatorname{argmax}_x\ \theta x - b\exp(x^2/(2a)) = \operatorname{sign}(\theta)\sqrt{W(a\theta^2/b^2)}$. 故,

$$\begin{aligned}
\max_x\ \theta x - b\exp(x^2/(2a)) &= \max_y\ b\left(\frac{\sqrt{a}\theta}{b}y - \exp(y^2/2)\right) \\
&= \sqrt{a}|\theta|\sqrt{W(a\theta^2/b^2)} - b\exp\left(\frac{W(a\theta^2/b^2)}{2}\right) \\
&= \sqrt{a}|\theta|\sqrt{W(a\theta^2/b^2)} - b\sqrt{\frac{a\theta^2/b^2}{W(a\theta^2/b^2)}} \\
&= \sqrt{a}|\theta|\left(\sqrt{W(a\theta^2/b^2)} - \frac{1}{\sqrt{W(a\theta^2/b^2)}}\right)\ .
\end{aligned}$$

□

通过定理A.2得到不等式.

参考文献

[1] J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory (COLT)*, pages 807–823, 2014. URL `http://proceedings.mlr.press/v35/abernethy14`. 79

[2] J. D. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In Rocco A. Servedio and Tong Zhang, editors, *Proc. of Conference on Learning Theory (COLT)*, pages 263–274. Omnipress, 2008. URL `https://repository.upenn.edu/cgi/viewcontent.cgi?article=1492&context=statistics_papers`. 54, 79

[3] J. D. Abernethy, C. Lee, and A. Tewari. Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems 28*, pages 2197–2205. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/6030-fighting-bandits-with-a-new-kind-of-smoothness.pdf`. 115

[4] R. Arora, O. Dekel, and A. Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proc. of the International Conference on Machine Learning (ICML)*, January 2012. URL `https://www.microsoft.com/en-us/research/publication/online-bandit-learning-adaptive-adversary-regret-policy-regret/`. 102

[5] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proc. of the Conference on Learning Theory (COLT)*, 2009. URL `https://www.di.ens.fr/willow/pdfscurrent/COLT09a.pdf`. 115

[6] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 107–132, 2011. URL `http://proceedings.mlr.press/v19/audibert11a.html`. 115

[7] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002. URL `https://homes.di.unimi.it/cesa-bianchi/Pubblicazioni/ml-02.pdf`. 116

[8] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit

problem. *SIAM journal on computing*, 32(1):48–77, 2002. URL http://rob.schapire.net/papers/AuerCeFrSc01.pdf. 115

[9] P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *J. Comput. Syst. Sci.*, 64(1):48–75, 2002. URL http://homes.dsi.unimi.it/~cesabian/Pubblicazioni/jcss-02.pdf. 32

[10] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001. URL https://link.springer.com/article/10.1023/A:1010896012157. 80

[11] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011. URL https://link.springer.com/book/10.1007/978-1-4419-9467-7. 13, 34, 35, 42, 48

[12] H. H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017. URL https://people.ok.ubc.ca/bauschke/Research/103.pdf. 80

[13] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. URL https://web.iem.technion.ac.il/images/user-files/becka/papers/3.pdf. 54

[14] A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM Journal on Optimization*, 12(1):79–108, 2001. URL https://epubs.siam.org/doi/pdf/10.1137/S1052623499354564. 54

[15] A. Beygelzimer, F. Orabona, and C. Zhang. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 488–497. JMLR. org, 2017. URL https://arxiv.org/abs/1702.07958. 85

[16] B. Birnbaum, N. R. Devanur, and L. Xiao. Distributed algorithms via gradient descent for fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 127–136, 2011. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/prop_response.pdf. 80

[17] A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *COLT*, volume 99, pages 203–208, 1999. URL https://www.ri.cmu.edu/pub_files/pub1/blum_a_1999_1/blum_a_1999_1.pdf. 22

[18] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. URL https://web.stanford.edu/~boyd/cvxbook/. 7

[19] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. URL `http://sbubeck.com/SurveyBCB12.pdf`. 115, 116

[20] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006. URL `https://www.cambridge.org/us/academic/subjects/computer-science/pattern-recognition-and-machine-learning/prediction-learning-and-games?format=HB&isbn=9780521841085`. 36, 54, 72, 75, 87, 115

[21] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theory*, 50(9):2050–2057, 2004. URL `https://homes.di.unimi.it/~cesabian/Pubblicazioni/J20.pdf`. 22, 23

[22] K. Chaudhuri, Y. Freund, and D. J. Hsu. A parameter-free hedging algorithm. In *Advances in neural information processing systems*, pages 297–305, 2009. URL `https://arxiv.org/abs/0903.2851`. 100, 101

[23] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993. URL `https://epubs.siam.org/doi/pdf/10.1137/0803026`. 42

[24] A. Chernov and V. Vovk. Prediction with advice of unknown number of experts. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010. URL `https://arxiv.org/abs/1408.2040`. 101

[25] C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Proc. of the Conference on Learning Theory (COLT)*, volume 23, pages 6.1–6.20, 2012. URL `http://proceedings.mlr.press/v23/chiang12.html`. 80

[26] F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, New York, NY, USA, 2007. 84

[27] A. Cutkosky. *Algorithms and Lower Bounds for Parameter-free Online Learning*. PhD thesis, Stanford University, 2018. URL `https://www-cs.stanford.edu/people/ashokc/papers/thesis.pdf`. 38

[28] A. Cutkosky. Anytime online-to-batch, optimism and acceleration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proc. of the 36th International Conference on Machine Learning*, volume 97 of *Proc. of Machine Learning Research*, pages 1446–1454, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/cutkosky19a/cutkosky19a.pdf`. 23

[29] A. Cutkosky. Combining online learning guarantees. In *Proc. of the Conference on Learning Theory (COLT)*, 2019. URL `https://arxiv.org/abs/1902.09003`. 101

[30] A. Cutkosky and F. Orabona. Black-box reductions for parameter-free online learning in Banach spaces. In *Proc. of the Conference on Learning Theory (COLT)*, 2018. URL https://arxiv.org/abs/1802.06293. 101

[31] S. de Rooij, T. van Erven, P. D. Grünwald, and W. M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15(37):1281–1316, 2014. URL http://jmlr.org/papers/v15/rooij14a.html. 80

[32] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, 2010. URL https://stanford.edu/~jduchi/projects/DuchiHaSi10_colt.pdf. 30, 32

[33] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. URL https://www.sciencedirect.com/science/article/pii/S002200009791504X. 54

[34] Y. Freund and R. E. Schapire. Large margin classification using the Perceptron algorithm. *Machine Learning*, pages 277–296, 1999. URL https://cseweb.ucsd.edu/~yfreund/papers/LargeMarginsUsingPerceptron.pdf. 85

[35] C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003. URL https://link.springer.com/article/10.1023/A:1026319107706. 85

[36] C. Gentile and N. Littlestone. The robustness of the p-norm algorithms. In *Proc. of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 1–11, New York, NY, USA, 1999. ACM. URL http://doi.acm.org/10.1145/307400.307405. 54

[37] G. J. Gordon. Regret bounds for prediction problems. In *Proc. of the twelfth annual conference on Computational learning theory (COLT)*, pages 29–40, 1999. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.4767&rep=rep1&type=pdf. 15

[38] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. In *Proc. of the 10th Annual Conference on Computational Learning Theory*, pages 171–183, 1997. 54

[39] E. Hazan, A. Kalai, S. Kale, and A. Agarwal. Logarithmic regret algorithms for online convex optimization. In *International Conference on Computational Learning Theory*, pages 499–513. Springer, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.3483&rep=rep1&type=pdf. 31, 80

[40] E. Hazan, A. Rakhlin, and P. L. Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pages 65–72, 2008. URL https://papers.nips.cc/paper/3319-adaptive-online-gradient-descent.pdf. 31

[41] A. Hoorfar and M. Hassani. Inequalities on the lambert w function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008. URL `http://emis.ams.org/journals/JIPAM/images/107_07_JIPAM/107_07_www.pdf`. 117

[42] P. Joulani, A. György, and C. Szepesvári. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds. In *Proc. of the International Conference on Algorithmic Learning Theory (ALT)*, volume 76, pages 681–720, 2017. URL `http://proceedings.mlr.press/v76/joulani17a.html`. 80

[43] J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, January 1997. URL `https://users.soe.ucsc.edu/~manfred/pubs/J36.pdf`. 54

[44] J. Kivinen and M. K. Warmuth. Averaging expert predictions. In *European Conference on Computational Learning Theory*, pages 153–167. Springer, 1999. URL `https://users.soe.ucsc.edu/~manfred/pubs/C50.pdf`. 80

[45] W. M. Koolen and T. van Erven. Second-order quantile methods for experts and combinatorial games. In *Proc. of the Conference On Learning Theory (COLT)*, pages 1155–1175, 2015. URL `https://arxiv.org/abs/1502.08009`. 101

[46] R. Krichevsky and V. Trofimov. The performance of universal encoding. *IEEE Trans. on Information Theory*, 27(2):199–207, 1981. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1056331`. 101

[47] S. Lacoste-Julien, M. Schmidt, and F. Bach. A simpler approach to obtaining an O(1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012. URL `https://arxiv.org/abs/1212.2002`. 22

[48] T. L. Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, pages 1091–1114, 1987. URL `https://projecteuclid.org/euclid.aos/1176350495`. 116

[49] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985. URL `https://core.ac.uk/download/pdf/82425825.pdf`. 116

[50] J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008. URL `https://papers.nips.cc/paper/3178-the-epoch-greedy-algorithm-for-multi-armed-bandits-with-side-information`. 116

[51] T. Lattimore and C. Szepesvári. Bandit algorithms. *preprint*, 2018. URL `https://tor-lattimore.com/downloads/book/book.pdf`. 110, 116

[52] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994. URL https://www.sciencedirect.com/science/article/pii/S0890540184710091. 54

[53] H. Lu, R. M. Freund, and Y. Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018. URL https://arxiv.org/abs/1610.05708. 80

[54] B. McMahan and J. Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2724–2732. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5148-minimax-optimal-algorithms-for-unconstrained-linear-optimization.pdf. 38

[55] H. B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *Proc. of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 525–533, 2011. URL http://proceedings.mlr.press/v15/mcmahan11b/mcmahan11b.pdf. 80

[56] H B. McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017. URL http://www.jmlr.org/papers/volume18/14-428/14-428.pdf. 79, 80

[57] H. B. McMahan and F. Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *Proc of the Annual Conference on Learning Theory, COLT*, 2014. URL https://arxiv.org/abs/1403.0628. 101

[58] H. B. McMahan and M. J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, 2010. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/36483.pdf. 30, 32

[59] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica. Ad click prediction: a view from the trenches. In *Proc. of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41159.pdf. 80

[60] A. S. Nemirovsky and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, New York, NY, USA, 1983. URL https://books.google.com/books/about/Problem_Complexity_and_Method_Efficiency.html?id=6ULvAAAAMAAJ. 54

[61] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009. URL https://link.springer.com/content/pdf/10.1007/s10107-007-0149-x.pdf. 79

[62] Y. Nesterov and V. Shikhman. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3):917–940, 2015. URL https://link.springer.com/article/10.1007/s10957-014-0677-5. 23

[63] F. Orabona. Dimension-free exponentiated gradient. In *Advances in Neural Information Processing Systems 26*, pages 1806–1814. Curran Associates, Inc., 2013. URL https://papers.nips.cc/paper/4920-dimension-free-exponentiated-gradient.pdf. 38, 101

[64] F. Orabona and D. Pál. Scale-free algorithms for online linear optimization. In *International Conference on Algorithmic Learning Theory*, pages 287–301. Springer, 2015. URL https://arxiv.org/abs/1502.05744. 80

[65] F. Orabona and D. Pal. Coin betting and parameter-free online learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 577–585. Curran Associates, Inc., 2016. URL https://arxiv.org/pdf/1602.04128.pdf. 37, 88, 101

[66] F. Orabona and D. Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018. URL https://arxiv.org/pdf/1601.01974.pdf. Special Issue on ALT 2015. 32, 37

[67] F. Orabona and T. Tommasi. Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems*, pages 2160–2170, 2017. URL https://arxiv.org/abs/1705.07795. 101

[68] F. Orabona, K. Crammer, and N. Cesa-Bianchi. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99:411–435, 2015. URL https://arxiv.org/abs/1304.2994. 79, 80

[69] A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Proc. of the Conference on Learning Theory (COLT)*, volume 30, pages 993–1019, 2013. URL http://proceedings.mlr.press/v30/Rakhlin13.html. 80

[70] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=ryQu7f-RZ. 7

[71] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952. URL https://projecteuclid.org/euclid.bams/1183517370. 116

[72] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. URL https://press.princeton.edu/titles/1815.html. 8, 13, 35

[73] F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf. 85

[74] S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University, 2007. URL https://www.cs.huji.ac.il/~shais/papers/ShalevThesis07.pdf. 24, 50, 52, 79

[75] S. Shalev-Shwartz and Y. Singer. Online learning meets optimization in the dual. In *International Conference on Computational Learning Theory*, pages 423–437. Springer, 2006. URL https://storage.googleapis.com/pub-tools-public-publication-data/pdf/25.pdf. 79

[76] S. Shalev-Shwartz and Y. Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical report, The Hebrew University, 2007. URL https://www.cse.huji.ac.il/~shais/papers/ShalevSi07report.pdf. 80

[77] S. Shalev-Shwartz and Y. Singer. Convex repeated games and Fenchel duality. In *Advances in neural information processing systems*, pages 1265–1272, 2007. URL https://ttic.uchicago.edu/~shai/papers/ShalevSi06_fench.pdf. 79

[78] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. of ICML*, pages 807–814, 2007. URL https://ttic.uchicago.edu/~nati/Publications/Pegasos.pdf. 22

[79] N. Srebro, K. Sridharan, and A. Tewari. Optimistic rates for learning with a smooth loss. *arXiv preprint arXiv:1009.3896*, 2010. URL https://arxiv.org/abs/1009.3896. 29

[80] J. Steinhardt and P. Liang. Adaptivity and optimism: An improved exponentiated gradient algorithm. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 1593–1601, 2014. URL https://cs.stanford.edu/~pliang/papers/eg-icml2014.pdf. 80

[81] M. Streeter and B. McMahan. No-regret algorithms for unconstrained online convex optimization. In *Advances in Neural Information Processing Systems 25*, pages 2402–2410. Curran Associates, Inc., 2012. URL https://arxiv.org/pdf/1211.2260.pdf. 38, 101

[82] M. Streeter and H. B. McMahan. Less regret via online conditioning, 2010. URL https://arxiv.org/abs/1002.4862. arXiv:1002.4862. 32

[83] T. van Erven, W. M. Koolen, S. D. Rooij, and P. Grünwald. Adaptive Hedge. In *Advances in Neural Information Processing Systems*, pages 1656–1664, 2011. URL https://papers.nips.cc/paper/4191-adaptive-hedge.pdf. 80

[84] V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2001.tb00457.x. 80

[85] V. G. Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, pages 371–386, 1990. 54

[86] M. K. Warmuth and A. K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *Electronic proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, volume 326, 1997. URL https://users.soe.ucsc.edu/~manfred/pubs/C45.pdf. 54

[87] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/xiao10JMLR.pdf. 79, 80

[88] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proc. of International Conference on Machine learning*, pages 919–926, New York, NY, USA, 2004. ACM. URL http://tongzhang-ml.org/papers/icml04-stograd.pdf. 22

[89] J. Zimmert and Y. Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *arXiv preprint arXiv:1807.07623*, 2018. URL https://arxiv.org/abs/1807.07623. 80

[90] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of ICML*, pages 928–936, 2003. URL https://www.aaai.org/Papers/ICML/2003/ICML03-120.pdf. 15

[91] M. Zinkevich. *Theoretical guarantees for algorithms in multi-agent settings*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2004. URL http://martin.zinkevich.org/publications/thesis.ps. 80