1

Introduction and Problem Statement

This book is about approximately solving problems of the form

$$\arg\min_{x \in K} f(x)\,, \tag{1.1}$$

where K is a convex body in $\mathbb{R}^d$ (convex, compact and nonempty interior) and $f\colon K \to \mathbb{R}$ is a convex function. Problems of this kind are ubiquitous in machine learning, operations research, economics and beyond. The difficulty of this problem depends on many factors; for example, the dimension and smoothness properties of f. Most important, however, is the representation of the function f and constraint set K. Our focus is on zeroth-order stochastic optimisation, where you can query f at any $x \in K$ and observe $y = f(x) + \text{noise}$. By contrast, the vast majority of the literature on mathematical programming allows both the value of f and its derivatives to be computed at any $x \in K$, either exactly or with additive noise. This is not a book about applications, but for the sake of inspiration and motivation we list a few situations where zeroth-order optimisation is a natural fit.

- *Real-world experiments:* A chef wants to optimise the temperature and baking time when baking a soufflé. The constraint set K is some reasonable subset of the possible temperature/time pairings and $f(x)$ is the expected negative quality of the finished product. Noise arises here from exogenous factors, such as unintentional variation in recipe preparation.
- *Adversarial attacks in machine learning:* A company releases an image recognition system. Can you find an image that looks to the human eye like a stop sign but is classified by the system as something else? Unless the company has released the code and weights, you can only interact with a black-box function C that accepts images as input and returns a classification, possibly at some cost. A simple idea is to take an image $x_\circ$ and let K be a set of images that are visually indistinguishable from $x_\circ$. Then approximately

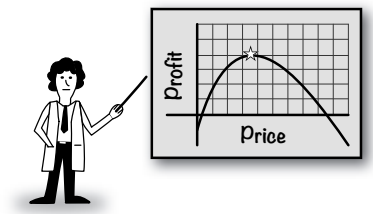
https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

solve the following optimisation problem:

$$\arg\min_{x \in K} S(C(x), C(x_\circ)) ,$$

where $S(C(x), C(y))$ is some measure of similarity between the classifications $C(x)$ and $C(y)$. That is, the problem is to find an image x that looks similar to $x_\circ$ but with $C(x)$ maximally different from $C(x_\circ)$.

- *Reinforcement learning and control:* There are many ways to do reinforcement learning. Suppose that π_θ is a policy parameterised by $\theta \in K$ and $f(\theta)$ is the (expected) loss when implementing policy π_θ. Then solving (1.1) corresponds to finding the optimal policy in $\{\pi_\theta : \theta \in K\}$.
- *Hyperparameter tuning:* Most machine learning systems have a range of parameters; for example, the width, depth and structure of a neural network or learning rates (schedules) for the training algorithm. You may want to automate the process of finding the best architecture and training parameters. The loss in this case could be the performance of the resulting system after training. So computing $f(x)$ requires running an entire training process, which is enormously expensive and possibly random. Generally speaking f cannot be differentiated.
- *Dynamic pricing:* In dynamic pricing a retailer interacts sequentially with an environment. Customers arrive in the system and the retailer suggests a price $X_t \in K \subset \mathbb{R}$. The loss $f(X_t)$ is the (expected) negative profit. When the price is too high the customer will not purchase and some loss is incurred (operating costs). On the other hand, when the price is too low there is also a loss. The function f is not known *a priori* to the retailer and variability in customers introduces noise in the observations.



The last application reveals another consideration. Approximately minimising the loss f as in (1.1) is not the only possible objective. In dynamic pricing every query to the loss function (decision) entails an actual cost (or profit). In this situation a more natural objective is to minimise the cumulative loss over all interactions. Most of this book is about this objective, which is most often seen in the literature on multi-armed bandits. As we shall soon see, an algorithm that approximately minimises the cumulative loss can be used to solve the optimisation problem (1.1), but the reverse does not hold. Note that many practical problems are not convex. Nevertheless, enormous

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

practical experience with first-order methods suggests this is less of a problem than it seems. The same may be true in the zeroth-order setting.

Many of the proposed applications in this book are aspirational rather than truly practical. Automatic hyperparameter tuning is used everywhere, but the algorithms employed are usually based on other methods, such as Gaussian process bandits. Presumably the reason for this is because convex bandit algorithms are not quite ready for the big stage yet. The simple methods converge slowly, while the complicated methods are challenging to implement and tune. Hopefully in the coming years new and better algorithms will be developed.

1.1 Methods and Challenges

In spite of its simplicity, bandit (and zeroth-order) convex optimisation is still not fully understood; and certainly much less so than first-order optimisation where the gradients of f are available. Nearly every method for bandit convex optimisation is based on estimating the gradient (and maybe higher-order derivatives) of a surrogate loss and combining this with a classical gradient-based optimisation method. This approach leads to a pernicious bias/variance trade-off, which is particularly exacerbated when the loss function is non-smooth. Unsurprisingly, then, the primary focus of the classical literature on stochastic zeroth-order optimisation is on smooth settings.

This book explores the full range of methods, including the most classical online projected gradient descent (Chapter 5), cutting plane methods (Chapter 9) and more exotic methods based on information theory and minimax duality (Chapter 8). Buried somewhere in nearly every approach is some kind of bias/variance trade-off. Optimising for the regret also introduces complications. The learner cannot freely query the loss at any point they might like without consideration for the loss suffered. This is known as the exploration/exploitation dilemma for which reinforcement learning is famous. We will see, in fact, that in certain settings optimising for the regret is (provably) harder than merely finding a near-optimiser of the loss. Another theme that raises its head in many chapters is the need to balance statistical and computational efficiency, both of the theoretical kind (poly time or not) and the practical kind. At present, those algorithms with the best statistical guarantees do not have known efficient implementations.

So, let us begin. The next section explains what prerequisites you may find useful. After that, the rest of the chapter is devoted to the formal problem settings studied later. Chapter 2 covers the history and includes large reference tables of

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

known results. Chapter 3 covers the necessary mathematical tools. Nearly every chapter thereafter is based on what can be achieved with some algorithmic idea.

1.2 Prerequisites

Most readers will benefit from a reasonable knowledge of online learning (Cesa-Bianchi and Lugosi, 2006; Hazan, 2016; Orabona, 2019) and bandits (Bubeck and Cesa-Bianchi, 2012; Slivkins, 2019; Lattimore and Szepesvári, 2020). We use some theory from interior point methods, which you could refresh by reading the lecture notes by Nemirovski (1996). None of this is essential, however, if you are prepared to take a few results on faith. Similarly, we use a few simple results from concentration of measure. Our reference was the book by Vershynin (2018) but Boucheron et al. (2013) also covers the material needed. We do use martingale versions of these results, which sadly do not appear in these books but are more or less trivial extensions. The standard reference for convex analysis is the book by Rockafellar (1970) to which we occasionally refer. No deep results are needed, however. Similarly, we make use of certain elementary results in convex geometry. Our reference for these is mostly the book by Artstein-Avidan et al. (2015). The symbol (🐦) on a proof, section or chapter means that you could (should?) skip this part on your first pass. The book contains a number of exercises, which are assigned a difficulty based on a star rating. Problems with one ★ are straightforward, requiring limited ingenuity or mathematical sophistication. Problems with two ★★ are moderately difficult and may require considerable ingenuity. Problems with three ★★★ will probably take several days of work. Problems marked with 🖹 may require extensive literature searches and problems marked with **?** have not been solved yet. All difficulty assessments are subjective estimates only.

1.3 Bandit Convex Optimisation

Let $K \subset \mathbb{R}^d$ be convex and $f_1, \ldots, f_n \colon K \to \mathbb{R}$ be an unknown sequence of convex functions. A learner interacts with the environment over n rounds. In round t the learner chooses an action $X_t \in K$. They then observe a noisy loss $Y_t = f_t(X_t) + \varepsilon_t$, where $(\varepsilon_t)_{t=1}^n$ is a sequence of noise random variables. The precise conditions on the noise are given in (1.2) below, but for now you could think of the noise as a sequence of independent standard Gaussian random variables. The learner's decision X_t is allowed to depend on an exogenous source of randomness and the data observed already, which is $X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}$.

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

The main performance metric in this book is the regret, which is

$$\mathrm{Reg}_n = \sup_{x \in K} \sum_{t=1}^{n} (f_t(X_t) - f_t(x)) \,.$$

The regret is a random variable with the randomness coming from both the noise and the learner's decisions. Normally the regret will be bounded in expectation or with high probability, depending on what is most convenient. Of course, the regret also depends on the loss functions. In general we will argue that our algorithms have small regret for any convex losses within some class. Stronger assumptions (smaller classes) lead to stronger results and/or simpler and/or more efficient algorithms. The following definition and notation are sufficient for our purposes.

Definition 1.1 Let $\mathscr{F}$ be the space of convex functions from K to $\mathbb{R}$, and with $\|\cdot\|$ the standard euclidean norm define the following properties of a function $f \in \mathscr{F}$:

`Prop (b)` f is bounded: $f(x) \in [0, 1]$ for all $x \in K$.

`Prop (l)` f is Lipschitz: $f(x) - f(y) \leq \|x - y\|$ for all $x, y \in K$.

`Prop (sm)` f is β-smooth: $f(x) - \frac{\beta}{2}\|x\|^2$ is concave on K.

`Prop (sc)` f is α-strongly convex: $f(x) - \frac{\alpha}{2}\|x\|^2$ is convex on K.

`Prop (lin)` f is linear: $f(x) = x^\top b + c$.

`Prop (quad)` f is quadratic: $f(x) = x^\top A x + x^\top b + c$.

We use the property symbols to define subsets of $\mathscr{F}$. For example:

- $\mathscr{F}_{\mathsf{b}} = \{f \in \mathscr{F} : f(x) \in [0, 1] \text{ for all } x \in K\}$.
- $\mathscr{F}_{\mathsf{b,sm,sc}}$ is the set of bounded convex functions that are smooth and strongly convex.

When smoothness and strong convexity are involved, our bounds will depend on the parameters $\alpha > 0$ and $0 \leq \beta < \infty$, which we assume are known constants.

1.3.1 Constraint Set

The set K is called the constraint set and its geometry also plays a role in the hardness of bandit convex optimisation. We make the following assumption throughout the entire book:

Assumption 1.2 The constraint set $K \subset \mathbb{R}^d$ is a convex body, which means that:

`(1)` K is convex;

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

(2) K has a nonempty interior; and
(3) K is compact.

Only convexity and the boundedness part of compactness are really essential for most results. Every nonempty convex set has a nonempty interior when embedded in a suitable affine subset of $\mathbb{R}^d$. Properties of K that influence the regret of various algorithms include its diameter and how well-rounded it is (Section 3.6). Mathematically speaking, nothing more is needed from the constraint set. Computationally the representation of K is very important. Standard options are as a polytope, the convex hull of a point cloud or as given by a separation/membership oracle. Matters involving computation and the constraint set are discussed in Section 3.9.

Remark 1.3 You may be interested to know what happens when K is not bounded; for example, when $K = \mathbb{R}^d$. We discuss this and other possible assumptions on the constraint set in Note 1.ii at the end of the chapter.

1.3.2 Noise

Our assumption on the noise is that the sequence $(\varepsilon_t)_{t=1}^n$ is conditionally subgaussian. By this we mean:

Assumption 1.4 The noise random variables $\varepsilon_1, \ldots, \varepsilon_n$ are conditionally subgaussian:

$$\mathbb{E}\left[\varepsilon_t \mid X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, X_t\right] = 0\,; \text{ and}$$
$$\mathbb{E}\left[\exp(\varepsilon_t^2) \mid X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, X_t\right] \leq 2\,. \tag{1.2}$$

Assumption 1.4 is considered global and will not be referred to subsequently. Note that (1.2) together with the fact that $x^2 \leq \exp(x^2) - 1$ for all x guarantees that the conditional variance of ε_t satisfies

$$\mathbb{E}[\varepsilon_t^2 | X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, X_t] \leq 1\,. \tag{1.3}$$

This definition of subgaussianity is based on the Orlicz norm definitions. We give a brief summary in Appendix B, or you can read the wonderful book by Vershynin (2018). Sometimes we work in the noise-free setting where $\varepsilon_t = 0$. The assumption of subgaussian noise is rather standard in the bandit literature. The Gaussian distribution $\mathcal{N}(0, 3/8)$ is subgaussian, and so is any distribution that is suitably bounded almost surely. All that is really needed is a suitable concentration of measure phenomenon, and hence all results in this book could be generalised to considerably larger classes of noise distribution without too much effort. We keep things simple, however.

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

1.3.3 Adversarial and Stochastic Convex Bandits

We have already outlined some of the assumptions on the function class to which the losses belong. The other major classification is whether or not the problem is adversarial or stochastic.

Adversarial Bandit Convex Optimisation

In the adversarial setting the most common assumption is that the noise $\varepsilon_t = 0$ while the functions $f_1, \ldots, f_n$ are chosen in an arbitrary way by the adversary. Sometimes the adversary is allowed to choose f_t at the same time as the learner chooses X_t, in which case we say the adversary is non-oblivious. Perhaps more commonly, however, the adversary is obliged to choose all loss functions $f_1, \ldots, f_n$ before the interaction starts. Adversaries of this kind are called oblivious. For our purposes it is convenient to allow nonzero noise even in the adversarial case. This is sometimes essential in applications; for example, in bandit submodular minimisation (Chapter 13). And besides, it feels natural that the adversarial setting should generalise the stochastic one.

Remark 1.5 In the adversarial setting you might be tempted to combine the noise and losses by defining the loss function to be $f_t + \varepsilon_t$. But this would prevent the noise distribution from depending on the action of the learner, which is permitted by Assumption 1.4 and is essential in certain applications, such as submodular bandits (Chapter 13).

Stochastic Bandit Convex Optimisation

The stochastic setting is more classical. The loss function is now constant over time: $f_t = f$ for all rounds t and unknown f. The standard performance metric in bandit problems is the regret, but in the stochastic setting it also makes sense to consider the simple regret. At the end of the interaction the learner is expected to output one last point $\widehat{X} \in K$ and the simple regret is

$$\mathrm{sReg}_n = f(\widehat{X}) - \inf_{x \in K} f(x) \,.$$

Thanks to convexity, there is a straightforward reduction from cumulative regret to simple regret: simply let $\widehat{X} = \frac{1}{n} \sum_{t=1}^{n} X_t$. Then by convexity,

$$\mathrm{sReg}_n \leq \frac{1}{n} \mathrm{Reg}_n \,. \tag{1.4}$$

Another standard measure of performance in the stochastic setting is the sample complexity, which is the number of interactions needed before the simple regret is at most $\varepsilon > 0$ with high probability. The following fact provides a conversion from a high probability bound on the cumulative regret to sample complexity:

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

Fact 1.6 Suppose that with probability at least $1-\delta$ the regret of some algorithm is bounded by $\mathrm{Reg}_n \le R(n)$. Then, by the above conversion, the sample complexity can be bounded by the smallest n such that $R(n)/n \le \varepsilon$. Concretely, if $R(n) = An^p$ for $p \in (0,1)$ and $A > 0$, then the sample complexity is at most $1 + (A/\varepsilon)^{1/(1-p)}$.

There is also a simple reduction from a bound on the expected regret to a (high probability) sample complexity bound. The idea is to run the regret-minimising algorithm $k_{\max} = O(\log(1/\delta))$ times with horizon n large enough that $\mathbb{E}[\frac{1}{n}\mathrm{Reg}_n] \le \frac{\varepsilon}{4}$. By Markov's inequality and (1.4), with constant probability the average iterate over any run is nearly optimal, which means that one of the $k_{\max}$ average iterates is nearly optimal with high probability. Lastly, a best arm identification procedure is applied to select a near minimiser among the $k_{\max}$ candidates. The full procedure and its analysis follow.

```
args: base algorithm ALG, R(·), ε > 0, δ ∈ (0,1)
let n = min{s: R(s)/s ≤ ε/4}
for k = 1 to k_max ≜ ⌈log(2/δ)/log(2)⌉:
    run ALG over n rounds
    observe iterates X_1, ..., X_n
    let x̂_k = (1/n) Σ_{s=1}^{n} X_s
return BAI(ε/2, δ/2, x̂_1, ..., x̂_{k_max})        # Algorithm 9.2
```

Algorithm 1.1 A master algorithm for obtaining high probability sample complexity bounds from a base algorithm with expected regret. See Section 9.3 for a detailed explanation of the BAI subroutine (Algorithm 9.2).

Proposition 1.7 *Suppose that* $\mathbb{E}[\mathrm{Reg}_n] \le R(n)$. *Then*

(1) *Algorithm 1.1 queries the loss at most*

$$O\left(\left(\min\left\{n\colon \frac{R(n)}{n} \le \frac{\varepsilon}{4}\right\} + \frac{\log(1/\delta)}{\varepsilon^2}\right)\log(1/\delta)\right)$$

times; and

(2) *With probability at least* $1-\delta$, *Algorithm 1.1 returns an* $\widehat{X} \in K$ *such that*

$$f(\widehat{X}) \le \inf_{x\in K} f(x) + \varepsilon\,.$$

Proof Let $f_\star = \inf_{x\in K} f(x)$. By (1.4), $\mathbb{E}[f(\widehat{x}_k)] - f_\star \le \frac{\varepsilon}{4}$. Since $f(x) - f_\star \ge 0$

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

for all $x \in K$, by Markov's inequality,

$$\mathbb{P}\left(f(\widehat{x}_k) - f_\star \geq \frac{\varepsilon}{2}\right) \leq \frac{1}{2}\,.$$

Hence, by the definition of $k_{\max}$, with probability at least $1 - \delta/2$ there exists a $k \in \{1, \ldots, k_{\max}\}$ such that

$$f(\widehat{x}_k) - f_\star \leq \frac{\varepsilon}{2}\,.$$

Finally, referring to Section 9.3, by Theorem 9.10 the call to Algorithm 9.2 uses

$$O\left(\frac{k_{\max}}{\varepsilon^2} \log\left(\frac{k_{\max}}{\delta}\right)\right)$$

queries to the loss function and with probability at least $1 - \delta/2$ returns an $\widehat{X} \in \{\widehat{x}_1, \ldots, \widehat{x}_{k_{\max}}\}$ with

$$f(\widehat{X}) \leq \min_{1 \leq k \leq k_{\max}} f(\widehat{x}_k) + \frac{\varepsilon}{2}\,.$$

Combining everything with a union bound completes the proof. □

Our focus for the remainder of the book is primarily on the cumulative regret, but we occasionally highlight the sample complexity of algorithms in order to compare to the literature. The arguments above show that bounds on the cumulative regret imply bounds on the simple regret and sample complexity. The converse is not true.

Regret is Random

You should note that Reg_n and sReg_n are random variables with the randomness arising from both the algorithm and the noise. Most of our results either control $\mathbb{E}[\mathrm{Reg}_n]$ or prove that Reg_n is bounded by such-and-such with high probability. Bounds that hold with high probability are generally preferred since they can be integrated to obtain bounds in expectation. But we will not be too dogmatic about this. Indeed, we mostly prove bounds in expectation to avoid tedious concentration of measure calculations. As far as we know, these always work out if you try hard enough.

1.4 Notation

A full list of notation is available in Appendix C.

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

Norms

The norm $\|\cdot\|$ is the euclidean norm for vectors and the spectral norm for matrices. For positive definite A, $\langle x, y\rangle_A = x^\top A y$ and $\|x\|_A^2 = \langle x, x\rangle_A$. Given a random variable X, $\|X\|_{\psi_k} = \inf\{t > 0 \colon \mathbb{E}[\exp(|X/t|^k)] \leq 2\}$ for $k \in \{1, 2\}$ are the Orlicz norms. Remember, X is subgaussian if $\|X\|_{\psi_2} < \infty$ and subexponential if $\|X\|_{\psi_1} < \infty$. You can read more about the Orlicz norms in Appendix B.

Sets

$\mathbb{R}$ and $\mathbb{Z}$ are the sets of real values and integers. The restriction of the reals to the (strictly) positive real line are $\mathbb{R}_+ = [0, \infty)$ and $\mathbb{R}_{++} = (0, \infty)$. The euclidean ball and sphere of radius r are denoted by $\mathbb{B}_r^d = \{x \in \mathbb{R}^d \colon \|x\| \leq r\}$ and $\mathbb{S}_r^{d-1} = \{x \in \mathbb{R}^d \colon \|x\| = r\}$. Hopefully the latter will not be confused with the space of positive (semi-)definite matrices on $\mathbb{R}^d$, which we denote by $(\mathbb{S}_+^d)\ \mathbb{S}_{++}^d$. Given $x \in \mathbb{R}^d$ and $\mathbf{0} \neq \eta \in \mathbb{R}^d$ we let $H(x, \eta) = \{y \colon \langle y - x, \eta\rangle \leq 0\}$, which is a closed half-space passing through x with outwards-facing normal η. Given $x \in \mathbb{R}^d$ and positive definite matrix A, $E(x, A) = \{y \in \mathbb{R}^d \colon \|x - y\|_{A^{-1}} \leq 1\}$, which is an ellipsoid centred at x. When $E = E(x, A)$ is an ellipsoid, $E(r) = E(x, \sqrt{r}A)$ denotes the same ellipsoid scaled by a factor of r. The space of probability measures on $K \subset \mathbb{R}^d$ is $\Delta(K)$ where we always take the Borel σ-algebra $\mathscr{B}(K)$. Given a natural number m, $\Delta_m = \{p \in \mathbb{R}_+^m \colon \|p\|_1 = 1\}$ and $\Delta_m^+ = \Delta_m \cap \mathbb{R}_{++}^m$. For $x, y \in \mathbb{R}^d$ we let $[x, y] = \{(1 - \lambda)x + \lambda y \colon \lambda \in [0, 1]\}$, which is the chord connecting x and y.

Elementary Notation

We define arithmetic operations on sets in the Minkowski fashion. Concretely, given $A, B \subset \mathbb{R}^d$ the Minkowski sum is $A + B = \{x + y \colon x \in A, y \in B\}$. When $u \in \mathbb{R}$ we let $uA = \{ux \colon x \in A\}$ and of course $-A = (-1)A$ and $A - A = A + (-A)$. Occasionally for $x \in \mathbb{R}^d$ we abbreviate $\{x\} + A = x + A$. The closure of A is $\mathrm{cl}(A)$, its interior is $\mathrm{int}(A) = \{x \in A \colon \exists \epsilon > 0, x + \mathbb{B}_\varepsilon^d \subset A\}$ and its boundary is $\partial A = \mathrm{cl}(A) \setminus \mathrm{int}(A)$. When A is convex, the relative interior of A is $\mathrm{ri}(A) = \{x \in A \colon \exists \varepsilon > 0, (x + \mathbb{B}_\varepsilon^d) \cap \mathrm{aff}(A) \subset A\}$ where $\mathrm{aff}(A)$ is the affine hull of A. The polar of A is $A^\circ = \{u \colon \sup_{x \in A} \langle x, u\rangle \leq 1\}$. We use $\mathbb{1}$ for the identity matrix and $\mathbf{0}$ for the zero matrix or zero vector. Dimensions and types will always be self-evident from the context. The euclidean projection onto K is $\Pi_K(x) = \arg\min_{y \in K} \|x - y\|$. Suppose that $f \colon \mathbb{R}^d \supset A \to \mathbb{R}$ is differentiable at $x \in A$; then we write $f'(x)$ for its gradient and $f''(x)$ for its Hessian. When f is convex we write $\partial f(x)$ for the set of subderivatives of f at x. More generally, $Df(x)[h]$ is the directional derivative of f at x in the direction h. Higher-order directional derivatives are denoted by $D^k f(x)[h_1, \ldots, h_k]$. Note that for convex f, $Df(x)[h]$ is defined for all $x \in \mathrm{int}(\mathrm{dom}(f))$ and $h \in \mathbb{R}^d$ but the mapping

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

$h \mapsto Df(x)[h]$ need not be linear, as the convex function $\mathbb{R}\colon x \mapsto |x|$ shows. Densities are always with respect to the Lebesgue measure. The diameter of a nonempty set K is

$$\operatorname{diam}(K) = \sup_{x,y \in K} \|x - y\| \,.$$

The Lipschitz constant of a function $f\colon K \to \mathbb{R}$ is

$$\operatorname{lip}_K(f) = \sup \left\{ \frac{f(x) - f(y)}{\|x - y\|} \colon x, y \in K, x \neq y \right\} .$$

When $f\colon \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is convex we write $\operatorname{lip}(f)$ to mean $\operatorname{lip}_{\operatorname{dom}(f)}(f)$ where $\operatorname{dom}(f) = \{x\colon f(x) < \infty\}$. Suppose that $A, B \in \mathbb{S}_+^d$. Then $A \preceq B$ if $B - A \in \mathbb{S}_+^d$ and $A \succeq B$ if $A - B \in \mathbb{S}_+^d$. $A \prec B$ and $A \succ B$ are defined similarly but with $\mathbb{S}_+^d$ replaced by $\mathbb{S}_{++}^d$.

Probability Spaces

We will not formally define the probability space on which the essential random variables $X_1, Y_1, \ldots, X_n, Y_n$ live. You can see how this should be done in the book by Lattimore and Szepesvári (2020). In general $\mathbb{P}$ is the probability measure on some space carrying these random variables and we let $\mathcal{F}_t = \sigma(X_1, Y_1, \ldots, X_t, Y_t)$ be the σ-algebra generated by the first t rounds of interaction. Events are often defined as $\{\text{condition}\}$. For example, $\{f(X_t) > \varepsilon\}$ is the event that $f(X_t) > \varepsilon$. We abbreviate $\mathbb{P}_t(\cdot) = \mathbb{P}(\cdot|\mathcal{F}_t)$ and $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_t]$.The multivariate Gaussian distribution with mean μ and covariance Σ is $\mathcal{N}(\mu, \Sigma)$. Given $A \subset \mathbb{R}^d$ we let $\mathcal{U}(A)$ be the uniform probability measure on A, which can be defined in multiple ways. We only need it when A is finite or $A \in \{\mathbb{B}_r^d, \mathbb{S}_r^{d-1}\}$, where the definition is obvious.

Regret

Recall the regret is defined by

$$\operatorname{Reg}_n = \sup_{x \in K} \sum_{t=1}^{n} (f_t(X_t) - f_t(x)) \,.$$

The sup cannot always be replaced by a max because even for compact K the loss functions may not be continuous. For example, when $K = [0, 1]$ and $f\colon K \to [0, 1]$ is defined by $f(0) = 1$ and $f(x) = x$ for $x > 0$, then f is convex and does not have a minimiser on K. We occasionally need the regret relative to a specific $x \in K$, which is

$$\operatorname{Reg}_n(x) = \sum_{t=1}^{n} (f_t(X_t) - f_t(x)) \,.$$

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

1.5 Notes

1.i The function classes outlined in Definition 1.1 are by no means the only ones considered. For example, the definition of smoothness can be generalised beyond the second-order smoothness. There are multiple ways to do this, but one is to call a function $f\colon \mathbb{R}^d \to \mathbb{R}$ smooth of order $p \in [2, \infty)$ on $\mathbb{R}^d$ if for all $x, y \in \mathbb{R}^d$,

$$\|D^q f(x) - D^q f(y)\| \leq \beta \|x - y\|^{p-q} ,$$

where q is the largest integer strictly smaller than p and the norm of the left-hand side is the operator norm. When $p = 2$ and f is convex this definition is equivalent to what appears in Definition 1.1 (Nesterov, 2018, Theorem 2.1.5). Zeroth-order convex optimisation has been studied for highly smooth functions with slightly varying definitions of smoothness by Polyak and Tsybakov (1990); Bach and Perchet (2016); Akhavan et al. (2020, 2024b) and others. This line of work is briefly discussed in the notes of Chapter 6.

1.ii The focus of this book is on a particular kind of *constrained* optimisation where the learner must play in a convex body K, which by definition is compact, convex and has a nonempty interior. As mentioned, only boundedness and convexity are really important for the results in this book. But assuming boundedness discards many interesting settings. Most fundamentally, it prohibits the *unconstrained* setting where $K = \mathbb{R}^d$. Similarly, K cannot be a half-space or other unbounded subset of $\mathbb{R}^d$. Practically speaking, unbounded domains can often be reduced to bounded ones with prior knowledge that the minimiser must lie in a bounded convex set. But such reductions do not always play well with the other assumptions; for example, that the loss is bounded on K.

1.iii Another situation related to the constraint set that arises often is that the losses are defined on all of $\mathbb{R}^d$ and the learner can query any point in $\mathbb{R}^d$ but is only evaluated against the best point in K. This setting is sometimes referred to as the *improper* setting. The definition ensures that the improper setting is easier than both the constrained setting (the learner has more power) and the unconstrained setting (the adversary has less power).

1.iv Other performance criteria also exist in the literature. For example, both the distance to the minimiser and the gradient of the loss are potentially sensible measures of quality. Neither of these is considered in this book.

1.v We assume throughout that the noise either vanishes completely or is 1-subgaussian: $\mathbb{E}_{t-1}[\exp(\varepsilon_t^2)|X_t] \leq 2$ and $\mathbb{E}_{t-1}[\varepsilon_t] = 0$. At the same time, we often assume that the range of the losses is in $[0, 1]$. This means the scale of the

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press

noise is the same as the scale of the losses and makes it hard to know how the range or variance of the noise might affect the regret bound if either of these quantities were scaled individually. One analysis that decouples these quantities is given in Section 6.6; occasionally we give pointers to the literature or suggest exercises/open problems.

1.vi More exotic interaction protocols have also been investigated. Consider for a moment adversarial setting without noise. A number of authors have explored what changes if the learner is allowed to choose *two* points $X_{t,1}, X_{t,2} \in K$ and observes $f_t(X_{t,1})$ and $f_t(X_{t,2})$. One might believe that such a modification would have only a mild effect but this is not at all the case. Having access to two evaluations makes the bandit setup behave more like the full information setting (Agarwal et al., 2010; Nesterov and Spokoiny, 2017; Duchi et al., 2015). Essentially the learner can compute directional derivatives to arbitrary precision, which is not possible in the standard setting. Note that in the stochastic setting the noise is what makes it impossible to compute directional derivatives, while in the adversarial setting there need not be any relation from one loss to the next.

1.vii At the end of the day, the number of settings and possible assumptions is *enormous*. We have naturally selected a subset most aligned with our interest and expertise. Many of the algorithms and analysis presented can be generalised straightforwardly to other settings. But sometimes the conditions and modifications are subtle. As much as possible we try to give pointers to the literature, but no doubt much has been missed.

https://doi.org/10.1017/9781009607551.002 Published online by Cambridge University Press