Lectures on Optimization

Sinho Chewi

February 19, 2025

Contents

1 [1/14] Introduction and basics of convex functions

These lecture notes supplement S&DS 432/632 (Advanced Optimization Techniques), taught in Spring 2025. They are not meant to be comprehensive.

The notes are primarily based on the books [Bub15; Nes18], as well as my personal understanding of the subject formed through discussions with many people over the years. Please send me corrections and feedback via email. I thank Linghai Liu, Ruixiao Wang, and Ilias Zadik for correcting my mistakes.

Logistics. The problem sets, syllabus, and all other information can be found at the course website and the Canvas page. Grading is based on six problem sets and one take-home final exam, each of which counts for $1/7$ of the total grade. All questions related to logistics should be directed to my email: sinho.chewi@yale.edu.

Audience. This course focuses on the theory of optimization. In particular, the course is **mathematical** in nature and taught in a theorem–proof format. The course assumes familiarity with basic proofs and logical reasoning, as well as linear algebra, multivariate calculus, and probability theory.

The reader should also be familiar with asymptotic notions (big-O notation). We use the shorthand notation $a \lesssim b$ (resp. $a \gtrsim b$) to mean that $a \le Cb$ (resp. $a \ge b/C$) for an absolute constant $C > 0$ (i.e., a constant that does not depend on other parameters of the problem), and $a \asymp b$ to mean that both $a \lesssim b$ and $a \gtrsim b$ hold. We use $a = O(b)$ and $a \lesssim b$ interchangeably.

1.1 Overview of the course

The basic problem of optimization is to compute an approximate minimizer of a given function $f : \mathcal{X} \to \mathbb{R}$. In this course, $\mathcal{X}$ is always taken to be a subset of $\mathbb{R}^d$, although generalizations are possible (e.g., to manifolds).

Black-box optimization and the oracle model. What does it mean to "compute"? The answer depends on the representation of f and our model of computation. We start by studying *black-box optimization*. In this model, we presume that we can *evaluate* f, and possibly its derivatives, at any chosen point $x \in \mathcal{X}$.

The advantage of the black-box model is that it applies very *generally*: it is difficult to find situations in which we need to optimize a function but we cannot even evaluate it! Consequently, algorithms developed in this model can be applied to the majority of problems encountered in practice[1]—witness the ubiquity of gradient descent.

The disadvantage is that by its very generality, it cannot take advantage of additional structural information about f which can bring computational savings. That is why, later in the course, we turn toward the study of *structured* optimization problems.

It is easy, at least at an intuitive level, to describe algorithms which are valid in the black-box model. Namely, they are algorithms which only "interact" with f through evaluations of f and its derivatives. The existence of an algorithm, together with a corresponding mathematical analysis of the number of iterations to reach an approximate minimizer contingent upon assumptions on f, provide an *upper bound* on the complexity of the optimization task. In this course, we are also interested in *lower bounds*, which delineate fundamental limitations encountered by *any* algorithm. In order to prove such a lower bound, we need to formalize the notion of "interaction" alluded to above, and this leads to the important concept of an *oracle*.

First, observe that it does not make sense to discuss the complexity of optimizing a *single* function f. For if $x_\star$ is the minimizer of f, we can consider the algorithm "output $x_\star$", which yields the correct answer in one iteration. But this algorithm is silly, since it utterly fails at optimizing any other function whose minimizer does not happen to be $x_\star$. Reflecting upon this situation, we do not consider an optimization algorithm to be sensible when it happens to succeed for one particular problem; rather, we expect it to succeed on many similar problems. Hence, we talk about a *class* of functions $\mathcal{F}$ of interest, and we require our algorithms to succeed on *every* $f \in \mathcal{F}$.

The algorithm is designed to succeed on $\mathcal{F}$ and thus, in an anthropomorphic sense, it "knows" $\mathcal{F}$. However, it does not know which particular $f \in \mathcal{F}$ it is trying to optimize. (If it possessed knowledge of f, then we run into the issue from before, namely it could simply output the minimizer.) The role of the oracle is to act as an intermediary between the algorithm and the function. Namely, we assume that the algorithm is allowed to ask certain questions ("queries") to the oracle for f, and this is the only means by which the algorithm can gather more information about f. The allowable queries and responses determine the nature of the oracle, e.g.:

- a **zeroth-order oracle** accepts a query point $x \in \mathbb{R}^d$ and outputs $f(x)$;
- a **first-order oracle** accepts a query point $x \in \mathbb{R}^d$ and outputs $(f(x), \nabla f(x))$.

Most of the course focuses on optimization with a first-order oracle, but other oracles are possible (e.g., *linear optimization oracles* and *proximal oracles*). The zeroth-order and

[1] There is a caveat: in this course, we solely consider continuous optimization problems. Combinatorial optimization is an entirely different beast.

first-order oracles are easy to justify, as they correspond to the black-box model described above. As the oracles become more exotic, it becomes necessary to show that they are reasonable, by describing important applications in which such access to f is feasible.

The *query complexity* of $\mathcal{F}$ for a particular choice of oracle, as a function of the prescribed tolerance ε, is then (informally) defined to be the minimum number N such that there exists an algorithm which, for any $f \in \mathcal{F}$, makes N queries to the oracle for f and outputs a point x with $f(x) - \min f \le \varepsilon$.

It is worth noting that query complexity is not the same as computational complexity. Indeed, query complexity only counts the number of interactions with the oracle, and the algorithm is allowed to perform unlimited computations between interactions. In principle, this could lead to a situation in which query complexity is wholly unrepresentative of the true computational cost of optimization—this would be the case if optimal algorithms in the oracle model were contrived and impractical. Thankfully, this is not the case. The oracle model is widely adopted as the standard model for optimization because it is the setting in which we can make precise claims about complexity, and because it generally aligns with optimization in practice.

This summarizes the conceptual framework for optimization theory—the "identity cards of the field" [Nes18], although a careful treatment of the framework only becomes necessary when discussing lower bounds (and hence we elaborate on the details then). As a branch of mathematics, the theory of optimization could be defined as the quest to characterize the query complexity of various classes $\mathcal{F}$, under various oracle models, and thereby identify optimal algorithms. This indeed remains a core element of the field, but as query complexity reaches maturity, research has shifted toward different types of questions, often inspired by practical developments.

The role of convexity. In order to optimize efficiently, we need to place assumptions on f, ideally minimal ones. For example, we can assume that f is continuous. In this course, however, we are interested in *quantitative* rates of convergence for algorithms, and for this purpose, a *qualitative* assumption such as continuity is not enough. A quantitative form of continuity is to assume that f is *L-Lipschitz in the ℓ_∞ norm*:

$$|f(x) - f(y)| \le L \max_{i \in [d]} |x[i] - y[i]| \qquad \text{for all } x, y \in \mathcal{X}\,. \tag{1.1}$$

Also, for concreteness, let us take $\mathcal{X}$ to be the cube, $\mathcal{X} = [0,1]^d$. In the language of the framework above, we consider the class

$$\mathcal{F} = \{f : [0,1]^d \to \mathbb{R} \mid f \text{ satisfies (1.1)}\}\,. \tag{1.2}$$

One can then prove the following negative result.

Theorem 1.1. For any $0 < \varepsilon < L/2$ and any deterministic algorithm, the complexity of ε-approximately minimizing functions in the class defined in (1.2) to within ε using a zeroth-order oracle is at least $\lfloor \frac{L}{2\varepsilon} \rfloor^d$.

Thus, for $\varepsilon < L/4$, the complexity grows *exponentially* with the dimension. The proof is not difficult; see, e.g., [Nes18, Theorem 1.1.2]. It is also robust: variants of the result can be proven when the notion of Lipschitzness is w.r.t. the ℓ_2 norm; when the oracle is taken to be a first-order oracle; when the algorithm is allowed to be randomized; etc. The message is clear: in order for optimization to be tractable in the worst case, we must impose some structural assumptions.

The black-box oracles we have been considering are *local* in nature: given a query point $x \in \mathbb{R}^d$, the oracle reveals some information about the behavior of f in a local neighborhood of x. Assumptions such as Lipschitzness effectively govern how large this local neighborhood is. But ultimately, to render optimization tractable, we must ensure that local information yields global consequences. As justified in the next subsection, a key assumption that makes this possible is *convexity*.

Of course, not every problem is convex, and non-convex optimization often still succeeds. But for the purpose of understanding the core principles underlying optimization, there is no better starting place. It is important to remember that convex problems abound in every application domain; here, we give two classical examples from statistics.

Example 1.2 (logistic regression)**.** The data consists of n pairs $(X_i, Y_i) \in \mathbb{R}^d \times \{0, 1\}$, where X_i is a vector of covariates and Y_i is a binary response. The statistical model assumes that the pairs are independently drawn, the covariates are deterministic, and Y_i has a Bernoulli distribution with parameter $\exp(\langle \theta, X_i \rangle)/\{1 + \exp(\langle \theta, X_i \rangle)\}$. The goal is to infer the parameter θ.

The maximum likelihood estimator (MLE) for this model is the solution to the convex optimization problem

$$\widehat{\theta}_{\mathrm{MLE}} \in \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \bigl(\log(1 + \exp\langle \theta, X_i \rangle) - Y_i \langle \theta, X_i \rangle\bigr) .$$

Example 1.3 (LASSO)**.** The data consists of n pairs $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$. The statistical model assumes that the pairs are independently drawn, and that $Y_i = \langle \theta, X_i \rangle + \xi_i$, where the ξ_i's are i.i.d. noise variables independent of the X_i's. When the parameter θ is assumed to be sparse, it is standard to use the LASSO estimator, which is the solution to the convex optimization problem

$$\widehat{\theta}_{\text{LASSO}} = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (Y_i - \langle \theta, X_i \rangle)^2 + \lambda \, \|\theta\|_1 \right\}.$$

Here, $\lambda > 0$ is the regularization parameter and $\|\cdot\|_1$ denotes the ℓ_1 norm, defined via $\|\theta\|_1 := \sum_{i=1}^d |\theta[i]|$.

In these examples, the estimator is defined as the solution to a convex problem which is not solvable in closed form, necessitating the use of numerical optimization. Actually, it is not that most problems in the "wild" are convex and hence there was a need to develop convex optimization. In fact, it often goes the other way around: convex optimization is such a powerful tool that problems are intentionally formulated to be convex. This is the case for the LASSO estimator, which can be motivated as a convex relaxation of the (statistically superior) ℓ_0-constrained least-squares estimator.

First-order methods. This course largely focuses on first-order methods, namely, gradient descent and its variants. This class of methods is natural from the perspective of the theory. Equally importantly, first-order methods are lightweight and therefore scalable to large problem sizes, making them the method of choice even for highly non-convex settings which fall squarely outside of the theory.

Beyond the black-box model. After developing results for the black-box model, we study structured problems which admit more efficient solutions. The LASSO estimator of Example 1.3 can be treated as a "composite" optimization problem (a sum of a smooth and a non-smooth function), and the estimators in both Example 1.2 and Example 1.3 (and empirical risk minimization more generally) are "finite sum" problems whose computation can be sped up via the use of stochastic gradients. Other examples include the use of alternative geometries (mirror descent) and the use of coordinate-wise structure (alternating maximization/coordinate descent).

We also study interior-point methods, which are a practically effective suite of algorithms which solve linear programs (LPs) and semidefinite programs (SDPs) with polynomial iteration complexities.

Further topics are considered as time permits.

1.2 Preliminaries on convexity and smoothness

We assume familiarity with the basic notion of convexity, and we briefly review it here.

Definition 1.4. A subset $\mathcal{C} \subseteq \mathbb{R}^d$ is **convex** if for all $x, y \in \mathcal{C}$ and all $t \in [0,1]$, the point $(1-t)\,x + t\,y$ also lies in $\mathcal{C}$.

Definition 1.5. Let $\mathcal{C}$ be convex and let $\alpha \ge 0$. A function $f : \mathcal{C} \to \mathbb{R}$ is α-**convex** if for all $x, y \in \mathcal{C}$ and all $t \in [0,1]$,

$$f((1-t)\,x + t\,y) \le (1-t)\,f(x) + t\,f(y) - \frac{\alpha}{2}\,t\,(1-t)\,\|y-x\|^2\,. \tag{1.3}$$

When $\alpha = 0$, this is just the usual definition of a convex function. When $\alpha > 0$, we say that the function is *strongly* convex.

The definition above has the advantage that it does not require f to be differentiable. However, for the purposes of checking and utilizing convexity, it is convenient to have the following equivalent reformulations, which should be committed to memory. For simplicity, we focus on the case $\mathcal{C} = \mathbb{R}^d$.

Proposition 1.6 (convexity equivalences). Let $\mathcal{C} = \mathbb{R}^d$ and $\alpha \ge 0$.

1. If f is continuously differentiable, (1.3) is equivalent to each of the following:

$$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\alpha}{2}\,\|y-x\|^2 \qquad \text{for all } x, y \in \mathbb{R}^d\,. \tag{1.4}$$

$$\langle \nabla f(y) - \nabla f(x), y - x\rangle \ge \alpha\,\|y-x\|^2 \qquad \text{for all } x, y \in \mathbb{R}^d\,. \tag{1.5}$$

2. If f is twice continuously differentiable, (1.3) is equivalent to

$$\langle v, \nabla^2 f(x)\,v\rangle \ge \alpha\,\|v\|^2 \qquad \text{for all } v, x \in \mathbb{R}^d\,. \tag{1.6}$$

Proof. Assume that f is continuously differentiable.

(1.3) $\Longrightarrow$ (1.4): Rearranging (1.3) yields, for $t > 0$,

$$f(y) \ge f(x) + \frac{f((1-t)\,x + t\,y) - f(x)}{t} + \frac{\alpha\,(1-t)}{2}\,\|y-x\|^2\,.$$

Sending $t \searrow 0$ yields (1.4).

(1.4) ⇒ (1.5): Swap x and y in (1.4) and add the resulting inequality back to (1.4).

(1.5) ⇒ (1.3): By the fundamental theorem of calculus, for $v := y - x$,

$$f(y) = f(x) + \int_0^1 \langle \nabla f(x + sv), v \rangle \, \mathrm{d}s \,,$$
$$f((1-t)\,x + t\,y) = f(x) + \int_0^1 \langle \nabla f(x + stv), tv \rangle \, \mathrm{d}s \,.$$

Hence, (1.5) yields

$$\begin{aligned} f((1-t)\,x + t\,y) - (1-t)\,f(x) - t\,f(y) &= -t \int_0^1 \langle \nabla f(x + sv) - \nabla f(x + stv), v \rangle \, \mathrm{d}s \\ &\le -t \int_0^1 \alpha s\,(1-t)\,\|v\|^2 \, \mathrm{d}s = -\frac{\alpha}{2}\,t\,(1-t)\,\|v\|^2 \,. \end{aligned}$$

Finally, assume that f is twice continuously differentiable. Letting $y = x + \varepsilon v$ in (1.5) and sending $\varepsilon \searrow 0$ establishes (1.6). Conversely, the fundamental theorem of calculus shows that

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle = \int_0^1 \langle \nabla^2 f(x + t\,(y - x))\,(y - x),\, y - x \rangle \, \mathrm{d}t \,,$$

and hence (1.6) implies (1.5). □

The equivalent statements each have their own interpretation: for $\alpha = 0$, (1.3) states that f lies below each of its secant lines between the intersection points; (1.4) states that f globally lies above each of its tangent lines; (1.6) states that ∇f is a monotone vector field; and (1.6) is a statement about curvature.

As noted above, the key feature of convexity is that local information yields global conclusions. Before describing this, let us first recall some basic facts about optimization. For simplicity, we consider unconstrained optimization throughout.

Lemma 1.7 (existence of minimizer)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuous and its level sets be bounded. Then, there exists a global minimizer of f.

Proof. The proof uses some analysis. Let $x_0 \in \mathbb{R}^d$ and let $\mathcal{K} := \{f \le f(x_0)\}$ denote the level set. By the continuity assumption, $\mathcal{K}$ is closed and bounded, thus compact. Let $\{x_n\}_{n\in\mathbb{N}}$ be a minimizing sequence, $f(x_n) \to \inf f$. By compactness, it admits a subsequence, still denoted $\{x_n\}_{n\in\mathbb{N}}$, which converges to some $x_\star \in \mathbb{R}^d$. By continuity, $f(x_\star) = \lim_{n\to\infty} f(x_n) = \inf f$. □

Lemma 1.8 (necessary conditions for optimality)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be minimized at $x_\star$.

1. If f is continuously differentiable, then $\nabla f(x_\star) = 0$.
2. If f is twice continuously differentiable, then $\nabla^2 f(x_\star) \succeq 0$.

Proof. Let $v \in \mathbb{R}^d$ and $\varepsilon > 0$; then, $f(x_\star + \varepsilon v) - f(x_\star) \ge 0$. If f is continuously differentiable, this yields $\int_0^1 \langle \nabla f(x_\star + \varepsilon t v), v\rangle \, \mathrm{d}t \ge 0$. By continuity of ∇f, sending $\varepsilon \searrow 0$ proves that $\langle \nabla f(x_\star), v\rangle \ge 0$ for all $v \in \mathbb{R}^d$, which entails $\nabla f(x_\star) = 0$.

If f is twice continuously differentiable, we can expand once more to obtain $0 \le \int_0^1 \int_0^1 \langle \nabla^2 f(x_\star + \varepsilon s t v)\, v, v\rangle \, \mathrm{d}s \, \mathrm{d}t$. By continuity of $\nabla^2 f$, sending $\varepsilon \searrow 0$ then proves that $\langle \nabla^2 f(x_\star)\, v, v\rangle \ge 0$ for all $v \in \mathbb{R}^d$. □

The conditions $\nabla f(x_\star) = 0$, $\nabla^2 f(x_\star) \succeq 0$ are necessary for optimality, but not sufficient in general. The issue is that the proof of Lemma 1.8 is entirely local, so the same conclusion holds even if $x_\star$ is only assumed to be a *local* minimizer. On the other hand, under the assumption of convexity, the first-order necessary condition becomes sufficient.

Lemma 1.9 (sufficient condition for optimality)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be *convex* and continuously differentiable, and let $\nabla f(x_\star) = 0$. Then, $x_\star$ is a global minimizer of f.

In particular, every local minimizer of f is a global minimizer.

Proof. This easily follows from (1.4) with $x = x_\star$. □

Next, we note that the minimizer is unique if f is strictly convex.

Lemma 1.10 (uniqueness of minimizer)**.** Assume that $f : \mathbb{R}^d \to \mathbb{R}$ is strictly convex, i.e., for all distinct $x, y \in \mathbb{R}^d$ and $t \in (0,1)$, $f((1-t)\,x + t\,y) < (1-t)\,f(x) + t\,f(y)$. Then, if f admits a minimizer $x_\star$, it is unique.

Proof. If we had two distinct minimizers $x_\star$, $\tilde{x}_\star$, so that $f(x_\star) = f(\tilde{x}_\star)$, then strict convexity would imply $f(\frac{1}{2}\,x_\star + \frac{1}{2}\,\tilde{x}_\star) < f(x_\star)$, which is a contradiction. □

If f is strongly convex, then it is strictly convex. Also, from, e.g., (1.4), we see that f grows at least quadratically at ∞, which implies that it has bounded level sets. We can therefore conclude:

Corollary 1.11. Let $f : \mathbb{R}^d \to \mathbb{R}$ be strongly convex and continuously differentiable. Then, it admits a unique minimizer $x_\star$, which is characterized by $\nabla f(x_\star) = 0$.

Finally, when discussing algorithms, we also need a dual condition—an *upper* bound on the Hessian—which in this context is called *smoothness*.[2]

Definition 1.12. Let $\beta \ge 0$. We say that $f : \mathbb{R}^d \to \mathbb{R}$ is β-**smooth** if it is continuously differentiable and

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|y - x\|^2 \qquad \text{for all } x, y \in \mathbb{R}^d\,. \tag{1.7}$$

The following proposition is established in the same way as Proposition 1.6, so we omit the proof.

Proposition 1.13 (smoothness equivalences)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable and $\beta \ge 0$. Then, f is β-smooth if and only if

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \le \beta \|y - x\|^2 \qquad \text{for all } x, y \in \mathbb{R}^d\,.$$

If f is twice continuously differentiable, this is also equivalent to

$$\langle v, \nabla^2 f(x)\, v \rangle \le \beta \|v\|^2 \qquad \text{for all } v, x \in \mathbb{R}^d\,.$$

If f is convex, β-smooth, and twice continuously differentiable, then $0 \preceq \nabla^2 f \preceq \beta I$, which implies that the gradient ∇f is β-Lipschitz:

$$\|\nabla f(y) - \nabla f(x)\| \le \beta \|y - x\| \qquad \text{for all } x, y \in \mathbb{R}^d\,. \tag{1.8}$$

This remains true even without assuming twice differentiability (Exercise 3.1).

Bibliographical notes

For further discussion on the oracle model, see [NY83, §1].

Exercises

Exercise 1.1. Let $f = \frac{\alpha}{2} \|\cdot\|^2$, where $\alpha \ge 0$. Show via direct computation that (1.3) holds with equality.

[2] This is not to be confused with the mathematical usage of "smoothness" as "infinitely differentiable".

2 [1/16] Gradient flow

Before we turn toward our main first-order algorithm of interest, namely gradient descent, we first study the situation in continuous time via the gradient flow. Throughout this section, we let $(x_t)_{t\geq 0}$ denote the gradient flow for f:

$$\dot{x}_t = -\nabla f(x_t)\,. \tag{GF}$$

This is an ordinary differential equation (ODE), and since the main purpose of this section is to develop intuition, we assume that f is twice continuously differentiable and do not worry about showing that (GF) is well-posed. We use the following notation throughout these notes:

$$x_\star \in \arg\min f\,, \qquad f_\star := \min f = f(x_\star)\,.$$

Generally, we always assume that f admits a minimizer.

The most basic property of GF is that it always decreases the function value.

Lemma 2.1 (descent property of GF)**.** For any $f : \mathbb{R}^d \to \mathbb{R}$, the gradient flow $(x_t)_{t\geq 0}$ of f satisfies

$$\partial_t f(x_t) = -\|\nabla f(x_t)\|^2 \leq 0\,.$$

Proof. By the chain rule, $\partial_t f(x_t) = \langle \nabla f(x_t), \dot{x}_t\rangle = -\|\nabla f(x_t)\|^2$. □

To obtain quantitative convergence results, we now use the assumption of convexity. Our first result shows that under strong convexity, the gradient flow *contracts*.

Theorem 2.2 (contraction of GF)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be α-convex. Let $(y_t)_{t\geq 0}$ be another gradient flow for f, i.e., $\dot{y}_t = -\nabla f(y_t)$. Then, for all $t \geq 0$,

$$\|y_t - x_t\| \leq \exp(-\alpha t)\,\|y_0 - x_0\|\,.$$

Proof. We differentiate the squared distance between the two flows:

$$\partial_t(\|y_t - x_t\|^2) = 2\,\langle y_t - x_t, \dot{y}_t - \dot{x}_t\rangle = -2\,\langle y_t - x_t, \nabla f(y_t) - \nabla f(x_t)\rangle \leq -2\alpha\,\|y_t - x_t\|^2\,,$$

where the last inequality is (1.5). The proof is concluded by applying Grönwall's lemma (see Lemma 2.3) below. □

The proof above arrives at what is called a *differential inequality*, that is, an inequality which holds between a quantity and its derivative(s). This is a common strategy for analyzing ODEs/PDEs, and it can be loosely viewed as the continuous-time analogue of induction. The following standard lemma is useful for handling such inequalities.

Lemma 2.3 (Grönwall). Suppose that $u : [0,T] \to \mathbb{R}$ is a continuously differentiable curve that satisfies the differential inequality

$$\dot u(t) \le A u(t) + B(t)\,, \qquad t \in [0,T]\,.$$

Then, it holds that

$$u(t) \le u(0) \exp(At) + \int_0^t B(s) \exp(A\,(t-s))\,\mathrm{d}s\,, \qquad t \in [0,T]\,.$$

Proof. The idea is to differentiate $t \mapsto \exp(-At)\,u(t)$:

$$\partial_t[\exp(-At)\,u(t)] = \exp(-At)\,\{-Au(t) + \dot u(t)\} \le B(t)\exp(-At)\,.$$

By the fundamental theorem of calculus,

$$\exp(-At)\,u(t) - u(0) \le \int_0^t B(s)\exp(-As)\,\mathrm{d}s\,.$$

Rearranging yields the result. □

There are many variants of Grönwall's lemma that can be proven in similar ways, e.g., we can allow time-varying A as well.

Returning to Theorem 2.2, we can apply Lemma 2.3 with $A = -2\alpha$ and $B = 0$ to conclude that $\|y_t - x_t\|^2 \le \exp(-2\alpha t)\,\|y_0 - x_0\|^2$, which proves the theorem. Note in particular that we can take $y_t = x_\star$ for all $t \ge 0$, so it yields the following statement about convergence to the minimizer: $\|x_t - x_\star\| \le \exp(-\alpha t)\,\|x_0 - x_\star\|$.

The next result is about convergence in function value, and unlike Theorem 2.2, it yields convergence for the case $\alpha = 0$ as well.

Theorem 2.4 (convergence of GF in function value). Let $f : \mathbb{R}^d \to \mathbb{R}$ be α-convex, $\alpha \ge 0$. Then, for all $t \ge 0$,

$$f(x_t) - f_\star \le \frac{\alpha}{2\,(\exp(\alpha t) - 1)}\,\|x_0 - x_\star\|^2\,.$$

When $\alpha = 0$, the right-hand side should be interpreted as its limiting value as $\alpha \to 0$, namely, $\frac{1}{2t}\,\|x_0 - x_\star\|^2$.

Proof. We differentiate $t \mapsto \|x_t - x_\star\|^2$, but this time we apply (1.4):

$$\partial_t(\|x_t - x_\star\|^2) = -2\,\langle \nabla f(x_t), x_t - x_\star\rangle \le -\alpha\,\|x_t - x_\star\|^2 - 2\,(f(x_t) - f_\star)\,.$$

Applying Grönwall's lemma (Lemma 2.3) with $A = -\alpha$, $B(t) = -2\,(f(x_t) - f_\star)$,

$$0 \le \|x_t - x_\star\|^2 \le \exp(-\alpha t)\,\|x_0 - x_\star\|^2 - 2\int_0^t \exp(-\alpha\,(t-s))\,(f(x_s) - f_\star)\,\mathrm{d}s\,.$$

By the descent property (Lemma 2.1), $f(x_s) \ge f(x_t)$, so that

$$\begin{aligned}\int_0^t \exp(-\alpha\,(t-s))\,(f(x_s) - f_\star)\,\mathrm{d}s &\ge (f(x_t) - f_\star)\int_0^t \exp(-\alpha\,(t-s))\,\mathrm{d}s \\ &= (f(x_t) - f_\star)\,\frac{1-\exp(-\alpha t)}{\alpha}\,.\end{aligned}$$

Rearranging yields the result. □

When $\alpha > 0$, Theorem 2.4 shows that $f(x_t) - f_\star = O(\exp(-\alpha t))$. When $\alpha = 0$, the rate becomes $f(x_t) - f_\star = O(1/t)$. Actually, the rate in Theorem 2.4 is not sharp (see Exercise 2.1 and Exercise 2.2). However, the statement and proof are chosen because they form the basis of our approach in discrete time.

Next, we observe that convexity is not needed for convergence in function value. Due to the descent property (Lemma 2.1), it suffices to have a lower bound on the norm of the gradient to ensure that we make sufficient progress. For example, we can impose the following condition.

Definition 2.5. Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable and $\alpha > 0$. We say that f satisfies a **Polyak–Łojasiewicz (PŁ) inequality** with constant α if

$$\|\nabla f(x)\|^2 \ge 2\alpha\,(f(x) - f(x_\star)) \qquad \text{for all } x \in \mathbb{R}^d\,. \tag{PŁ}$$

The next statement is an immediate corollary of Lemma 2.1, (PŁ), and Grönwall's lemma (Lemma 2.3).

Corollary 2.6 (convergence of GF under PŁ). Let $f : \mathbb{R}^d \to \mathbb{R}$ satisfy (PŁ) with constant $\alpha > 0$. Then, for all $t \ge 0$,

$$f(x_t) - f_\star \le (f(x_0) - f_\star)\exp(-2\alpha t)\,.$$

We present a few key properties of the PŁ inequality.

Proposition 2.7 (strong convexity ⇒ PŁ ⇒ quadratic growth)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ and $\alpha > 0$. The following implications hold.

1. If f is α-convex, then f satisfies (PŁ) with constant α.
2. If f satisfies (PŁ) with constant α, then it satisfies the following **quadratic growth** property:

$$f(x) - f_\star \ge \frac{\alpha}{2} \inf_{x_\star \in \mathcal{X}_\star} \|x - x_\star\|^2\,, \qquad \text{for all } x \in \mathbb{R}^d\,, \tag{QG}$$

   where $\mathcal{X}_\star$ denotes the set of minimizers of f.

Proof.

1. Setting $y = x_\star$ in (1.4), we obtain

$$\begin{aligned} -(f(x) - f_\star) &\ge \langle \nabla f(x), x_\star - x\rangle + \frac{\alpha}{2}\,\|x - x_\star\|^2 \\ &\ge -\|\nabla f(x)\|\,\|x_\star - x\| + \frac{\alpha}{2}\,\|x - x_\star\|^2 \ge -\frac{1}{2\alpha}\,\|\nabla f(x)\|^2\,, \end{aligned}$$

   where the last inequality uses $ab \le \frac{\lambda}{2}\,a^2 + \frac{1}{2\lambda}\,b^2$ for all $\lambda > 0$.

2. Let $(x_t)_{t\ge 0}$ denote the gradient flow for f started at $x_0 = x$. For simplicity, we present a proof *assuming* that the gradient flow converges to a point $x_\star$, although this assumption can be avoided (cf. [KNS16]). By Corollary 2.6, we see that $x_\star \in \mathcal{X}_\star$.

   We start by observing that

$$\partial_t(\|x_t - x_0\|^2) = -2\,\langle \nabla f(x_t), x_t - x_0\rangle \le 2\,\|\nabla f(x_t)\|\,\|x_t - x_0\|$$

   and hence

$$\partial_t \|x_t - x_0\| \le \|\nabla f(x_t)\|\,.$$

   We differentiate the following quantity: $\mathscr{L}_t := \sqrt{\frac{\alpha}{2}}\,\|x_t - x_0\| + \sqrt{f(x_t) - f_\star}$.

$$\dot{\mathscr{L}}_t \le \sqrt{\frac{\alpha}{2}}\,\|\nabla f(x_t)\| - \frac{\|\nabla f(x_t)\|^2}{2\sqrt{f(x_t) - f_\star}} \le 0\,,$$

   where we applied (PŁ). Since $\mathscr{L}_0 = \sqrt{f(x) - f_\star}$ and $\mathscr{L}_\infty = \sqrt{\frac{\alpha}{2}}\,\|x - x_\star\|$, we deduce the result from $\mathscr{L}_0 \ge \mathscr{L}_\infty$.

□

Hence, strong convexity implies (PŁ), but is (PŁ) truly weaker than convexity? Indeed, there are examples. In particular, the PŁ condition has been of interest in recent years because it holds for certain overparametrized models (Exercise 2.3).

We conclude this section by studying the implication of Lemma 2.1 alone. The fundamental theorem of calculus shows that

$$\frac{1}{t}\int_0^t \|\nabla f(x_s)\|^2\,\mathrm{d}s \le \frac{f(x_0)-f(x_t)}{t} \le \frac{f(x_0)-f_\star}{t}\,.$$

We therefore arrive at the following simple consequence.

Corollary 2.8 (convergence of GF in gradient norm)**.** For any $f : \mathbb{R}^d \to \mathbb{R}$,

$$\min_{s\in[0,t]} \|\nabla f(x_s)\| \le \sqrt{\frac{f(x_0)-f_\star}{t}}\,.$$

(In contrast, note that if we additionally assume convexity, then Exercise 2.1 shows that $\|\nabla f(x_t)\| = O(1/t)$.)

This implies there exists a sequence of times $\{t_n\}_{n\in\mathbb{N}} \nearrow \infty$ such that $\|\nabla f(x_{t_n})\| \to 0$. (Indeed, $\min_{s\in[n,2n]} \|\nabla f(x_s)\| = O(1/n^{1/2})$, so we can choose $t_n \in [n, 2n]$.) However, the gradient flow may not converge. Famously, it is a result of [Łoj63] that for *real analytic* f, if the gradient flow remains bounded, then it does converge, and hence necessarily to a stationary point. Of course, such a stationary point may not be a global minimizer.

The idea of subsequent sections is to replicate the preceding analysis in discrete time.

Bibliographical notes

My understanding of Theorem 2.4, Exercise 2.1, and Exercise 2.2 is based on extensive discussions with Jason M. Altschuler, Adil Salim, Andre Wibisono, and Ashia Wilson. The proof in Exercise 2.1 is taken from [OV01], and the extension in Exercise 2.2 to $\alpha > 0$ is recorded in [LMW24, §F]. Both of these references pertain to the Langevin diffusion, but underneath the hood they make use of principles from optimization; see [Che25] for an introduction to this perspective.

The PŁ inequality is attributed to [Łoj63; Pol63] and it was popularized in [KNS16]. The proof that (PŁ) implies the quadratic growth inequality goes back at least to the celebrated work of [OV00].

Exercises

Exercise 2.1. Let f be convex. Show that the following quantity is decreasing, $\dot{\mathscr{L}}_t \le 0$:

$$\mathscr{L}_t := t^2 \left\|\nabla f(x_t)\right\|^2 + 2t\,(f(x_t) - f_\star) + \left\|x_t - x_\star\right\|^2 .$$

Deduce the following gradient bound:

$$\left\|\nabla f(x_t)\right\|^2 \le \frac{1}{t^2} \left\|x_0 - x_\star\right\|^2 .$$

Moreover, use (1.4) to argue that $2t\,(f(x_t) - f_\star) \le t^2 \left\|\nabla f(x_t)\right\|^2 + \left\|x_t - x_\star\right\|^2$, hence

$$f(x_t) - f_\star \le \frac{1}{4t} \left\|x_0 - x_\star\right\|^2 . \tag{2.1}$$

Note that this improves upon Theorem 2.4 by a factor of 2. Furthermore, show that (2.1) is sharp, as follows: for any $R, t > 0$, let $f : x \mapsto \frac{R}{2t} \max\{0, x\}$, $x_0 = R$, and show that (2.1) holds with equality.

Exercise 2.2. Extend Exercise 2.1 to the case $\alpha > 0$. Toward this end, consider

$$\mathscr{L}_t := A_t \left\|\nabla f(x_t)\right\|^2 + 2B_t\,(f(x_t) - f_\star) + \left\|x_t - x_\star\right\|^2 .$$

Choose A_t, B_t carefully to ensure that $\dot{\mathscr{L}}_t \le -\alpha \mathscr{L}_t$, and thereby deduce the following sharp bounds:

$$\left\|\nabla f(x_t)\right\|^2 \le \frac{\alpha^2 \left\|x_0 - x_\star\right\|^2}{\exp(2\alpha t)\,(1 - \exp(-\alpha t))^2} , \qquad f(x_t) - f_\star \le \frac{\alpha \left\|x_0 - x_\star\right\|^2}{2\,(\exp(2\alpha t) - 1)} .$$

Exercise 2.3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be α-convex with $\alpha > 0$, and let $g : \mathbb{R}^d \to \mathbb{R}^n$ with $d \ge n$. Assume that g is surjective and that for all $x \in \mathbb{R}^d$, if $\nabla g(x)$ denotes the Jacobian at x (interpreted as a $d \times n$ matrix), then $\nabla g(x)^\mathsf{T}\, \nabla g(x) \succeq \sigma I_n$. Show that the composition $f \circ g$ satisfies (PŁ) with constant $\alpha\sigma$. Note that for $d > n$, there are typically multiple minimizers of $f \circ g$.

3 [1/21] Gradient descent: smooth case

In this section, we study the **gradient descent** algorithm:

$$x_{n+1} := x_n - h\,\nabla f(x_n) . \tag{GD}$$

From the perspective of numerical analysis, this is the *Euler* or *forward* discretization of (GF). Our aim is to show that if f is smooth, and the step size is sufficiently small (as a function of the smoothness), then the conclusions for (GF) transfer to (GD). Throughout this section, we assume that f is twice continuously differentiable and β-smooth.

Some of the results in this section pertain to a single step of (GD), so we use the following notation:

$$x^+ := x - h\,\nabla f(x)\,.$$

The first step is to establish the descent property.

Lemma 3.1 (descent lemma)**.** For any β-smooth $f : \mathbb{R}^d \to \mathbb{R}$, if $h \le 1/\beta$, then

$$f(x^+) - f(x) \le -\frac{h}{2}\,\|\nabla f(x)\|^2\,.$$

Proof. By the smoothness inequality (1.7),

$$f(x^+) \le f(x) + \langle \nabla f(x), x^+ - x\rangle + \frac{\beta}{2}\,\|x^+ - x\|^2 = f(x) - h\,\|\nabla f(x)\|^2 + \frac{\beta h^2}{2}\,\|\nabla f(x)\|^2\,.$$

If $h \le 1/\beta$, then $-h\,(1 - \beta h/2) \le -h/2$. □

It is natural to state the subsequent results in terms of the following parameter.

Definition 3.2. Let f be α-convex and β-smooth. Then, the **condition number** of f is defined to be the ratio $\kappa := \beta/\alpha \ge 1$.

When f is quadratic, $f(x) = \frac{1}{2}\,\langle x, A\,x\rangle$ with A symmetric, then α, β correspond to the minimum and maximum eigenvalues of A respectively, and the ratio β/α is known in numerical linear algebra as the condition number of the matrix A. Thus, Definition 3.2 provides a natural generalization of this notion. With this definition in hand, we now arrive at our first convergence result for (GD).

Theorem 3.3 (contraction of GD)**.** Let f be α-convex and β-smooth. For all $x, y \in \mathbb{R}^d$ and step size $h \le 1/\beta$,

$$\|y^+ - x^+\| \le (1 - \alpha h)^{1/2}\,\|y - x\|\,.$$

Proof. Expanding the square,

$$\|y^+ - x^+\|^2 = \|y - x\|^2 - 2h\,\langle y - x, \nabla f(y) - \nabla f(x)\rangle + h^2\,\|\nabla f(y) - \nabla f(x)\|^2\,.$$

By (3.4) in Exercise 3.1 below, for $h \le 1/\beta$ and from (1.5) we have

$$\|y^+ - x^+\|^2 \le \|y - x\|^2 - h\,\langle \nabla y - x, \nabla f(y) - \nabla f(x)\rangle \le (1 - \alpha h)\,\|y - x\|^2\,. \qquad \square$$

In particular, if we take $y = x_\star$, $h = 1/\beta$, and iterate, it yields

$$\|x_N - x_\star\| \le \big(1 - \frac{1}{\kappa}\big)^{N/2}\,\|x_0 - x_\star\| \le \exp\big(-\frac{N}{2\kappa}\big)\,\|x_0 - x_\star\|\,.$$

Thus, to obtain $\|x_N - x_\star\| \le \varepsilon$, it suffices to take $N \ge 2\kappa\log(\|x_0 - x_\star\|/\varepsilon)$.

The essence of these proofs is that the first-order term (scaling as h) replicates the continuous-time calculation, and we must apply smoothness in an appropriate way to control the second-order term (scaling as h^2). In the above proof, note that if we naïvely use Lipschitzness of the gradient (1.8) to control the second-order term, it leads to the suboptimal choice of step size $h = 1/(\beta\kappa)$, and a contraction factor of $(1 - 1/\kappa^2)^{1/2}$. To obtain $\|x_N - x_\star\| \le \varepsilon$, we would then have the estimate $N \ge 2\kappa^2\log(\|x_0 - x_\star\|/\varepsilon)$, which is substantially worse. In conclusion, a bit of finesse is necessary. (In fact, Theorem 3.3 can also be improved, and the sharp rate is derived in Exercise 3.2.)

Next, we turn toward the analogue of Theorem 2.4.

Theorem 3.4 (convergence of GD in function value)**.** Let f be α-convex and β-smooth. For any step size $h \le 1/\beta$,

$$\|x^+ - x_\star\|^2 \le (1 - \alpha h)\,\|x - x_\star\|^2 - 2h\,(f(x^+) - f_\star)\,. \tag{3.1}$$

Therefore,

$$f(x_N) - f_\star \le \frac{\alpha}{2\,\{(1 - \alpha h)^{-N} - 1\}}\,\|x_0 - x_\star\|^2\,. \tag{3.2}$$

When $\alpha = 0$, the right-hand side should be interpreted as its limiting value as $\alpha \to 0$, namely, $\frac{1}{2Nh}\,\|x_0 - x_\star\|^2$.

Proof. Expanding the square and applying convexity via (1.4),

$$\begin{aligned}\|x^+ - x_\star\|^2 &= \|x - x_\star\|^2 - 2h\,\langle \nabla f(x), x - x_\star\rangle + h^2\,\|\nabla f(x)\|^2 \\ &\le (1 - \alpha h)\,\|x - x_\star\|^2 - 2h\,(f(x) - f_\star) + h^2\,\|\nabla f(x)\|^2\,.\end{aligned}$$

For $h \le 1/\beta$, the descent lemma (Lemma 3.1) now implies (3.1).

The proof of (3.2), based on iterating the recursive inequality (3.1), is justified after Lemma 3.5 below. □

We remark for later use that the proof of (3.1) goes through even if we replace $x_\star$ with any other point $z \in \mathbb{R}^d$, i.e.,

$$\|x^+ - z\|^2 \le (1 - \alpha h)\, \|x - z\|^2 - 2h\,(f(x^+) - f(z))\,, \qquad \text{for all } z \in \mathbb{R}^d\,. \tag{3.3}$$

Iterating (3.1) is a matter of unrolling the recursion, but in order to maintain the analogy with continuous time, we refer to the lemma below as "discrete Grönwall".

Lemma 3.5 (discrete Grönwall)**.** Suppose that for some $A > 0$,

$$u_{n+1} \le A u_n + B_n \qquad \text{for } n = 0, 1, \dots, N-1\,.$$

Then,

$$u_N \le A^N u_0 + \sum_{n=1}^{N} A^{N-n} B_{n-1}\,.$$

Proof. We multiply the given inequality by $A^{-(n+1)}$ to form a telescoping sum:

$$A^{-N}\, u_N - u_0 = \sum_{n=0}^{N-1} A^{-(n+1)}\,(u_{n+1} - A u_n) \le \sum_{n=0}^{N-1} A^{-(n+1)}\, B_n\,.$$

Rearrange to obtain the result. □

To complete the proof of Theorem 3.4, we apply Lemma 3.5 with $u_n = \|x_n - x_\star\|^2$, $A = 1 - \alpha h$, and $B_n = -2h\,(f(x_{n+1}) - f_\star)$, yielding

$$2h \sum_{n=1}^{N} (1 - \alpha h)^{N-n}\,(f(x_n) - f_\star) \le (1 - \alpha h)^N\, \|x_0 - x_\star\|^2\,.$$

For $h \le 1/\beta$, the descent lemma (Lemma 3.1) implies $f(x_n) - f_\star \ge f(x_N) - f_\star$, so

$$f(x_N) - f_\star \le \frac{\|x_0 - x_\star\|^2}{2h \sum_{n=1}^{N} (1 - \alpha h)^{-n}} = \frac{\alpha\, \|x_0 - x_\star\|^2}{2\,\{(1 - \alpha h)^{-N} - 1\}}\,.$$

In particular, let us set $h = 1/\beta$. For $\alpha > 0$ it yields

$$f(x_N) - f_\star \le \frac{\alpha \|x_0 - x_\star\|^2}{2\,\{(1 - 1/\kappa)^{-N} - 1\}}$$

and for $\alpha = 0$, it yields

$$f(x_N) - f_\star \le \frac{\beta \|x_0 - x_\star\|^2}{2N}\,.$$

The proof of convergence under (PŁ) is strikingly easy.

Theorem 3.6 (convergence of GD under PŁ)**.** Let f be β-smooth and satisfy (PŁ) with constant $\alpha > 0$. Then, for all $h \le 1/\beta$,

$$f(x_N) - f_\star \le (1 - \alpha h)^N\,(f(x_0) - f_\star)\,.$$

Proof. By the descent lemma (Lemma 3.1) and (PŁ),

$$\begin{aligned} f(x^+) - f_\star = f(x) - f_\star + f(x^+) - f(x) &\le f(x) - f_\star - \frac{h}{2}\,\|\nabla f(x)\|^2 \\ &\le (1 - \alpha h)\,(f(x) - f_\star)\,. \end{aligned}$$ □

Finally, we present the result for obtaining a stationary point.

Theorem 3.7. Let f be β-smooth and $h \le 1/\beta$. Then,

$$\min_{n=0,1,\dots,N-1} \|\nabla f(x_n)\| \le \sqrt{\frac{2\,(f(x_0) - f_\star)}{Nh}}\,.$$

Proof. Telescope the descent lemma (Lemma 3.1):

$$\frac{h}{2N} \sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2 \le \frac{1}{N} \sum_{n=0}^{N-1} (f(x_n) - f(x_{n+1})) \le \frac{f(x_0) - f_\star}{N}\,.$$ □

We summarize the results for GD in Table 1.

Assumptions	**Criterion**	**Iterations**
α-convex, β-smooth	$\lVert x_N - x_\star\rVert \le \varepsilon$	$O(\kappa \log(R/\varepsilon))$
α-convex, β-smooth	$f(x_N) - f_\star \le \varepsilon$	$O(\kappa \log(\alpha R^2/\varepsilon))$
convex, β-smooth	$f(x_N) - f_\star \le \varepsilon$	$O(\beta R^2/\varepsilon)$
α-(PŁ), β-smooth	$f(x_N) - f_\star \le \varepsilon$	$O(\kappa \log(\Delta_0/\varepsilon))$
β-smooth	$\min_{n=0,1,\dots,N-1} \lVert \nabla f(x_n)\rVert \le \varepsilon$	$O(\beta \Delta_0/\varepsilon^2)$

Table 1: Rates for GD with step size $1/\beta$. Here, $R \coloneqq \lVert x_0 - x_\star\rVert$ and $\Delta_0 \coloneqq f(x_0) - f_\star$.

Example 3.8 (logistic regression revisited)**.** For fun, let us revisit logistic regression (Example 1.2) from a statistical lens. For concreteness, we consider Gaussian design, $X_i \overset{\text{i.i.d.}}{\sim} \mathsf{normal}(0, I)$, and assume that the data is generated from the model with a true parameter $\theta^\star$. Let $\widehat{\mathcal{L}}$ denote the MLE objective, let $\mathcal{L} \coloneqq \mathbb{E}\,\widehat{\mathcal{L}}$ denote the population risk, and let $R \coloneqq \lVert \theta^\star \rVert \ge 1$. The state-of-the-art result [CLM24] shows that if $n \gtrsim Rd$ for a sufficiently large implied constant, $\widehat{\theta}_{\mathrm{MLE}}$ exists with probability $\ge 1 - \exp(-d)$ and satisfies the optimal risk bound $\mathcal{L}(\widehat{\theta}_{\mathrm{MLE}}) - \mathcal{L}(\theta^\star) \lesssim d/n$.

In practice, we cannot compute $\widehat{\theta}_{\mathrm{MLE}}$ exactly, so we use optimization. From [CLM24], any estimator $\widehat{\theta}$ satisfying $\widehat{\mathcal{L}}(\widehat{\theta}) - \widehat{\mathcal{L}}(\widehat{\theta}_{\mathrm{MLE}}) \lesssim d/n$ satisfies the same statistical risk bound as $\widehat{\theta}_{\mathrm{MLE}}$, up to a universal constant. We take $\widehat{\theta} = \widehat{\theta}_{\mathrm{GD}}$ to be the output of GD after N steps, and check how large N must be in order for this to hold. As justified in Exercise 3.3, we can expect an iteration complexity of $N \asymp R^2 n/d$.

Bibliographical notes

My understanding of Theorem 3.4 is again based on extensive discussions with Jason M. Altschuler, Adil Salim, Andre Wibisono, and Ashia Wilson.

Exercises

Exercise 3.1. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and β-smooth. Apply Lemma 3.1 to the function $y \mapsto f(y) - \langle \nabla f(x), y\rangle$ and observe that this function is minimized at x in order to prove

$$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2\beta}\,\lVert \nabla f(y) - \nabla f(x)\rVert^2\,.$$

From this, deduce that

$$\lVert \nabla f(y) - \nabla f(x)\rVert^2 \le \beta\,\langle \nabla f(y) - \nabla f(x), y - x\rangle\,. \tag{3.4}$$

Finally, use the Cauchy–Schwarz inequality to show that ∇f is β-Lipschitz, i.e., that (1.8) holds. Note that this proof that convexity and β-smoothness together imply (1.8) does not require f to be twice differentiable.

Exercise 3.2. Let f be α-convex and β-smooth. Let $T := \mathrm{id} - h\,\nabla f$ denote the one-step GD mapping. By the fundamental theorem of calculus,

$$\begin{aligned}\|y^+ - x^+\| = \|T(y) - T(x)\| &= \Big\|\int_0^1 \nabla T((1-t)\,x + t\,y)\,(y-x)\,\mathrm{d}t\Big\| \\ &\le \Big(\int_0^1 \|\nabla T((1-t)\,x + ty)\|_{\mathrm{op}}\,\mathrm{d}t\Big)\,\|y - x\|\,.\end{aligned}$$

For any $z \in \mathbb{R}^d$, bound the eigenvalues of $\nabla T(z)$ and show that the choice of step size h which minimizes the bound on $\|\nabla T(z)\|_{\mathrm{op}}$ is $h = 2/(\alpha + \beta)$. Deduce the sharp rate

$$\|y^+ - x^+\| \le \frac{\kappa - 1}{\kappa + 1}\,\|y - x\|\,.$$

Note that for large κ, the contraction factor is approximately $\exp(-2/\kappa)$, so this improves upon the iteration complexity implied by Theorem 3.3 by a factor of nearly 4.

Exercise 3.3. What does Theorem 3.4 imply for logistic regression (Example 1.2)? In the setting of Example 3.8, use the fact that $\lambda_{\max}(\frac{1}{n}\sum_{i=1}^n X_i X_i^\mathsf{T}) \lesssim 1$ with high probability[3] to justify the claimed $R^2 n/d$ iteration complexity.

4 [1/23] Lower bounds for smooth optimization

The goal of this section is to establish lower complexity bounds for convex smooth optimization. Refer to §1.1 for a conceptual first discussion of the oracle model.

Before doing so, we present some reductions between the convex and strongly convex settings which save us some effort.

4.1 Reductions between the convex and strongly convex settings

For brevity, let us say that an algorithm *successfully optimizes* a function class $\mathscr{F}$ in $\phi(\mathscr{F}, R, \varepsilon)$ iterations if, given any $f \in \mathscr{F}$ and $x_0 \in \mathbb{R}^d$ with $\|x_0 - x_\star\| \le R$, it outputs x with $f(x) - f_\star \le \varepsilon$ using no more than $\phi(\mathscr{F}, R, \varepsilon)$ queries to a first-order oracle for f.

[3] This is a standard fact about the Wishart distribution; see, e.g., [Ver18, Theorem 4.4.5].

Lemma 4.1. Assume there is an algorithm which successfully optimizes the class of convex and β-smooth functions in $\phi(\beta R^2/\varepsilon)$ iterations.

Then, there is an explicit algorithm which successfully optimizes the class of α-convex and β-smooth functions in $O(\phi(8\kappa)\log(\alpha R^2/\varepsilon))$ iterations.

Proof. Let f be α-strongly convex and β-smooth, and apply the given algorithm to f to obtain a new point x_1 with tolerance ε_1. By (QG), we have

$$\frac{\alpha}{2}\|x_1 - x_\star\|^2 \le f(x_1) - f_\star \le \varepsilon_1\,.$$

Set $\varepsilon_1 = \alpha R^2/8$, so that

$$\|x_1 - x_\star\| \le \frac{1}{2}R = \frac{1}{2}\|x_0 - x_\star\|\,. \tag{4.1}$$

For $\kappa := \beta/\alpha$, this requires $\phi(8\kappa)$ iterations. From (4.1), if we now repeat this procedure $O(\log(\alpha R^2/\varepsilon))$ times, we can reach a point $\tilde{x}$ satisfying $\tilde{R} := \|\tilde{x} - x_\star\| \le \sqrt{\varepsilon/\alpha}$. Finally, apply the given algorithm one more time starting from $\tilde{x}$ with target accuracy ε to obtain a point x with $f(x) - f_\star \le \varepsilon$. The complexity of this final step is $\phi(\beta\tilde{R}^2/\varepsilon) = \phi(\kappa)$. □

For example, if we combine the $\alpha = 0$ case of Theorem 3.4 with Lemma 4.1, taking $\phi(x) = O(x)$, we recover the $\alpha > 0$ case of Theorem 3.4, up to constants.

Lemma 4.2. Assume there is an algorithm which successfully optimizes the class of α-convex and β-smooth functions in $\phi(\kappa)\log(\alpha R^2/\varepsilon)$ iterations.

Then, there is an explicit algorithm which successfully optimizes the class of convex and β-smooth functions in $O(\phi(2\beta R^2/\varepsilon))$ iterations.

Proof. Let f be convex and β-smooth. We apply the given algorithm to the regularized function $f_\delta := f + \frac{\delta}{2}\|\cdot - x_0\|^2$, obtaining a point x such that $f_\delta(x) \le \min f_\delta + \varepsilon/2$. If $x_{\delta,\star}$ denotes the minimizer of f_δ, then

$$f(x) \le f_\delta(x) \le f_\delta(x_{\delta,\star}) + \frac{\varepsilon}{2} \le f_\delta(x_\star) + \frac{\varepsilon}{2} = f_\star + \frac{\delta}{2}\|x_0 - x_\star\|^2 + \frac{\varepsilon}{2}\,.$$

We now set $\delta = \varepsilon/R^2$, so that $f(x) - f_\star \le \varepsilon$.

It remains to estimate the complexity. We first note that $f_\delta(x_{\delta,\star}) \le f_\delta(x_\star)$ implies $\|x_0 - x_{\star,\delta}\| \le \|x_0 - x_\star\|$, so the initial distance to the minimizer of f_δ is also bounded by R. We can assume that $\varepsilon \le \beta R^2$ (or else the minimization problem is trivial). Then, the smoothness of f_δ is bounded by $\beta + \delta \le 2\beta$, and the condition number of f_δ is bounded by $2\beta R^2/\varepsilon$. Substitute these quantities into the complexity of the given algorithm. □

Thus, the $\alpha > 0$ case of Theorem 3.4 and Lemma 4.2 recover the $\alpha = 0$ case of Theorem 3.4 up to constants.

Taken together, Lemma 4.1 and Lemma 4.2 show that the 0-convex and strongly convex settings are essentially equivalent to each other, in that an optimal method for one class yields an optimal method for the other class. Thus, we now aim to address the following question: what is the smallest possible $\phi(\cdot)$?

4.2 Lower bounds

According to the discussion in §1.1, establishing a lower complexity bound requires showing that *any* algorithm which interacts with the first-order oracle using at most a prescribed number of queries cannot have performance better than the lower bound. Actually, although this is possible (see [NY83]), it is not especially easy. It was shown by Nesterov in an earlier edition of [Nes18] that by imposing natural restrictions on the class of algorithms under consideration, it is possible to establish the lower bounds in a more transparent way. Accordingly, his approach has become standard in the field, and it is the approach we adopt here as well. It does, however, have the drawback of not applying to general query algorithms; for example, it does not apply against randomized algorithms.

The class of algorithms we consider is the following one.

Definition 4.3. An algorithm is called a **gradient span** algorithm if it deterministically generates a sequence of points $\{x_n\}_{n\in\mathbb{N}}$ such that for all $n \in \mathbb{N}$,

$$x_{n+1} \in x_0 + \operatorname{span}\{\nabla f(x_0), \dots, \nabla f(x_n)\}\,.$$

For example, GD is a gradient span algorithm. On the basis of this assumption, we now establish the following result; recall the asymptotic notation $\gtrsim$, which only hides a universal constant.

Theorem 4.4 (lower bound for convex, smooth minimization)**.** For any $1 \le N \le \frac{d-1}{2}$, $\beta > 0$, and $x_0 \in \mathbb{R}^d$, there exists a convex and β-smooth function $f : \mathbb{R}^d \to \mathbb{R}$ such that for any gradient span algorithm,

$$f(x_N) - f_\star \gtrsim \frac{\beta\,\|x_0 - x_\star\|^2}{N^2}\,.$$

In other words, in order to obtain $f(x_N) - f_\star \le \varepsilon$, the number of iterations must satisfy

$$N \gtrsim \sqrt{\frac{\beta\,\|x_0 - x_\star\|^2}{\varepsilon}}\,.$$

Before proving this result, we observe that by applying Lemma 4.2 with $\phi(x) \asymp \sqrt{x}$, it yields the following corollary.

Theorem 4.5 (lower bound for strongly convex, smooth minimization)**.** For any $0 < \alpha < \beta$, any $\varepsilon > 0$, any d sufficiently large, and any $x_0 \in \mathbb{R}^d$, there exists an α-convex and β-smooth function $f : \mathbb{R}^d \to \mathbb{R}$ such that for any gradient span algorithm, in order to obtain $f(x_N) - f_\star \le \varepsilon$, the number of iterations must satisfy

$$N \gtrsim \sqrt{\kappa} \log \frac{\alpha \|x_0 - x_\star\|^2}{\varepsilon} .$$

Proof of Theorem 4.4. By translating the problem, we may assume $x_0 = 0$. The construction is based on the following function:

$$f_n : \mathbb{R}^d \to \mathbb{R}, \qquad f_n(x) := \frac{\beta}{4} \left\{ \frac{1}{2} \left(x[1]^2 + \sum_{k=1}^{n-1} (x[k] - x[k+1])^2 + x[n]^2 \right) - x[1] \right\} .$$

For any $v \in \mathbb{R}^d$,

$$\langle v, \nabla^2 f_n(x)\, v \rangle = \frac{\beta}{4} \left(v[1]^2 + \sum_{k=1}^{n-1} (v[k] - v[k+1])^2 + v[n]^2 \right) \le \beta \|v\|^2 ,$$

so each f_n is convex and β-smooth.

We prove by induction that when we apply a gradient span algorithm to f_d, the n-th iterate x_n belongs to the subspace

$$\mathcal{V}_n := \{x \in \mathbb{R}^d : x[k] = 0 \text{ for all } k = n+1, \dots, d\} .$$

Clearly, $x_0 \in \mathcal{V}_0$. Inductively, suppose that $x_k \in \mathcal{V}_k$ for all $k \le n$. Then,

$$\nabla f_d(x_k) = \frac{\beta}{4} \left(x_k[1]\, e_1 + \sum_{j=1}^{k} (x_k[j] - x_k[j+1])\, (e_j - e_{j+1}) \right) - \frac{\beta}{4} e_1 \in \mathcal{V}_{k+1} ,$$

hence

$$x_{n+1} \in \operatorname{span}\{\nabla f_d(x_0), \dots, \nabla f_d(x_n)\} \subseteq \mathcal{V}_{n+1} .$$

This completes the induction. Also, since $f_N = f_d$ on $\mathcal{V}_N$, it follows that

$$f_d(x_N) = f_N(x_N) \ge (f_N)_\star .$$

The next step is to estimate $(f_n)_\star := \min f_n$ for all n. By setting the gradient to zero, $\nabla f_n(x_{n,\star}) = 0$, we obtain the following system of equations:

$$\begin{aligned} 2x_{n,\star}[1] - x_{n,\star}[2] &= 1\,, \\ x_{n,\star}[k-1] - 2x_{n,\star}[k] + x_{n,\star}[k+1] &= 0\,, \qquad \text{for } k = 1, \dots, n\,, \\ -x_{n,\star}[n-1] + 2x_{n,\star}[n] &= 0\,. \end{aligned}$$

The solution is $x_{n,\star}[k] = 1 - \frac{k}{n+1}$ for all $k \in [n]$. Writing $f_n(x) = \frac{\beta}{4}\,\{\frac{1}{2}\,\langle x, A_n\, x\rangle - \langle e_1, x\rangle\}$, the system above reads $A_n x_{n,\star} = e_1$, hence

$$(f_n)_\star = f_n(x_{n,\star}) = -\frac{\beta}{8}\,\langle e_1, x_{n,\star}\rangle = -\frac{\beta}{8}\,\big(1 - \frac{1}{n+1}\big)\,.$$

Moreover, $\|x_0 - x_{n,\star}\|^2 = \|x_{n,\star}\|^2 \le n$. Finally, it yields

$$\begin{aligned} f_d(x_N) - (f_d)_\star \ge (f_N)_\star - (f_d)_\star &= \frac{\beta}{8}\,\big(\frac{1}{N+1} - \frac{1}{d+1}\big) \\ &\ge \frac{\beta\,\|x_0 - x_{d,\star}\|^2}{8d}\,\big(\frac{1}{N+1} - \frac{1}{d+1}\big)\,. \end{aligned}$$

Choosing $d \asymp N$, e.g., $d = 2N+1$, yields the stated lower bound. □

Notably, the iteration complexity lower bounds Theorem 4.4 and Theorem 4.5 are smaller than the bounds attained by GD in Theorem 3.4 by a square root. As developed in the next sections, in fact the lower bounds are tight and GD is suboptimal.

We make two further remarks. First, it is perhaps surprising that the lower bound construction is a *quadratic* function; in some sense, quadratics are the hardest convex and smooth functions to optimize. Second, the lower bound requires the ambient dimension to be larger than the iteration count; this is crucial for the proof technique, which relies on the algorithm discovering one new dimension per iteration. This turns out to be fundamental because there are better methods in low dimension, for quadratics and even for general convex functions.

Exercises

Exercise 4.1. In the setting of Theorem 4.4 and using the same construction as in the proof, show that $\|x_N - x_\star\|^2 \gtrsim \|x_0 - x_\star\|^2$. In other words, in the 0-convex case, it is not possible to make progress in the sense of distance to the minimizer by more than a constant factor.

Exercise 4.2. We used the reductions from §4.1 to reduce the strongly convex lower bound to the 0-convex lower bound for the sake of brevity, but it is of course possible to develop the strongly convex lower bound directly. Consider the function

$$f : \mathbb{R}^\infty \to \mathbb{R}, \qquad f(x) := \frac{\beta - \alpha}{8} \Big\{ x[1]^2 + \sum_{n=1}^{\infty} (x[n] - x[n+1])^2 - 2x[1] \Big\} + \frac{\alpha}{2} \|x\|^2 .$$

By adapting the proof of Theorem 4.4, show that any gradient span algorithm satisfies

$$f(x_N) - f_\star \ge \frac{\alpha}{2} \|x_N - x_\star\|^2 \ge \frac{\alpha}{2} \Big(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \Big)^{2N} \|x_0 - x_\star\|^2 .$$

5 [1/28–1/30] Acceleration

We now show that the lower bounds of Theorem 4.4 and Theorem 4.5 can be attained via algorithms which improve upon GD. This is known as the *acceleration* phenomenon in optimization. We begin with the quadratic case.

5.1 Quadratic case: the conjugate gradient method

In this section, the objective function is quadratic:

$$f : \mathbb{R}^d \to \mathbb{R}, \qquad f(x) = \frac{1}{2} \langle x, A\, x \rangle - \langle b, x \rangle ,$$

where A is a symmetric matrix, $A \succ 0$. Note also that minimizing f corresponds to solving the system of equations $Ax_\star = b$. We now introduce the *conjugate gradient* method.

The method is succinctly described as follows:

$$x_{n+1} := \arg\min \big\{ f(x) \bigm| x \in x_0 + \operatorname{span}\{\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_n)\} \big\} . \tag{CG}$$

This scheme is very natural in light of the definition of a gradient span algorithm (Definition 4.3) that we encountered for the lower bounds. However, it is not yet clear that (CG) can be implemented cheaply. Using the fact that f is quadratic, our aim is to show that (CG) can be rewritten as a simple iteration that uses one gradient query per step.

As is usually the case in linear algebra, instead of working with the set of vectors $\{\nabla f(x_0), \nabla f(x_1), \dots, \nabla f(x_n)\}$, it is more convenient to work with an *orthogonal* set $\{p_0, p_1, \dots, p_n\}$. Here, orthogonality is with respect to the inner product $\langle \cdot, \cdot \rangle_A$, i.e., we will require $\langle p_i, A\, p_j \rangle = 0$ for all $i \ne j$. We start with $p_0 := \nabla f(x_0)$, and we write $\mathcal{K}_n := \operatorname{span}\{p_0, p_1, \dots, p_n\}$. We must address the following two questions:

- Given $\mathcal{K}_n$ and x_n, how can we compute $x_{n+1} = \arg\min_{x_0+\mathcal{K}_n} f$?
- Given $\mathcal{K}_n$ and $\nabla f(x_{n+1})$, how can we compute p_{n+1} and thus $\mathcal{K}_{n+1}$?

For the first question, we may assume inductively that $x_n = \arg\min_{x_0+\mathcal{K}_{n-1}} f$, which means that $\langle \nabla f(x_n), p_k \rangle = 0$ for all $k < n$. The next point is taken to be $x_{n+1} = x_n + h_n p_n$, chosen so that $\langle \nabla f(x_{n+1}), p_k \rangle = 0$ for all $k \le n$. Since ∇f is linear,

$$\langle \nabla f(x_{n+1}), p_k \rangle = \langle \nabla f(x_n) + h_n\, A p_n, p_k \rangle\,.$$

For $k < n$, this equals zero by the inductive hypothesis on x_n, and the orthogonality of $\{p_0, p_1, \dots, p_n\}$. We choose h_n to ensure that this equals zero for $k = n$ too:

$$h_n = -\frac{\langle \nabla f(x_n), p_n \rangle}{\|p_n\|_A^2}\,.$$

For the second question, we want to compute the Gram–Schmidt orthogonalization of $\nabla f(x_{n+1})$ w.r.t. $\{p_0, p_1, \dots, p_n\}$ in the $\langle \cdot, \cdot \rangle_A$ inner product. We claim that $\nabla f(x_{n+1})$ is already A-orthogonal to p_k for $k < n$, so that

$$p_{n+1} = \nabla f(x_{n+1}) - \langle \nabla f(x_{n+1}), p_n \rangle_A \frac{p_n}{\|p_n\|_A^2}\,. \tag{5.1}$$

To justify this, we show that for $k < n$, $\boxed{Ap_k \in \mathcal{K}_{k+1}}$, hence

$$\langle \nabla f(x_{n+1}), p_k \rangle_A = \langle \nabla f(x_{n+1}), Ap_k \rangle = 0$$

using the fact shown above that $\nabla f(x_{n+1})$ is orthogonal (in the usual inner product) to $\mathcal{K}_n$. Finally, the boxed equation is shown through the following lemma.

Lemma 5.1. For all $n \in \mathbb{N}$,

$$\mathcal{K}_n = \operatorname{span}\{p_0, Ap_0, \dots, A^n p_0\}\,.$$

Proof. We proceed via induction, where the case $n = 0$ is obvious. Assuming it holds at iteration n, let us show that $p_{n+1} \in \widetilde{\mathcal{K}}_{n+1} := \operatorname{span}\{p_0, Ap_0, \dots, A^{n+1} p_0\}$. By (5.1), it suffices to show that $\nabla f(x_{n+1}) \in \widetilde{\mathcal{K}}_{n+1}$. However, as discussed above, $\nabla f(x_{n+1}) = \nabla f(x_n) + h_n\, Ap_n = p_0 + h_0\, Ap_0 + \cdots + h_n\, Ap_n \in \widetilde{\mathcal{K}}_{n+1}$.

Conversely, we must show that $A^{n+1} p_0 \in \mathcal{K}_{n+1}$. Since $A^n p_0 \in \mathcal{K}_n$, we can write $A^n p_0 = \sum_{k=0}^n c_k p_k$, thus $A^{n+1} p_0 = \sum_{k=0}^n c_k\, Ap_k$. By the inductive hypothesis, each Ap_k for $k < n$ belongs to $\mathcal{K}_n$, so it suffices to have $Ap_n \in \mathcal{K}_{n+1}$. However, we can observe that $Ap_n = h_n^{-1}\,(\nabla f(x_{n+1}) - \nabla f(x_n)) \in \mathcal{K}_{n+1}$ by (5.1). □

Definition 5.2. The subspaces $\{\mathcal{K}_n\}_{n\in\mathbb{N}}$ are called **Krylov subspaces**.

Finally, let us write the iterations in a form which is convenient for implementation. Note first that $\langle \nabla f(x_n), \nabla f(x_{n+1})\rangle = 0$ (indeed, $\nabla f(x_{n+1})$ is orthogonal to all of $\mathcal{K}_n$). So,

$$\frac{\langle \nabla f(x_{n+1}), p_n\rangle_A}{\|p_n\|_A^2} = \frac{\langle \nabla f(x_{n+1}), \nabla f(x_{n+1}) - \nabla f(x_n)\rangle}{h_n\, \|p_n\|_A^2} = -\frac{\|\nabla f(x_{n+1})\|^2}{\langle \nabla f(x_n), p_n\rangle}$$

and $\|\nabla f(x_n)\|^2 = \langle \nabla f(x_n), \nabla f(x_n)\rangle = \langle \nabla f(x_n), p_n\rangle$ using (5.1) and the fact that $\nabla f(x_n)$ is orthogonal to $\mathcal{K}_{n-1}$. This yields the following iteration, where we write $r_n := Ax_n - b = \nabla f(x_n)$ for the residual.

$$x_{n+1} = x_n - \frac{\|r_n\|^2}{\langle p_n, A\,p_n\rangle}\, p_n\,, \quad r_{n+1} = r_n - \frac{\|r_n\|^2}{\langle p_n, A\,p_n\rangle}\, Ap_n\,, \quad p_{n+1} = r_{n+1} + \frac{\|r_{n+1}\|^2}{\|r_n\|^2}\, p_n\,.$$

This algorithm requires one matrix-vector multiplication per iteration, namely, the computation of Ap_n.

Note that if $p_{n+1} = 0$, then $\nabla f(x_{n+1}) \in \mathcal{K}_n$, yet $\nabla f(x_{n+1}) \perp \mathcal{K}_n$ and thus $\nabla f(x_{n+1}) = 0$, $x_{n+1} = x_\star$. Since $p_{d+1} = 0$ (an orthogonal set in $\mathbb{R}^d$ cannot have more than d non-zero elements), we arrive at the following conclusion.

Theorem 5.3 (termination of CG)**.** The CG algorithm returns the exact minimizer in at most d iterations.

Let us now show that CG can find an approximate minimizer at the accelerated rate.

Theorem 5.4 (accelerated convergence for CG)**.** Let $0 \prec \alpha I \preceq A \preceq \beta I$. Then, CG outputs x_N satisfying $f(x_N) - f_\star \le \varepsilon$ in $N = O(\sqrt{\kappa}\log\frac{f(x_0)-f_\star}{\varepsilon})$ iterations.

Proof. By the descent lemma (Lemma 3.1) and the defining property of CG,

$$f(x_{n+1}) \le f\big(x_n - \frac{1}{\beta}\nabla f(x_n)\big) \le f(x_n) - \frac{1}{2\beta}\,\|\nabla f(x_n)\|^2\,,$$

so that

$$f(x_0) - f_\star \ge \frac{1}{2\beta}\sum_{n=0}^{N-1}\|\nabla f(x_n)\|^2\,.$$

On the other hand, since $\nabla f(x_n) \perp x_{k+1} - x_k$ for $k < n$,

$$f_\star - f(x_n) \ge \langle \nabla f(x_n), x_\star - x_n \rangle = \langle \nabla f(x_n), x_\star - x_0 \rangle \, .$$

If we sum these inequalities and use orthogonality of the gradients,

$$\begin{aligned} N\,(f(x_N) - f_\star) &\le \sum_{n=0}^{N-1} (f(x_n) - f_\star) \le \Big\langle \sum_{n=0}^{N-1} \nabla f(x_n), x_0 - x_\star \Big\rangle \le \Big\| \sum_{n=0}^{N-1} \nabla f(x_n) \Big\| \, \|x_0 - x_\star\| \\ &\le \Big(\sum_{n=0}^{N-1} \|\nabla f(x_n)\|^2 \Big)^{1/2} \sqrt{\frac{2\,(f(x_0) - f_\star)}{\alpha}} \le 2\sqrt{\kappa}\,(f(x_0) - f_\star) \, . \end{aligned}$$

Let N be such that $f(x_N) - f_\star \ge (f(x_0) - f_\star)/2$. The inequality above then implies that $N \le 4\sqrt{\kappa}$. Thus, every $4\sqrt{\kappa}$ iterations, the objective gap decreases by a factor of 2. □

By applying the restart strategy as in Lemma 4.1, one can also show an iteration complexity scaling with $\sqrt{\kappa}$ in the strongly convex case. However, we instead give a different proof in order to explain the classical link with polynomial approximation.

Due to Lemma 5.1, $x_N - x_0 \in \mathcal{K}_{N-1}$ can be written in the form $x_N - x_0 = \sum_{n=0}^{N-1} c_n A^n p_0$, so $x_N - x_\star = x_0 - x_\star + \sum_{n=0}^{N-1} c_n A^{n+1}\,(x_0 - x_\star) = P_N(A)\,(x_0 - x_\star)$ where P_N is a polynomial of degree at most N satisfying $P_N(0) = 1$. Conversely, if Q_N is any other degree-N polynomial with $Q_N(0) = 1$, then $\tilde{x}_N := x_0 + A^{-1}\,(Q_N(A) - I)\,p_0 \in x_0 + \mathcal{K}_{N-1}$ satisfies $\tilde{x}_N - x_\star = x_0 - x_\star + A^{-1}\,(Q_N(A) - I)\,p_0 = Q_N(A)\,(x_0 - x_\star)$.

This equivalence, together with the fact that the output x_N of CG minimizes f over $x_0 + \mathcal{K}_{N-1}$, shows that

$$f(x_N) - f_\star \le \frac{1}{2} \min\{\|Q_N(A)\,(x_0 - x_\star)\|_A^2 : Q_N \in \mathbb{R}_{\le N}[X]\,,\ Q_N(0) = 1\}\,,$$

where $\mathbb{R}_{\le N}[X]$ denotes the set of polynomials with real-valued coefficients and with degree at most N. Furthermore, since A and $Q_N(A)$ commute,

$$\|Q_N(A)\,(x_0 - x_\star)\|_A^2 \le \|Q_N(A)\|_{\mathrm{op}}^2 \, \|x_0 - x_\star\|_A^2 \le \big(\max_{[\lambda_{\min}(A), \lambda_{\max}(A)]} |Q_N|^2\big)\, \|x_0 - x_\star\|_A^2 \, .$$

We have arrived at the following result.

Lemma 5.5 (CG and polynomial approximation)**.** Assume that $0 \prec \alpha I \preceq A \preceq \beta I$. Then, the output x_N of CG satisfies

$$f(x_N) - f_\star \le \min\Big\{ \max_{\lambda \in [\alpha,\beta]} |Q_N(\lambda)|^2 : Q_N \in \mathbb{R}_{\le N}[X]\,,\ Q_N(0) = 1 \Big\}\,(f(x_0) - f_\star)\, .$$

Informally, this result states that CG performs as well as the best possible degree-N polynomial in A. To bound the rate of convergence of CG, it therefore remains to exhibit a judicious polynomial Q_N. This is accomplished by the family of Chebyshev polynomials, on which many volumes have been written.

Definition 5.6. The degree-n **Chebyshev polynomial** T_n is defined so that $\cos(n\theta) = T_n(\cos\theta)$ for all $\theta \in \mathbb{R}$.

It is not obvious at first glance that T_n is indeed a degree-n polynomial, but this can be established via trigonometric identities. The use of the Chebyshev polynomials to establish a rate of convergence for CG is explored in Exercise 5.1.

Here, we point out another interesting fact that arises from this connection. Recall from the proof of Lemma 5.5 that if we can compute $\tilde{x}_N := x_0 + A^{-1}\,(Q_N(A) - I)\,p_0$, then it incurs error at most $f(\tilde{x}_N) - f_\star \le (\max_{\lambda\in[\alpha,\beta]} |Q_N(\lambda)|^2)\,(f(x_0) - f_\star)$. In particular, rather than using CG, we can try to compute the polynomial $x \mapsto (Q_N(x) - 1)/x$ directly, where Q_N is the polynomial in Exercise 5.1 which witnesses the fast convergence of CG. Although we omit the details, it is worth noting that the family of Chebyshev polynomials satisfies a so-called three-term recurrence:

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x)\,, \qquad x \in \mathbb{R}\,.$$

In fact, orthogonal families of polynomials usually do.[4] From an algorithmic standpoint, it leads to an optimization algorithm of the form

$$x_{n+1} = c_0\,Ax_n + c_1\,x_{n-1} + c_2\,b\,,$$

where $c_0, c_1, c_2 \in \mathbb{R}$ are fixed coefficients. Note that unlike GD, x_{n+1} depends on the previous *two* iterates. This is often referred to as *momentum*, and also forms the basis for acceleration for general convex functions.

Remark 5.7 (practicality of CG)**.** Solving the linear system $Ax = b$ via Gaussian elimination requires $O(d^3)$ operations and is numerically unstable, whereas for well-conditioned matrices A, CG returns an approximate solution in $\widetilde{O}(\sqrt{\kappa})$ iterations, each of which requires a matrix-vector multiplication. A matrix-vector multiplication requires $O(d^2)$ time in the worst case, but can be faster if A is sparse. In practice, CG is widely used, especially when combined with other strategies such as preconditioning.

[4] This arises in connection with second-order differential operators.

5.2 General case: continuous time

Although it does not follow the historical development of events, we begin our treatment of acceleration for general convex smooth functions in continuous time. As identified in [SBC16], the continuous-time ODE is

$$\begin{aligned} \dot x_t &= p_t\,, \\ \dot p_t &= -\nabla f(x_t) - \gamma_t p_t\,. \end{aligned} \tag{AGF}$$

We refer to (AGF) as the *accelerated gradient flow*, and the variable p_t admits the physical interpretation of momentum (for a particle with unit mass). The dynamics consists of two parts: the equations

$$\begin{aligned} \dot x_t &= p_t\,, \\ \dot p_t &= -\nabla f(x_t) \end{aligned}$$

are known as Hamilton's equations, and they are the standard first-order reformulation of Newton's law of motion $\ddot x_t = -\nabla f(x_t)$ with potential energy f. Hamilton's equations conserve the energy (or Hamiltonian) $H(x,p) \coloneqq f(x) + \frac{1}{2}\,\|p\|^2$, and this conservation property is perhaps undesirable for an optimization algorithm which seeks to minimize f. Thus, the second part of the dynamics, $\dot p_t = -\gamma_t p_t$ adds a dissipative *friction* force, where $\gamma_t \ge 0$ is a possibly time-varying coefficient of friction.

In the case where f is merely assumed to be convex, it turns out that the right choice of friction coefficient is $\gamma_t = 3/t$. This is mysterious at first sight and was obtained by taking the continuous-time limit of Nesterov's discrete algorithm in the next subsection. We begin with a convergence analysis in this setting. (Similar caveats as for §2 apply here; we assume that f is smooth, that it admits a minimizer $x_\star$, and that (AGF) is well-posed.)

Theorem 5.8 (convergence of AGF under convexity)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and let $(x_t)_{t\ge 0}$ evolve along AGF with $\gamma_t = 3/t$ and $p_0 = 0$. Then, for all $t \ge 0$,

$$f(x_t) - f_\star \le \frac{2\,\|x_0 - x_\star\|^2}{t^2}\,.$$

Proof. Consider the auxiliary point $z_t \coloneqq x_t + \frac{t}{2}\,p_t$, and the Lyapunov function

$$\mathscr{L}_t \coloneqq \frac{t^2}{2}\,(f(x_t) - f_\star) + \|z_t - x_\star\|^2\,.$$

The computation below shows that $\dot{\mathscr{L}}_t \le 0$, which implies the result. The choice of Lyapunov function is mysterious, so we partially demystify it after the proof.

Straightforward differentiation and convexity yield

$$\begin{aligned}\dot{\mathscr{L}}_t &= t\,(f(x_t) - f_\star) + \frac{t^2}{2}\,\langle \nabla f(x_t), p_t\rangle - t\,\langle \nabla f(x_t), z_t - x_\star\rangle \\ &= t\,(f(x_t) - f_\star) - t\,\langle \nabla f(x_t), x_t - x_\star\rangle \le 0\,. \qquad \square\end{aligned}$$

Although the Lyapunov function above appears fortuitous, it can be derived in a reasonably systematic manner; see Exercise 5.2. The strongly convex case is similar, and is left as Exercise 5.3.

Theorem 5.9 (convergence of AGF under strong convexity)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be α-convex and let $(x_t)_{t\ge 0}$ evolve along AGF with $\gamma_t = 2\sqrt{\alpha}$ and $p_0 = 0$. For all $t \ge 0$,

$$f(x_t) - f_\star \le 2\exp(-\sqrt{\alpha}\,t)\,(f(x_0) - f_\star)\,.$$

Recall that under convexity and α-convexity, the objective gap $f(x_t) - f_\star$ for GF converges at the rates $O(1/t)$ and $O(\exp(-2\alpha t))$ respectively. On the other hand, for AGF, the convergence happens at the rates $O(1/t^2)$ and $O(\exp(-\sqrt{\alpha}\,t))$ respectively. This is strongly suggestive of the square root factor speed-up, that is, *acceleration*. However, we caution that it is dangerous to deduce conclusions from continuous-time analysis alone. For example, we can run any ODE faster, which can make the continuous-time convergence rate arbitrarily fast; however, this does not translate into a better discrete-time algorithm, since speeding up time makes the ODE more unstable and therefore requires a smaller step size for discretization.

So how, then, can we discretize AGF? Part of the subtlety of acceleration is that not all discretizations work. For example, we could consider

$$\begin{aligned}x_{n+1} &\approx x_n + h\,p_{n+1}\,,\\ p_{n+1} &\approx p_n - h\,\nabla f(x_n) - \gamma_n h\,p_n\end{aligned}$$

which is equivalent to the update

$$x_{n+1} = x_n - h^2\,\nabla f(x_n) + (1 - \gamma_n h)\,(x_n - x_{n-1})\,.$$

Or, if we do not presume to know the coefficients for the discrete-time scheme in advance, we could write the update as

$$x_{n+1} = x_n - \eta_n\,\nabla f(x_n) + \theta_n\,(x_n - x_{n-1})\,.$$

In other words, we take a gradient step and then apply momentum. This is known as Polyak's heavy ball method, and although it can be tuned to converge at the rate of CG for quadratic objectives, this same tuning leads to divergence for general convex functions [LRP16]. On the other hand, the optimal method in the next subsection can be written in the form

$$x_{n+1} = x_n + \theta_n\,(x_n - x_{n-1}) - \eta_n\,\nabla f\big(x_n + \theta_n\,(x_n - x_{n-1})\big)\,.$$

In other words, we add momentum and then take a gradient step.

5.3 General case: discrete time

The acceleration phenomenon is undoubtedly one of the most elusive and fascinating aspects of optimization, so it is no surprise that it has been explored through many different angles over the course of countless research papers. At this junction, we must choose how to present the method and in what level of detail.

Having explored acceleration carefully in the quadratic case and in continuous time, here we follow the expedient route by giving perhaps the most direct and shortest proof, at the cost of generality and intuition.[5]

We analyze the following method with $x_{-1} = x_0$:

$$x_{n+1} \coloneqq x_n + \theta_n\,(x_n - x_{n-1}) - \frac{1}{\beta}\,\nabla f\big(x_n + \theta_n\,(x_n - x_{n-1})\big)\,. \tag{AGD}$$

Theorem 5.10 (convergence of AGD)**.** Let f be convex and β-smooth. Define the sequence: $\lambda_0 \coloneqq 0$ and $\lambda_{n+1} \coloneqq \frac{1}{2}\,(1 + \sqrt{1 + 4\lambda_n^2})$ for $n \in \mathbb{N}$. Set $\theta_n \coloneqq (\lambda_n - 1)/\lambda_{n+1}$. Then, AGD satisfies

$$f(x_N) - f_\star \le \frac{2\beta\,\|x_0 - x^\star\|^2}{N^2}\,.$$

Proof. Let $y_n \coloneqq x_n + \theta_n\,(x_n - x_{n-1})$, so that $x_{n+1} = y_n - \frac{1}{\beta}\,\nabla f(y_n)$. Recall from (3.3) that for any $z \in \mathbb{R}^d$, it holds that

$$\|x_{n+1} - z\|^2 \le \|y_n - z\|^2 - \frac{2}{\beta}\,(f(x_{n+1}) - f(z))\,.$$

Rearranging, it yields

$$f(x_{n+1}) - f(z) \le \frac{\beta}{2}\,(\|y_n - z\|^2 - \|x_{n+1} - z\|^2) = -\frac{\beta}{2}\,\|x_{n+1} - y_n\|^2 - \beta\,\langle x_{n+1} - y_n, y_n - z\rangle\,.$$

[5] Perhaps I will change my mind in a future edition of these notes.

We apply this inequality with two points, $z = x_n$ and $z = x_\star$. By multiplying the first inequality by $\lambda_{n+1} - 1 \ge 0$ and adding it to the second inequality, it implies

$$\begin{aligned}
&(\lambda_{n+1} - 1)\,(f(x_{n+1}) - f(x_n)) + f(x_{n+1}) - f_\star \\
&\qquad \le -\frac{\beta\lambda_{n+1}}{2}\,\|x_{n+1} - y_n\|^2 - \beta\,\langle x_{n+1} - y_n,\, \lambda_{n+1}\,y_n - (\lambda_{n+1} - 1)\,x_n - x_\star\rangle \\
&\qquad = \frac{\beta}{2\lambda_{n+1}}\,\bigl(\|\lambda_{n+1}\,y_n - (\lambda_{n+1} - 1)\,x_n - x_\star\|^2 - \|\lambda_{n+1}\,x_{n+1} - (\lambda_{n+1} - 1)\,x_n - x_\star\|^2\bigr)\,,
\end{aligned}$$

where the last line uses the identity $\|a\|^2 + 2\,\langle a, b\rangle = \|a + b\|^2 - \|b\|^2$. Our goal is to produce a telescoping sum, which is the case if we ensure that

$$\lambda_{n+1}\,x_{n+1} - (\lambda_{n+1} - 1)\,x_n = \lambda_{n+2}\,y_{n+1} - (\lambda_{n+2} - 1)\,x_{n+1}\,.$$

By substituting in $y_{n+1} = x_{n+1} + \theta_{n+1}\,(x_{n+1} - x_n)$, some algebra shows that it suffices to take $\theta_{n+1} = (\lambda_{n+1} - 1)/\lambda_{n+2}$.

After multiplying the above inequality by λ_{n+1} and summing, we find that

$$\frac{\beta}{2}\,\|\lambda_1\,y_0 - (\lambda_1 - 1)\,x_0 - x_\star\|^2 \ge \sum_{n=0}^{N-1}\{\lambda_{n+1}^2\,(f(x_{n+1}) - f_\star) - \lambda_{n+1}\,(\lambda_{n+1} - 1)\,(f(x_n) - f_\star)\}\,.$$

We also want the right-hand side to telescope, so we set $\lambda_{n+1}\,(\lambda_{n+1} - 1) = \lambda_n^2$, which yields the recursion $\lambda_{n+1} = \frac{1}{2}\,(1 + \sqrt{1 + 4\lambda_n^2})$. With $\lambda_0 = 0$, it yields

$$f(x_N) - f_\star \le \frac{\beta\,\|y_0 - x_\star\|^2}{2\lambda_N^2} = \frac{\beta\,\|x_0 - x_\star\|^2}{2\lambda_N^2}\,.$$

Finally, it is straightforward to show by induction that $\lambda_N \ge N/2$. □

By applying the reduction in Lemma 4.1, it also yields an accelerated algorithm for the strongly convex case, i.e., an algorithm that achieves $f(x_N) - f_\star \le \varepsilon$ in $O(\sqrt{\kappa}\log\frac{\alpha R^2}{\varepsilon})$ iterations, where $R := \|x_0 - x_\star\|$.

Example 5.11. If we apply the accelerated method to logistic regression (see Example 3.8), it improves the iteration complexity from $O(R^2 n/d)$ to $O(R\sqrt{n/d})$.

Bibliographical notes

The simple proof of Theorem 5.4 is taken from [NY83]. The discussion on Chebyshev polynomials follows [Vis12].

The literature on acceleration is too large to be surveyed here, but we mention a recent result in a somewhat different direction: what is the best rate of GD just by changing the step sizes? Thus, we consider the iteration $x_{n+1} = x_n - h_n \nabla f(x_n)$, with the only freedom being to choose the sequence $\{h_n\}_{n\in\mathbb{N}}$. It turns out a constant step size schedule is not optimal, and as established in [AP24a; AP24b], the so-called silver step size schedule achieves the rates of Lemma 4.1 and Lemma 4.2 with $\phi(x) = x^{\log_\rho 2} \approx x^{0.786}$ with $\rho := 1+\sqrt{2}$. This is a rate intermediate between the unaccelerated rate of GD and the accelerated rate of AGD.

Exercises

Exercise 5.1. Define the polynomial $Q_n(x) = T_n(\frac{\alpha+\beta-2x}{\beta-\alpha})/T_n(\frac{\alpha+\beta}{\beta-\alpha})$. Show that $Q_n(0) = 1$ and use Definition 5.6 to establish the identity

$$T_n(x) = \frac{1}{2}\left((x-\sqrt{x^2-1})^n + (x+\sqrt{x^2-1})^n\right) \qquad \text{for } x \in [-1,1]\,.$$

One can show that this identity actually holds for all $x \in \mathbb{R}$. Use this to show that

$$\max_{x\in[\alpha,\beta]} |Q_n(x)| \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n.$$

Note that by combining this with Lemma 5.5, it yields an exponential rate of convergence for CG matching the lower bound of Exercise 4.2.

Exercise 5.2. To better understand the proof of Theorem 5.8, consider a Lyapunov function of the form

$$\mathscr{L}_t = \|x_t - x_\star\|^2 + a_t\,\langle x_t - x_\star, p_t\rangle + b_t\,\|p_t\|^2 + c_t\,(f(x_t) - f_\star)\,.$$

Note that this is the most general Lyapunov function consisting of a combination of a quadratic function in $x_t - x_\star$ and p_t, as well as the objective gap; here, it is crucial that we include the mixed term $a_t\,\langle x_t - x_\star, p_t\rangle$. Our goal is to choose the coefficients a_t, b_t, c_t so that $\dot{\mathscr{L}}_t \le 0$.

Compute the derivative in time of $\mathscr{L}_t$ along AGF with $\gamma_t = 3/t$, and apply convexity to the term $\langle \nabla f(x_t), x_t - x_\star\rangle$. In the resulting expression, since the terms $\langle x_t - x_\star, p_t\rangle$ and $\langle \nabla f(x_t), p_t\rangle$ do not have definite signs, ensure that the coefficients in front of these terms vanish through a suitable choice of a_t, b_t, c_t. Show that this leads to $a_t = t + \bar{a}t^3$ for some $\bar{a} \ge 0$. Next, from the remaining terms, obtain the condition $\dot{b}_t \le \min\{\frac{a_t}{2}, \frac{6b_t}{t} - a_t\}$, which implies $3\dot{b}_t \le 6b_t/t$, hence we consider $b_t = b_0 + \bar{b}t^2$ for some $b_0, \bar{b} \ge 0$. Furthermore, argue that we must take $\bar{a} = 0$ and $\bar{b} = \frac{1}{4}$. To ensure that $\mathscr{L}_0$ only depends on $\|x_0 - x_\star\|$, we set $b_0 = c_0 = 0$. Finally, check that with these choices, we have $b_t \ge a_t^2/4$, which is necessary to ensure that $\mathscr{L}_t \ge c_t\,(f(x_t) - f_\star)$.

Show that the Lyapunov function derived in this way coincides with the one used in Theorem 5.8.

Exercise 5.3. Prove Theorem 5.9.

Hint: Let $z_t \coloneqq x_t + \frac{2}{\gamma}\,p_t$ and consider

$$\mathscr{L}_t \coloneqq f(x_t) - f_\star + \frac{\alpha}{2}\,\|z_t - x_\star\|^2\,.$$

6 [2/4–2/13] Non-smooth convex optimization

Thus far, we have considered the *unconstrained* minimization of convex and *smooth* functions f. The next step is to consider a far more general class of problems by allowing for constraints and non-smoothness.

The two issues are related. To minimize f over a convex set $\mathcal{C}$, it is equivalent to minimize $f + \chi_{\mathcal{C}}$ over all of $\mathbb{R}^d$, where $\chi_{\mathcal{C}}$ is the convex indicator function for $\mathcal{C}$:

$$\chi_{\mathcal{C}}(x) \coloneqq \begin{cases} 0\,, & x \in \mathcal{C}\,, \\ +\infty\,, & x \notin \mathcal{C}\,. \end{cases} \tag{6.1}$$

In this reformulation, the objective function is allowed to take the value $+\infty$ and is certainly non-smooth. Even if we do not reformulate the problem in this way, convex constraint sets often arise as the intersection of primitive constraints: $\mathcal{C} = \{f_i \le 0 \text{ for all } i \in [m]\}$. This is equivalent to $\mathcal{C} = \{\max_{i\in[m]} f_i \le 0\}$, and the function $\max_{i\in[m]} f_i$ is non-smooth.

On the other hand, without strong convexity, it is not guaranteed that f admits a minimizer over all of $\mathbb{R}^d$ (e.g., f is a linear function, or consider the exponential function over $\mathbb{R}$). It often makes sense to consider non-smooth minimization over bounded sets. Thus, we tackle constraints and non-smoothness together.

Although we do not assume smoothness, we still need some minimal regularity for the function f. As justified in Lemma 6.7, convex functions are actually Lipschitz continuous in the interior of their domains, so it is natural to take as our new function class under consideration the class of convex and Lipschitz functions over bounded convex sets.

6.1 Convex analysis

We now work with convex functions $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. The fact that f can now take on the value $+\infty$ leads to some technical issues, but it allows us to seamlessly handle constraints. Convexity can be defined in the usual way, but it is sometimes convenient to instead work with the epigraph.

Definition 6.1. The **epigraph** of $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is the following subset of $\mathbb{R}^d \times \mathbb{R}$:

$$\operatorname{epi} f := \{(x, t) \in \mathbb{R}^d \times \mathbb{R} : f(x) \le t\}\,.$$

Definition 6.2. A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is **convex** if for all $x, y \in \mathbb{R}^d$ and all $t \in [0, 1]$, it holds that

$$f((1-t)\,x + t\,y) \le (1-t)\,f(x) + t\,f(y)\,.$$

Equivalently, f is convex if and only if $\operatorname{epi} f$ is a convex set.

Definition 6.3. The **domain** of a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is the set

$$\operatorname{dom} f := \{x \in \mathbb{R}^d : f(x) < \infty\}\,.$$

The first point to emphasize is that at this level of generality, f can still be quite pathological. Indeed, consider the following function:

$$f(x) := \begin{cases} 0\,, & \|x\| < 1\,, \\ \phi(x)\,, & \|x\| = 1\,, \\ +\infty\,, & \|x\| > 1\,, \end{cases} \tag{6.2}$$

where ϕ is an *arbitrary* non-negative function defined on the sphere $\{\|\cdot\| = 1\}$. Then, one can check that f is convex. However, ϕ need not be continuous or be coherent in any way whatsoever. To avoid these types of situations, the basic regularity property that we impose is that f is lower semicontinuous.

Definition 6.4. A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is **lower semicontinuous** if for all sequences $\{x_n\}_{n\in\mathbb{N}}$ converging to a point $x \in \mathbb{R}^d$, it holds that

$$f(x) \le \liminf_{n\to\infty} f(x_n)\,.$$

In other words, when we pass to the limit of a convergent sequence, the value of f can only drop down. One way to motivate the relevance of this condition for convex optimization is that we often consider suprema $f = \sup_{\omega \in \Omega} f_\omega$ where $\{f_\omega\}_{\omega \in \Omega}$ is a collection of continuous functions; in fact, in many cases, we consider suprema of affine functions. When Ω is finite, we know that the maximum of finitely many continuous functions is continuous. But when Ω is infinite, the suprema of infinitely many continuous functions need not be continuous. The class of lower semicontinuous functions is the smallest class of functions which contains all continuous functions and is closed under taking arbitrary suprema. Further properties are explored in Exercise 6.1.

It follows from that exercise that f is convex and lower semicontinuous if and only if its epigraph is closed and convex. So, when it comes to functions, we impose convexity and lower semicontinuity; and when it comes to sets, we impose convexity and closedness. For example, one can also check that the convex indicator $\chi_{\mathcal{C}}$ is lower semicontinuous if and only if $\mathcal{C}$ is closed. We use the following terminology.[6]

Definition 6.5. A convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is **regular** if: it is not identically equal to $+\infty$, it is lower semicontinuous, and its domain has non-empty interior.

Note that the definition excludes one more pathological case, the function $f(x) = +\infty$ for all $x \in \mathbb{R}^d$, which is of no interest to us. Since the domain of a convex function is a convex set, if it has empty interior then it must be contained in a lower-dimensional affine space, and when we restrict to that space, the domain then has a non-empty interior; this is usually summarized by saying that any non-empty convex set has a non-empty *relative interior*. We do not delve into the details here, but this is why we regard the condition that the domain has non-empty interior as "without loss of generality".

We also note that in the proof of existence of a minimizer, it is really only lower semicontinuity that matters.

Lemma 6.6 (existence of minimizer)**.** Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be lower semicontinuous and its level sets be bounded. Then, there exists a global minimizer of f.

Proof. The proof is the same as for Lemma 1.7, except that lower semicontinuity substitutes for continuity. □

[6] This is not standard terminology but it is convenient.

Regularity. Our next order of business is to establish properties of regular convex functions which allow us to manipulate them in proofs. In particular, we show that they are "almost" differentiable, even though we did not assume it a priori; the source of this regularity is the convexity condition.

Lemma 6.7 (Lipschitz continuity)**.** Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be convex and let $x_0 \in \operatorname{int} \operatorname{dom} f$. Then, f is locally Lipschitz continuous around x_0.

Proof. We may assume that $x_0 = 0$. Since 0 belongs to the interior of $\operatorname{dom} f$, we can fit a simplex centered at the origin inside the domain: namely, there exists $\varepsilon > 0$ such that $\mathcal{C} := \operatorname{conv}\{\pm\varepsilon e_k : k \in [d]\}$ belongs to $\operatorname{dom} f$. First, we show that f is bounded on $\mathcal{C}$: the upper bound follows because $f(\pm\varepsilon e_k) < \infty$ for all $k \in [d]$ and the maximum of f over $\mathcal{C}$ is attained at one of the vertices (why?). For the lower bound, by convexity we have $f(x) \ge 2f(0) - f(-x) \ge 2f(0) - \max_{\mathcal{C}} f$ for all $x \in \mathcal{C}$.

Next, we show that f is Lipschitz on the smaller set $\mathcal{C}' := \operatorname{conv}\{\pm\frac{\varepsilon}{2}\, e_k : k \in [d]\}$. The point is that there is a constant $c_{d,\varepsilon} > 0$ such that for all $x, y \in \mathcal{C}'$, there is a point $y^+ \in \mathcal{C}$ such that the line segment from x to y is contained in the line segment from x to y^+, and the extension is not too short: $\|y^+ - x\| \ge c_{d,\varepsilon}$. Then, by convexity,

$$f(y) = f\Big(\frac{\|y^+ - y\|}{\|y^+ - x\|}\, x + \frac{\|y - x\|}{\|y^+ - x\|}\, y^+\Big) \le \frac{\|y^+ - y\|}{\|y^+ - x\|}\, f(x) + \frac{\|y - x\|}{\|y^+ - x\|}\, f(y^+)\,,$$

hence

$$f(y) - f(x) \le \frac{\|y - x\|}{\|y^+ - x\|}\, (f(y^+) - f(x)) \le \frac{\sup_{\mathcal{C}} f - \inf_{\mathcal{C}} f}{c_{d,\varepsilon}}\, \|y - x\|\,.$$

Interchanging x and y proves the Lipschitz bound. □

This lemma shows that locally near x_0, $f(x)$ grows at most linearly in the distance $\|x - x_0\|$ (as opposed to, say, $\sqrt{\|x - x_0\|}$). This suggests that f may be differentiable at x_0. This is not quite right, because f may have a kink at x_0, but nevertheless we can find an appropriate substitute for differentiability.

Definition 6.8. Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be convex. We say that $p \in \mathbb{R}^d$ is a **subgradient** of f at x if for all $y \in \mathbb{R}^d$, it holds that

$$f(y) \ge f(x) + \langle p, y - x\rangle\,. \tag{6.3}$$

We denote the set of subgradients of f at x as $\partial f(x)$, and we refer to this set as the **subdifferential** of f at x. Also, we set

$$\partial f := \{(x, p) \in \mathbb{R}^d \times \mathbb{R}^d : p \in \partial f(x)\}\,.$$

Note that by definition, if $0 \in \partial f(x)$, then x is a global minimizer of f.

If f is differentiable at $x_0 \in \operatorname{int}\operatorname{dom} f$, then $\partial f(x_0)$ is a singleton: $\partial f(x_0) = \{\nabla f(x_0)\}$ (Exercise 6.2). However, the subdifferential can be multi-valued. A key example is the absolute value function, $f : x \mapsto |x|$, for which $\partial f(0) = [-1, 1]$.

For the purpose of optimization, it is enough to have at least one subgradient, which is the content of the following theorem.

Theorem 6.9 (subdifferential)**.** Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be a regular convex function. If $x_0 \in \operatorname{int}\operatorname{dom} f$, then $\partial f(x_0)$ is **non-empty**, bounded, convex, and closed.

We follow a traditional route of deducing the non-emptiness from a separation theorem. The proof of the following result is deferred.

Theorem 6.10 (supporting hyperplane)**.** Let $\mathcal{C}$ be a closed and convex set, and let $x \in \partial\mathcal{C}$. Then, there exists a non-zero $p \in \mathbb{R}^d$ such that

$$\langle p, x\rangle \le \inf_{\mathcal{C}} \langle p, \cdot\rangle\,.$$

Proof of Theorem 6.9. Since $(x_0, f(x_0)) \in \partial \operatorname{epi} f$, and $\operatorname{epi} f$ is closed and convex (by regularity of f), there is a supporting hyperplane (p, q):

$$\langle p, x_0\rangle + q f(x_0) \le \inf_{(x,t)\in\operatorname{epi} f} \{\langle p, x\rangle + qt\}\,.$$

We can normalize the coefficients so that $\|p\|^2 + q^2 = 1$, and we note that $q \ge 0$.

If x is sufficiently close to x_0, then

$$\langle p, x_0 - x\rangle \le q\,(f(x) - f(x_0)) \le Lq\,\|x - x_0\|\,,$$

where L is the Lipschitz constant of f near x_0. Taking $x = x_0 - \varepsilon p$ for small $\varepsilon > 0$, we deduce that $\|p\| \le Lq$, hence from the normalization condition, $q \ne 0$. Thus, for any $x \in \operatorname{dom} f$, we deduce that

$$f(x) \ge f(x_0) - \frac{1}{q}\,\langle p, x - x_0\rangle\,,$$

thus, $-p/q \in \partial f(x_0)$.

The set $\partial f(x_0)$ is closed and convex as an intersection of the constraints in (6.3). Boundedness follows from Exercise 6.3. □

Constraints. When the constraint set $\mathcal{C}$ is simple, it is reasonable to suppose that we can compute the projection onto $\mathcal{C}$. We study some properties of this projection operator.

Definition 6.11. Let $\mathcal{C}$ be closed and convex. The **projection onto** $\mathcal{C}$ is the mapping $\pi_{\mathcal{C}} : \mathbb{R}^d \to \mathcal{C}$ defined by

$$\pi_{\mathcal{C}}(x) := \operatorname*{arg\,min}_{y \in \mathcal{C}} \|y - x\|^2 .$$

The "arg min" is non-empty because $\mathcal{C}$ is closed, and the uniqueness of the minimizer follows from a strict convexity argument as in Lemma 1.10. When $\mathcal{C}$ is a linear subspace, then $\pi_{\mathcal{C}}$ coincides with the linear algebra definition of projection, and in this case $\pi_{\mathcal{C}}$ is linear. In general, however, $\pi_{\mathcal{C}}$ is a *non-linear* operator.

The following lemma characterizes the projection.

Lemma 6.12 (characterization of projection)**.** Let $\mathcal{C}$ be closed and convex, and let $x \notin \mathcal{C}$. Then, $\pi_{\mathcal{C}}(x)$ is the unique point satisfying the following condition:

$$\langle \pi_{\mathcal{C}}(x) - x,\, x' - \pi_{\mathcal{C}}(x)\rangle \ge 0 \qquad \text{for all } x' \in \mathcal{C} . \tag{6.4}$$

Proof. As in the proof of Lemma 1.8, the first-order necessary condition for optimality reads $\langle \pi_{\mathcal{C}}(x) - x, v\rangle \ge 0$. However, because the optimization problem is constrained to lie in $\mathcal{C}$, this time we do not have the inequality for all v, but only for v of the form $x' - \pi_{\mathcal{C}}(x)$ where $x' \in \mathcal{C}$. □

This lemma furnishes the following important property.

Lemma 6.13 (convex projections are non-expansive)**.** Let $\mathcal{C}$ be closed and convex. Then, for all $x, y \in \mathbb{R}^d$,

$$\|\pi_{\mathcal{C}}(y) - \pi_{\mathcal{C}}(x)\| \le \|y - x\| .$$

Proof. By (6.4),

$$\begin{aligned}\langle \pi_{\mathcal{C}}(x) - x,\, \pi_{\mathcal{C}}(y) - \pi_{\mathcal{C}}(x)\rangle &\ge 0 ,\\ \langle \pi_{\mathcal{C}}(y) - y,\, \pi_{\mathcal{C}}(x) - \pi_{\mathcal{C}}(y)\rangle &\ge 0 .\end{aligned}$$

Adding these inequalities yields

$$\|\pi_{\mathcal{C}}(y) - \pi_{\mathcal{C}}(x)\|^2 \le \langle \pi_{\mathcal{C}}(y) - \pi_{\mathcal{C}}(x),\, y - x\rangle \le \|\pi_{\mathcal{C}}(y) - \pi_{\mathcal{C}}(x)\|\, \|y - x\| . \qquad \square$$

Actually, we can now return to prove the supporting hyperplane theorem.

Proof of Theorem 6.10. First, we show that if $\mathcal{C}$ is a closed convex set and $x \notin \mathcal{C}$, then we can separate $\mathcal{C}$ from x. Namely, by (6.4), the vector $p \coloneqq \pi_{\mathcal{C}}(x) - x$ is non-zero and satisfies

$$\inf_{x' \in \mathcal{C}} \langle p, x' \rangle \ge \langle p, \pi_{\mathcal{C}}(x) \rangle = \|\pi_{\mathcal{C}}(x) - x\|^2 + \langle p, x \rangle \ge \langle p, x \rangle \,.$$

To prove the supporting hyperplane theorem, note that since $x \in \partial\mathcal{C}$, there is a sequence of points $\{x_n\}_{n\in\mathbb{N}}$ which lies outside of $\mathcal{C}$, such that $x_n \to x$. For each n, let p_n be a hyperplane that separates $\mathcal{C}$ from x_n, and by normalizing we may assume that $\|p_n\| = 1$. Since $\{p_n\}_{n\in\mathbb{N}}$ is a bounded sequence, it contains a subsequence which converges to some unit vector p. By taking limits, it is easy to see that p is a supporting hyperplane. □

6.2 Projected subgradient methods

Methods for constrained optimization differ based on what they assume about the constraint set. The first method we study assumes access to the projection mapping $\pi_{\mathcal{C}}$ for the set $\mathcal{C}$. This assumption is appropriate when the set $\mathcal{C}$ is particularly "simple", e.g., $\mathcal{C}$ is the ball $\mathcal{C} = \{\|\cdot\| \le R\}$, in which case the projection can be computed in closed form. When $\mathcal{C}$ is more complex, e.g., $\mathcal{C}$ is a polytope, we need more sophisticated methods.

Projected subgradient descent is the following method:

$$x_{n+1} \coloneqq \pi_{\mathcal{C}}\big(x_n - h\,\frac{p_n}{\|p_n\|}\big)\,, \qquad p_n \in \partial f(x_n)\,. \tag{PSD}$$

Note that we use the normalized subgradient $p_n/\|p_n\|$. If we think about the example of the absolute value function $|\cdot|$ with subdifferential $[-1, 1]$ at the origin, we see that the magnitude of an arbitrary element of the subdifferential need not be informative. Instead, the intuition behind non-smooth optimization is to use the subgradients as separating directions: in particular, by convexity, $f(x) - f(x_n) \ge \langle p_n, x - x_n \rangle$, so any minimizer must lie on one side of the hyperplane defined by p_n.

We let $x_\star$ denote a minimizer of f over the closed convex set $\mathcal{C}$, and $f_\star \coloneqq f(x_\star)$.

Theorem 6.14 (convergence of PSD)**.** Let f be convex and L-Lipschitz continuous on the closed convex set $\mathcal{C}$. Then, PSD satisfies

$$f\Big(\frac{1}{N}\sum_{n=0}^{N-1} x_n\Big) - f_\star \le \frac{1}{N}\sum_{n=0}^{N-1}(f(x_n) - f_\star) \le \frac{L}{2Nh}\,\|x_0 - x_\star\|^2 + \frac{Lh}{2}\,.$$

In particular, by setting $h = R/\sqrt{N}$, where R is an upper bound on $\|x_0 - x_\star\|$, it yields the convergence rate

$$f\Big(\frac{1}{N}\sum_{n=0}^{N-1} x_n\Big) - f_\star \le \frac{LR}{\sqrt{N}}\,.$$

Proof. The first inequality holds by convexity, so we focus on the second. The idea is similar to the proof of Theorem 3.4, except that instead of using smoothness to handle the error term, we use Lipschitzness. By expanding the squared distance to the minimizer,

$$\begin{aligned}\|x_{n+1} - x_\star\|^2 &= \Big\|\pi_{\mathcal{C}}\Big(x_n - h\,\frac{p_n}{\|p_n\|}\Big) - \pi_{\mathcal{C}}(x_\star)\Big\|^2 \le \Big\|x_n - h\,\frac{p_n}{\|p_n\|} - x_\star\Big\|^2 \\ &= \|x_n - x_\star\|^2 - \frac{2h}{\|p_n\|}\,\langle p_n, x_n - x_\star\rangle + h^2 \\ &\le \|x_n - x_\star\|^2 - \frac{2h}{\|p_n\|}\,(f(x_n) - f_\star) + h^2\,,\end{aligned}$$

where we used Lemma 6.13. Since $\|p_n\| \le L$ for all n (Exercise 6.3), we sum the inequalities:

$$\frac{1}{N}\sum_{n=0}^{N-1}(f(x_n) - f_\star) \le \frac{L}{2Nh}\,\|x_0 - x_\star\|^2 + \frac{Lh}{2}\,. \qquad \square$$

Thus, the averaged iterate $\bar{x}_N$ satisfies $f(\bar{x}_N) - f_\star \le \varepsilon$ provided $N \ge L^2R^2/\varepsilon^2$. Note that this convergence rate is substantially worse than the one for the smooth case (Theorem 3.4). Another difference is that the descent lemma (Lemma 3.1) is available in the smooth case which implies monotonic decrease of the objective value; here, there is no descent lemma, so the guarantee only holds for the averaged iterate. The analysis can also be performed under strong convexity, see Exercise 6.5.

Interestingly, if we only assume that f is L-Lipschitz continuous over $\mathsf{B}(x_\star, R)$, rather than on all of $\mathcal{C}$, it is still possible to show that $\min_{n=0,\dots,N-1} f(x_n) - f_\star \le LR/\sqrt{N}$, although the proof becomes more involved [Nes18, §3.2.3].

The analysis above shows that when the projection operator is cheap to compute, optimization under constraints is straightforward provided that we interleave the gradient steps with projection steps. We next tackle a more general setting in which we separate out the constraints into a "simple" set $\mathcal{C}$ for which we can compute the projection operator, and additional functional constraints $\{f_i \le 0 \text{ for all } i \in [m]\}$. Thus, we consider

$$\min\{f(x) \mid x \in \mathcal{C}\,,\ f_i(x) \le 0 \text{ for all } i \in [m]\}\,.$$

We assume that $f, f_1, \dots, f_m$ are all regular convex functions, and write $f_{\max} := \max_{i\in[m]} f_i$. The next algorithm is known as the projected subgradient method with functional constraints. For $n = 0, 1, \dots, N-1$:

- If $f_{\max}(x_n) \le \varepsilon$, set

$$x_{n+1} := \pi_{\mathcal{C}}\Big(x_n - \frac{\varepsilon}{\|p_n\|^2}\, p_n\Big)\,, \qquad p_n \in \partial f(x_n)\,.$$

- Otherwise, set

$$x_{n+1} := \pi_{\mathcal{C}}\Big(x_n - \frac{f_{\max}(x_n)}{\|p_n\|^2}\, p_n\Big)\,, \qquad p_n \in \partial f_{\max}(x_n)\,.$$

The algorithm requires computing elements of the subdifferential for the function $\max_{i\in[m]} f_i$. We therefore first identify this subdifferential.

Lemma 6.15 (subdifferential of a maximum)**.** Let $f_1, \dots, f_m$ be regular convex functions. Then, for all $x \in \mathbb{R}^d$,

$$\partial\big(\max_{i\in[m]} f_i\big)(x) = \operatorname{conv}\Big\{\partial f_i(x) \Bigm| i \in [m]\,,\ f_i(x) = \max_{j\in[m]} f_j(x)\Big\}\,.$$

Proof. ($\supseteq$) Let $f_{\max} := \max_{i\in[m]} f_i$ and $I_\star(x) := \{i \in [m] : f_i(x) = f_{\max}(x)\}$. If λ is a probability vector and $p_i \in \partial f_i(x)$ for all $i \in I_\star(x)$, then

$$f_{\max}(y) \ge \sum_{i\in I_\star(x)} \lambda_i f_i(y) \ge \sum_{i\in I_\star(x)} \lambda_i\,(f_i(x) + \langle p_i, y - x\rangle) = f_{\max}(x) + \Big\langle \sum_{i\in I_\star(x)} p_i,\ y - x\Big\rangle\,.$$

Hence, $\sum_{i\in I_\star(x)} \lambda_i p_i \in \partial f_{\max}(x)$.

($\subseteq$) Since the purpose of this lemma from the perspective of these notes is simply to compute an element of $\partial f_{\max}(x)$, we omit the proof of this direction. It can be proven, e.g., via Lagrangian duality or via more subdifferential theory. □

The next theorem provides the convergence rate for the method.

Theorem 6.16 (convergence of PSD under functional constraints)**.** Let $f, f_1, \dots, f_m$ be convex and L-Lipschitz on the closed convex set $\mathcal{C}$. Then, PSD under functional constraints satisfies

$$\min\{f(x_n) \mid n = 0, 1, \dots, N-1\,,\ f_{\max}(x_n) \le \varepsilon\} - f_\star \le \varepsilon \tag{6.5}$$

provided that

$$N \ge \frac{L^2\,\|x_0 - x_\star\|^2}{\varepsilon^2}\,.$$

The theorem says that after N iterations, we can find a point $\hat{x}_N$ which almost satisfies the functional constraints, in the sense that $f_{\max}(\hat{x}_N) \le \varepsilon$, and moreover $f(\hat{x}_N) - f_\star \le \varepsilon$. The number of iterations is no more than the case without functional constraints.

Proof of Theorem 6.16. There are two cases for the algorithm. If the iteration n belongs to the first case, then as we saw in the proof of Theorem 6.14,

$$\|x_{n+1} - x_\star\|^2 \le \|x_n - x_\star\|^2 - \frac{2\varepsilon}{\|p_n\|^2}\,(f(x_n) - f_\star) + \frac{\varepsilon^2}{\|p_n\|^2}\,.$$

If $f(x_n) - f_\star \le \varepsilon$, then since $f_{\max}(x_n) \le \varepsilon$ (by the definition of the first case), we have met the success condition (6.5). Otherwise, $f(x_n) - f_\star > \varepsilon$, and the inequality above implies

$$\|x_{n+1} - x_\star\|^2 < \|x_n - x_\star\|^2 - \frac{\varepsilon^2}{\|p_n\|^2} \le \|x_n - x_\star\|^2 - \frac{\varepsilon^2}{L^2}\,.$$

What happens in the second case? Here, we *also* show that $\|x_{n+1} - x_\star\| < \|x_n - x_\star\|$: since $x_\star$ satisfies the functional constraints and x_n does not, the subgradient $p_n \in \partial f_{\max}(x_n)$ still acts as a separating hyperplane. Indeed,

$$\begin{aligned}
\|x_{n+1} - x_\star\|^2 &= \Big\|\pi_{\mathcal{C}}\big(x_n - \frac{f_{\max}(x_n)}{\|p_n\|^2}\,p_n\big) - \pi_{\mathcal{C}}(x_\star)\Big\|^2 \le \Big\|x_n - \frac{f_{\max}(x_n)}{\|p_n\|^2}\,p_n - x_\star\Big\|^2 \\
&= \|x_n - x_\star\|^2 - \frac{2 f_{\max}(x_n)}{\|p_n\|^2}\,\langle p_n, x_n - x_\star\rangle + \frac{f_{\max}(x_n)^2}{\|p_n\|^2} \\
&\le \|x_n - x_\star\|^2 - \frac{2 f_{\max}(x_n)}{\|p_n\|^2}\,f_{\max}(x_n) + \frac{f_{\max}(x_n)^2}{\|p_n\|^2} < \|x_n - x_\star\|^2 - \frac{\varepsilon^2}{L^2}\,.
\end{aligned}$$

Summing these inequalities across the iterations yields

$$\|x_N - x_\star\|^2 < \|x_0 - x_\star\|^2 - \frac{N\varepsilon^2}{L^2}\,.$$

For $N \ge L^2 \|x_0 - x_\star\|^2/\varepsilon^2$, this is not possible unless we reach the success condition (6.5) by iteration N. □

Example 6.17 (soft-margin SVM). An example of a problem that can be tackled via projected subgradient methods is soft-margin support vector machine (SVM) classification. Suppose that we have a dataset $\{(x_i, y_i)\}_{i\in[n]}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$. The output of the soft-margin SVM is the classifier $x \mapsto \operatorname{sgn}(\langle \theta^\star, x\rangle)$, where $\theta^\star$ minimizes

$$\theta \mapsto \frac{1}{n}\sum_{i=1}^n \ell_{\mathrm{hinge}}(y_i, \langle \theta, x_i\rangle) + \frac{\lambda}{2}\|\theta\|^2\,.$$

Here, $\ell_{\mathrm{hinge}}(y, \hat y) \coloneqq \max\{0, 1 - y\hat y\}$ is the hinge loss, $\lambda > 0$ is a regularization parameter, and we have omitted the bias term (which can be handled by augmenting the feature vector x as usual). This objective is strongly convex and Lipschitz over bounded sets, so we can apply projected subgradient descent (projecting onto, e.g., a Euclidean ball).

6.3 Cutting plane methods

Non-smooth optimization uses subgradient directions in order to “localize” the solution set. Pursuing this line of reasoning further leads to the family of cutting plane methods.

Suppose that we wish to minimize f over a bounded, closed, convex set $\mathcal{C}$. Let $\mathcal{C}_\star$ denote the set of minimizers. The idea is to construct a sequence of convex sets $\mathcal{C} = \mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \dots,$ which shrink toward $\mathcal{C}_\star$. The set $\mathcal{C}_n$ represents possible candidates for the solution to the problem at iteration n.

If $x_n \in \mathcal{C}_n$ and $p_n \in \partial f(x_n)$, then the subgradient inequality reads

$$0 \ge f(x_\star) - f(x_n) \ge \langle p_n, x_\star - x_n\rangle \qquad \text{for all } x_\star \in \mathcal{C}_\star\,.$$

Thus,

$$\mathcal{C}_\star \subseteq \mathcal{C}_n \cap \{x \in \mathbb{R}^d : \langle p_n, x\rangle \le \langle p_n, x_n\rangle\}\,.$$

We can take $\mathcal{C}_{n+1}$ to be any superset of the right-hand side above.

To finish specifying the scheme, we need a rule for choosing the points x_n and the sets $\mathcal{C}_n$, with the goal of $\mathcal{C}_n$ shrinking as fast as possible. The key is the following lemma from convex geometry, which we do not prove.

Lemma 6.18 (Grünbaum)**.** Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a convex body (i.e., a compact convex set with non-empty interior) and let $x_{\mathcal{C}}$ denote the *centroid* of $\mathcal{C}$: $x_{\mathcal{C}} := (\operatorname{vol}\mathcal{C})^{-1} \int_{\mathcal{C}} x \, \mathrm{d}x$. Then, for any half-space $\mathcal{H}$ containing $x_{\mathcal{C}}$,

$$\frac{\operatorname{vol}(\mathcal{C} \cap \mathcal{H})}{\operatorname{vol}(\mathcal{C})} \ge \Bigl(\frac{d}{d+1}\Bigr)^d \ge \frac{1}{\mathrm{e}},$$

where $\mathrm{e} \approx 2.72$ is a numerical constant.

Consequently, if we choose x_n to be the centroid of $\mathcal{C}_n$ and set

$$\mathcal{C}_{n+1} = \mathcal{C}_n \cap \{\langle p_n, \cdot \rangle \le \langle p_n, x_n \rangle\}, \qquad x_n = x_{\mathcal{C}_n}, \tag{CoGM}$$

then Grünbaum's inequality shows that $\operatorname{vol}(\mathcal{C}_n \setminus \mathcal{C}_{n+1})/\operatorname{vol}(\mathcal{C}_n) \le 1/\mathrm{e}$, or

$$\frac{\operatorname{vol}(\mathcal{C}_{n+1})}{\operatorname{vol}(\mathcal{C}_n)} \le 1 - \frac{1}{\mathrm{e}}.$$

Thus, we cut away a constant fraction of the volume at each iteration. This is known as the *center of gravity* method.

As stated, CoGM is not a practical method. The feasible set $\mathcal{C}_n$ at iteration n can be quite complicated, making it prohibitively expensive to compute its centroid. Centroids can be computed via Markov chain Monte Carlo (MCMC) methods for numerical integration, with guarantees available due to recent advances in log-concave sampling, but it is generally understood that this is a more difficult computational problem than the original convex optimization problem we set out to solve. Nevertheless, CoGM achieves the optimal complexity bound in the oracle model, so let us analyze its efficiency.

Theorem 6.19 (center of gravity)**.** Let $D := \operatorname{diam}\mathcal{C}$ and let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and L-Lipschitz on $\mathcal{C}$. Then, CoGM satisfies

$$f(x_{N-1}) - f_\star \le DL\,\Bigl(1 - \frac{1}{\mathrm{e}}\Bigr)^{N/d}.$$

Proof. By the argument above, at iteration N, $\operatorname{vol}(\mathcal{C}_N)/\operatorname{vol}(\mathcal{C}) \le \lambda^N$, where we can take $\lambda = 1 - 1/\mathrm{e}$. Now consider the set $\hat{\mathcal{C}} := (1-t)\,x_\star + t\,\mathcal{C}$, where we choose t so that $\operatorname{vol}(\hat{\mathcal{C}}) > \operatorname{vol}(\mathcal{C}_N)$; since $\operatorname{vol}(\hat{\mathcal{C}}) = t^d \operatorname{vol}(\mathcal{C})$, we can take any $t > \lambda^{N/d}$. With this choice, there exists $\hat{x} \in \hat{\mathcal{C}} \setminus \mathcal{C}_N$. By the definition of $\mathcal{C}_N$,

$$f(x_{N-1}) - f_\star \le f(\hat{x}) - f_\star \le t\,\bigl(\sup_{\mathcal{C}} f - f_\star\bigr) \le tDL.$$

The result follows by letting $t \searrow \lambda^{N/d}$. □

Thus, in principle, we can achieve $f(x_{N-1}) - f_\star \le \varepsilon$ in $O(d \log(DL/\varepsilon))$ iterations. Compared to Theorem 6.14, this result incurs only a logarithmic dependence on the ratio DL/ε, i.e., we can output a high-accuracy solution even for poorly conditioned convex sets. On the other hand, it incurs dependence on the dimension.

Recall that the lower bound for convex smooth optimization (Theorem 4.4) only applies in dimension $d \gtrsim \sqrt{\beta R^2/\varepsilon}$. The center of gravity method explains why: a β-smooth function over a ball of radius R is also βR-Lipschitz, so Theorem 6.19 yields an oracle complexity of $O(d \log(\beta R^2/\varepsilon))$ in this case. This is smaller than the lower bound of $\Omega(\sqrt{\beta R^2/\varepsilon})$ in Theorem 4.4 when $d \ll \sqrt{\beta R^2/\varepsilon}/\log(\beta R^2/\varepsilon)$, so a lower bound construction cannot exist in any smaller dimension.[7] Note also that for convex quadratic minimization, there are methods which find the minimizer in d queries (e.g., Theorem 5.3 for CG); the center of gravity method almost achieves this guarantee for general convex optimization.

Toward making cutting plane methods more practical, a famous example is the *ellipsoid method*. In this scheme, we take each set $\mathcal{C}_n$ to be an ellipsoid,

$$\mathcal{C}_n = \{x \in \mathbb{R}^d : \langle x - x_n, \Sigma_n^{-1}\,(x - x_n)\rangle \le 1\}\,. \tag{6.6}$$

At the next iteration, we must find a new ellipsoid $\mathcal{C}_{n+1}$ such that

$$\mathcal{C}_{n+1} \supseteq \mathcal{C}_n \cap \{x \in \mathbb{R}^d : \langle p_n, x\rangle \le \langle p_n, x_n\rangle\}\,. \tag{6.7}$$

Here, we use the following geometric lemma (Exercise 6.7).

Lemma 6.20 (ellipsoid)**.** Let $\mathcal{C}_n$ be the ellipsoid (6.6) and let $p_n \in \mathbb{R}^d$ be a non-zero vector. Define $\mathcal{C}_{n+1} := \{x \in \mathbb{R}^d : \langle x - x_{n+1}, \Sigma_{n+1}^{-1}\,(x - x_{n+1})\rangle \le 1\}$, where

$$\begin{aligned} x_{n+1} &:= x_n - \frac{1}{d+1}\,\frac{\Sigma_n p_n}{\sqrt{\langle p_n, \Sigma_n\, p_n\rangle}}\,, \\ \Sigma_{n+1} &:= \frac{d^2}{d^2-1}\Bigl(\Sigma_n - \frac{2}{d+1}\,\frac{\Sigma_n p_n p_n^\top \Sigma_n}{\langle p_n, \Sigma_n\, p_n\rangle}\Bigr)\,. \end{aligned}$$

Then, for $d > 1$, $\mathcal{C}_{n+1}$ satisfies (6.7) and

$$\frac{\operatorname{vol}(\mathcal{C}_{n+1})}{\operatorname{vol}(\mathcal{C}_n)} = \sqrt{\frac{d-1}{d+1}\,\Bigl(\frac{d^2}{d^2-1}\Bigr)^d} = 1 - \Omega\bigl(\frac{1}{d}\bigr)\,.$$

[7] This discussion is not entirely correct since Theorem 4.4 only applies to gradient span algorithms, which does not cover CoGM. However, the moral of the discussion is true for bona fide oracle lower bounds.

By following the proof of Theorem 6.19, replacing λ by $1 - \Omega(1/d)$, one obtains the same guarantee as for CoGM but with iteration count $O(d^2 \log(LD/\varepsilon))$. (See Exercise 6.6 for details.) Thus, the cost of obtaining an implementable version of the center of gravity method is a larger query complexity. Naturally, there have been numerous follow-up works in the field which aim at achieving the best of both worlds.

6.4 Lower bounds

In this section, we study lower bounds for convex non-smooth optimization.

Theorem 6.21 (lower bound for convex, non-smooth minimization)**.** For any $x_0 \in \mathbb{R}^d$, $d > N$, and $L, R > 0$, there exists a convex and L-Lipschitz function f over $\mathsf{B}(x_\star, R)$ such that $x_0 \in \mathsf{B}(x_\star, R)$ and for any gradient span algorithm,

$$f(x_N) - f_\star \gtrsim \frac{LR}{\sqrt{N}}\,.$$

Proof. Assume $x_0 = 0$ and define the function $f : \mathbb{R}^d \to \mathbb{R}$ by

$$f(x) \coloneqq \gamma \max_{i\in[d]} x[i] + \frac{\alpha}{2}\,\|x\|^2\,,$$

where $\alpha, \gamma > 0$ are to be chosen. Note that this function is Lipschitz with constant $\gamma + \alpha\,(\|x_\star\| + R)$. Also, if $I_\star(x) \coloneqq \{i \in [d] : x[i] = \max_{j\in[d]} x[j]\}$, then from Lemma 6.15,

$$\partial f(x) = \alpha x + \gamma \operatorname{conv}\{e_i : i \in I_\star(x)\}\,.$$

The optimal point is $x_\star[k] = -\gamma/(\alpha d)$ for $k \in [d]$, by checking that $0 \in \partial f(x_\star)$. Thus, $\|x_\star\| = \gamma/(\alpha\sqrt{d})$ and the Lipschitz constant is at most $2\gamma + \alpha R$.

We take a subgradient oracle which, given a point x, outputs $\alpha x + \gamma e_i \in \partial f(x)$, where $i = \min I_\star(x)$ is the *first* coordinate of x that achieves the maximum. From this property, it is straightforward to show via induction that $x_n \in \mathcal{V}_n$ for all n, where $\mathcal{V}_n$ is the subspace from the proof of Theorem 4.4.

Since $d > N$, it follows that $f(x_N) \ge 0$. On the other hand,

$$f_\star = f(x_\star) = -\frac{\gamma^2}{\alpha d} + \frac{\gamma^2}{2\alpha d} = -\frac{\gamma^2}{2\alpha d}\,.$$

We set $d = N + 1$, $\gamma = L/4$, $\alpha = \gamma/(R\sqrt{d})$ (to ensure that $\|x_0 - x_\star\| \le R$), which leads to a Lipschitz constant of $L/2 + L/(4\sqrt{d}) \le L$. It yields

$$f(x_N) - f_\star \ge -f(x_\star) \gtrsim \frac{LR}{\sqrt{N}}\,. \qquad \square$$

Note that this matches the guarantee of PSD (Theorem 6.14), so projected subgradient descent is *optimal* in the non-smooth setting. In other words, without smoothness, there is no acceleration phenomenon.

There is a version of Theorem 6.21 in the strongly convex case (Exercise 6.8).

Theorem 6.22 (lower bound for strongly convex, non-smooth minimization)**.** For any $x_0 \in \mathbb{R}^d$, $d > N$, and $\alpha, L > 0$, there exists $R > 0$ and an α-convex and L-Lipschitz function f over $\mathsf{B}(x_\star, R)$ such that $x_0 \in \mathsf{B}(x_\star, R)$ and for any gradient span algorithm,

$$f(x_N) - f_\star \gtrsim \frac{L^2}{\alpha N}\,.$$

Next, in the low-dimensional setting, the following lower bound holds.

Theorem 6.23 (lower bound for convex, non-smooth minimization II)**.** The oracle complexity of minimizing convex, L-Lipschitz functions over $[-R, R]^d$ to accuracy ε is at least $\Omega(d \log(LR/\varepsilon))$.

This shows that CoGM is optimal as well. Actually, we do not prove Theorem 6.23; instead, we focus on the related but harder task of feasibility.

Definition 6.24. Let $0 < \delta < R$. Let $\mathcal{C} \subseteq [-R, R]^d$ be a closed convex set such that there exists a ball $\mathsf{B}(x_\star, \delta) \subseteq \mathcal{C}$. The **feasibility problem** with parameters (δ, R) is the problem of outputting a point in $\operatorname{int} \mathcal{C}$, given access to a separation oracle. Namely, given a point $x \in \mathbb{R}^d$, the separation oracle either reports that $x \in \mathcal{C}$, or it outputs a non-zero vector $p \in \mathbb{R}^d$ such that $\sup_{\mathcal{C}} \langle p, \cdot \rangle \le \langle p, x \rangle$.

If one can solve the feasibility problem, then one can solve the convex Lipschitz minimization problem. Indeed, given a convex, L-Lipschitz function f over $[-R, R]^d$, suppose for the sake of argument that we know the optimal value $f_\star$. Consider the feasibility problem for set $\mathcal{C} := \{f - f_\star \le \varepsilon\}$. For $x_\star := \arg\min_{[-R,R]^d} f$, we claim that $\mathsf{B}(x_\star, \varepsilon/L) \subseteq \mathcal{C}$; indeed this follows from L-Lipschitzness.[8] Also, the subgradient oracle

[8] Actually this is not exactly true because $x_\star$ could lie near the boundary of $[-R, R]^d$. To fix this, one could instead look for a minimizer of f over $\mathcal{C}' := [-R + \delta, R - \delta]^d$, i.e., define $x_{\delta,\star}$ to be a minimizer over this smaller cube and set $\mathcal{C} := \mathcal{C}' \cap \{f - f(x_{\delta,\star}) \le \varepsilon\}$. If we take $\delta = \varepsilon/(L\sqrt{d})$, then by L-Lipschitzness we see that any point in $\mathcal{C}$ is a 2ε-minimizer of f over $[-R, R]^d$, and now $\mathsf{B}(x_{\delta,\star}, \delta) \subseteq \mathcal{C}$. This does not really change the argument.

for f yields a separation oracle for $\mathcal{C}$. Thus, solving the feasibility problem for $\mathcal{C}$ with parameters $(\varepsilon/L, R)$ yields an ε-solution to the problem of minimizing f.

Since the feasibility problem is harder, the following theorem is weaker than Theorem 6.23. However, it is easier to prove, and it contains most of the main ideas.

Theorem 6.25 (lower bound for feasibility)**.** For any deterministic algorithm, the feasibility problem with parameters (ε, R) requires $\Omega(d \log(R/\varepsilon))$ queries.

Proof. We play a game with the algorithm. Suppose that the algorithm has chosen query points $x_1, \dots, x_n$ thus far. Our goal is to choose a vector p_n—which is supposed to correspond to the output of a separation oracle—and we provide the algorithm with this vector, which it then uses to produce a new point x_{n+1} and so on. Simultaneously, we also maintain a sequence of convex bodies (actually, boxes) $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_N$.

At the end of the game, the algorithm has produced points $x_1, \dots, x_N$, and we have produced vectors $p_1, \dots, p_N$. By itself, this is not yet meaningful; the algorithm is not designed to produce useful results, *unless* $p_1, \dots, p_N$ are valid outputs from a separation oracle corresponding to a convex body $\mathcal{C}$ satisfying the assumptions of the feasibility problem. So, we aim to choose $p_1, \dots, p_N$ so that this holds with $\mathcal{C} = \mathcal{C}_N$. Now, we can use the following post hoc reasoning: *had we* run the algorithm with the separation oracle for $\mathcal{C}_N$ from the outset, then the algorithm would have output the same sequence of points $x_1, \dots, x_N$, because it is deterministic, so this construction yields a valid lower bound (i.e., it requires more than N iterations to solve the feasibility problem). This proof technique is known as the method of *resisting oracles*, and its main drawback is that it does not apply to randomized algorithms.[9]

Let us instantiate the resisting oracle for the feasibility problem. At each iteration n, the convex body $\mathcal{C}_n$ is the box $\{x \in \mathbb{R}^d : a_n \le x \le b_n\}$; here, $a_n, b_n \in \mathbb{R}^d$ and the inequality is interpreted pointwise. We start with $a_0 = -R\mathbf{1}_d$, $b_0 = +R\mathbf{1}_d$, where $\mathbf{1}_d$ is the all-ones vector; thus, $\mathcal{C}_0 = [-R, R]^d$.

When the algorithm makes the first query x_1, we update the box by cutting it in half, based on the first coordinate of x_1. Namely, if $x_1[1] \le 0$, we set $a_1[1] = 0$, and $a_1[k] = a_0[k]$ for all $k > 1$; we output the separating vector $-e_1$. If $x_1[1] \ge 0$, we set $b_1[1] = 0$ and $b_1[k] = b_0[k]$ for all $k > 1$; we output the separating vector $+e_1$. In either case, $\operatorname{vol}(\mathcal{C}_1) = \frac{1}{2} \operatorname{vol}(\mathcal{C}_0)$ and $x_1 \notin \operatorname{int} \mathcal{C}_1$.

When the algorithm makes the second query x_2, we repeat this procedure except that we cut the box in half along the second coordinate. We continue in this fashion, cycling through the coordinates.

[9] Lower bounds for randomized algorithms require the use of information theory.

Let c_n denote the center of $\mathcal{C}_n$. We now claim that for each n, $\mathsf{B}(c_n, r_n) \subseteq \mathcal{C}_n$, where $r_n = (R/2)\,(1/2)^{n/d}$. Indeed, this is true for $n = 0$. Also, for $n = ad$ for integer a, each side of the box has length $R\,(1/2)^a$, so the result is true in this case too. Finally, for $n = ad + b$, we have $\mathsf{B}(c_{(a+1)d}, R/2^{a+1}) \subseteq \mathcal{C}_{(a+1)d} \subseteq \mathcal{C}_n$ hence $\mathsf{B}(c_n, R/2^{a+1}) \subseteq \mathcal{C}_n$, and we note that $R/2^{a+1} \le (R/2)\,(1/2)^{n/d}$.

The resisting oracle construction succeeds up to iteration N provided that $\mathcal{C}_N$ contains a ball of radius ε. It therefore suffices to have $(R/2)\,(1/2)^{N/d} \ge \varepsilon$, i.e., $N \gtrsim d\log(R/\varepsilon)$. □

Exercises

Exercise 6.1.

1. Prove that a function f is lower semicontinuous if and only if for all $c \in \mathbb{R}$, the level set $\{f \le c\}$ is closed.

2. Prove that a supremum of lower semicontinuous functions is lower semicontinuous.

3. Show that the function defined in (6.2) is lower semicontinuous if and only if $\phi = 0$.

Exercise 6.2. Prove that if f is differentiable at $x_0 \in \operatorname{int}\operatorname{dom} f$, then $\partial f(x_0) = \{\nabla f(x_0)\}$.

Exercise 6.3. Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuous and convex on a convex set $\mathcal{C}$. Prove that f is Lipschitz continuous over $\mathcal{C}$ with constant L if and only if for every $x_0 \in \operatorname{int}\mathcal{C}$ and every $p \in \partial f(x_0)$, we have $\|p\| \le L$.

Exercise 6.4. Compute the subdifferential of the Euclidean norm $\|\cdot\|$.

Exercise 6.5. Assume that f is α-strongly convex and L-Lipschitz continuous over the closed convex set $\mathcal{C}$. Prove that for PSD,

$$f(\bar{x}_N) - f_\star \le \frac{\alpha}{2\,\{(1-\alpha h/L)^N - 1\}}\,\|x_0 - x_\star\|^2 + \frac{Lh}{2}\,,$$

where $\bar{x}_N$ is a suitable averaged iterate. Deduce that by setting $h = \varepsilon/L$, one can achieve $f(\bar{x}_N) - f_\star \le \varepsilon$ in $O(\frac{L^2}{\alpha\varepsilon}\log(\frac{\alpha R^2}{\varepsilon}))$ iterations (compared with $O(L^2R^2/\varepsilon^2)$ iterations, as implied by Theorem 6.14).

Also, show that under these assumptions, $\|x_0 - x_\star\| \le 2L/\alpha$.

Exercise 6.6. The analysis of the ellipsoid method (and general cutting plane schemes) presents an additional difficulty: since the next set $\mathcal{C}_{n+1}$ is only chosen to be a superset of $\mathcal{C}_n \cap \{\langle p_n, \cdot\rangle \le \langle p_n, x_n\rangle\}$, it is not guaranteed that $\mathcal{C} \subseteq \mathcal{C}_n$ for all n; in particular, the chosen point x_n may lie outside of $\mathcal{C}$.

Assume that we have access to a separation oracle for $\mathcal{C}$: given a point $x \notin \mathcal{C}$, the oracle outputs a non-zero vector $p \in \mathbb{R}^d$ such that $\sup_{\mathcal{C}} \langle p, \cdot \rangle \le \langle p, x \rangle$. Modify the cutting plane method as follows: if a chosen point x_n does not lie in $\mathcal{C}$, then let p_n be vector that separates x_n from $\mathcal{C}$ and instead update $\mathcal{C}_{n+1}$ to be a superset of $\mathcal{C}_n \cap \{\langle p_n, x_n \rangle \le \langle p_n, \cdot \rangle\}$. We also allow $\mathcal{C}_0 \supseteq \mathcal{C}$, so that x_0 is not necessarily feasible either. Prove that if the sets are chosen so that $\operatorname{vol}(\mathcal{C}_{n+1})/\operatorname{vol}(\mathcal{C}_n) \le \lambda < 1$ for all n, then the following assertions hold.

1. If $\operatorname{vol}(\mathcal{C}_N) < \operatorname{vol}(\mathcal{C})$, then there exists $n < N$ with $x_n \in \mathcal{C}$.

2. If $\operatorname{vol}(\mathcal{C}_N) < \operatorname{vol}(\mathcal{C})$, then there exists $n < N$ with $x_n \in \mathcal{C}$ *and* $f(x_n) - f_\star \le DL\lambda^{N/d}$. *Hint:* Define a sequence of sets $\mathcal{C}_0', \mathcal{C}_1', \mathcal{C}_2', \dots$ as follows. Start with $\mathcal{C}_0' = \mathcal{C}$ and $n_{-1} := 0$. For each $k \in \mathbb{N}$, let n_k denote the first integer greater than n_{k-1} for which $x_{n_k} \in \mathcal{C}$ and set $\mathcal{C}_{k+1}' := \mathcal{C}_k' \cap \{\langle p_{n_k}, \cdot \rangle \le \langle p_{n_k}, x_{n_k} \rangle\}$. Prove via induction that if $k(N)$ is the largest integer such that $n_{k(N)} \le N$, then $\mathcal{C}_{n_{k(N)}}' \subseteq \mathcal{C}_N$.

Exercise 6.7. Prove Lemma 6.20.

Exercise 6.8. Prove Theorem 6.22. (Use the same construction as in the proof of Theorem 6.21, but choose the parameters α and γ differently.)

7 [2/18] Frank–Wolfe

In order to overcome the lower bounds in the black-box setting, we must take advantage of additional structure in the problem. The first method we study in this vein is the *Frank–Wolfe* or *conditional gradient* method. Instead of assuming access to a projection oracle for the constraint set $\mathcal{C}$, it instead assumes access to a *linear optimization oracle* (LOO) over the set $\mathcal{C}$:

$$\text{Given } p \in \mathbb{R}^d\,, \text{ output } \operatorname*{arg\,min}_{\mathcal{C}} \langle p, \cdot \rangle\,. \tag{LOO}$$

Here, we assume that $\mathcal{C}$ is compact (bounded and closed).

The oracle equivalently maximizes the convex function $-\langle p, \cdot \rangle$ over $\mathcal{C}$, so the arg min is attained at a vertex of $\mathcal{C}$. Let us define these concepts properly.

Definition 7.1. A point $x \in \mathcal{C}$ is called an **extreme point** or a **vertex** of $\mathcal{C}$ if there do not exist $x_0, x_1 \in \mathcal{C}$ and $t \in (0, 1)$ such that $x = (1 - t)\, x_0 + t\, x_1$.

Theorem 7.2. Every compact convex set is the convex hull of its extreme points.

For example, the set of vertices of the closed unit ball $\overline{\mathrm{B}(0,1)}$ is the sphere $\partial\mathrm{B}(0,1)$. It follows that to implement (LOO), it suffices to solve $\arg\min_{\text{vertices of } \mathcal{C}} \langle p, \cdot \rangle$.

We now present the Frank–Wolfe method for minimizing f over $\mathcal{C}$:

$$x_{n+1} := (1-h_n)\, x_n + h_n \, \mathsf{LOO}(\nabla f(x_n))\,. \tag{FW}$$

Theorem 7.3 (convergence of FW)**.** Let f be convex and β-smooth over $\mathcal{C}$. Let $D := \operatorname{diam}\mathcal{C}$ and $h_n = 2/(n+2)$. Then, for any $N \ge 1$, FW satisfies

$$f(x_N) - f_\star \le \frac{2\beta D^2}{N+1}\,.$$

Proof. Let $y_n := \mathsf{LOO}(\nabla f(x_n))$. Using β-smoothness,

$$\begin{aligned}
f(x_{n+1}) - f(x_n) &\le \langle \nabla f(x_n), x_{n+1} - x_n\rangle + \frac{\beta}{2}\,\|x_{n+1} - x_n\|^2 \\
&\le h_n\,\langle \nabla f(x_n), y_n - x_n\rangle + \frac{\beta D^2 h_n^2}{2} \le h_n\,\langle \nabla f(x_n), x_\star - x_n\rangle + \frac{\beta D^2 h_n^2}{2} \\
&\le -h_n\,(f(x_n) - f_\star) + \frac{\beta D^2 h_n^2}{2}\,.
\end{aligned}$$

Rearranging,

$$f(x_{n+1}) - f_\star \le (1-h_n)\,(f(x_n) - f_\star) + \frac{\beta D^2 h_n^2}{2}\,.$$

For $h_n = 2/(n+2)$, we now prove the error bound by induction on n, where the base case $n = 0$ follows from the inequality above. If the error bound holds at iteration n, then

$$f(x_{n+1}) - f_\star \le \frac{n}{n+2}\,\frac{2\beta D^2}{n+1} + \frac{2\beta D^2}{(n+2)^2} \le \frac{2\beta D^2}{n+2}\,. \qquad \square$$

The analysis above is actually not the most natural one, since it fails to capture the affine invariance of the Frank–Wolfe algorithm (Exercise 7.1).

Besides positing different oracle access than projected gradient methods, the Frank–Wolfe method has the appealing property of producing sparse solutions. This connects with results known as approximate Carathéodory theorems. First, let us recall the classical statement of Carathéodory's theorem.

Theorem 7.4 (Carathéodory)**.** Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a compact convex set and let $x \in \mathcal{C}$. Then, x can be written as a convex combination of $d+1$ vertices of $\mathcal{C}$.

Caution: in this theorem, the choice of $d+1$ vertices of course depends on x itself. If every point in $\mathcal{C}$ could be written as a convex combination of the *same* $d+1$ vertices, this would say that $\mathcal{C}$ only has $d+1$ vertices at all.

Carathéodory's theorem says that even if a convex body has exponentially many vertices, such as the cube $[-1,1]^d$, any given point has a succinct representation using only $d+1$ vertices. However, the size of the representation grows with the ambient dimension. What happens if we relax the requirement that the representation is exact? The following simple argument, often attributed to B. Maurey, shows that the size of the representation is *dimension-free*, and the convex combination even uses equal weights.

Theorem 7.5 (approximate Carathéodory)**.** Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a compact convex set with diameter D, let $0 < \varepsilon < 1$, and let $x \in \mathcal{C}$. Then, there exist vertices $y_1, \dots, y_N \in \mathcal{C}$ with

$$\Bigl\|x - \frac{1}{N} \sum_{i=1}^N y_i\Bigr\| \le \varepsilon D\,, \qquad N \le \frac{1}{\varepsilon^2}\,.$$

Proof. By Theorem 7.4, there exist vertices $\bar y_1, \dots, \bar y_{d+1} \in \mathcal{C}$ and a probability distribution λ over $[d+1]$ such that $x = \sum_{j=1}^{d+1} \lambda_j \bar y_j$. Now consider the distribution $\mu = \sum_{j=1}^{d+1} \lambda_j \delta_{\bar y_j}$ and sample points $Y_1, \dots, Y_N \overset{\text{i.i.d.}}{\sim} \mu$. Note that each Y_i is a vertex of $\mathcal{C}$. Then, since the mean of μ is x, the usual variance calculation shows that

$$\mathbb{E}\Bigl[\Bigl\|x - \frac{1}{N} \sum_{i=1}^N Y_i\Bigr\|^2\Bigr] \le \frac{\sum_{j=1}^{d+1} \lambda_j \, \|x - \bar y_j\|^2}{N} \le \frac{D^2}{N}\,.$$

Choose N to make the right-hand side at most $\varepsilon^2 D^2$. □

The approximate Carathéodory theorem has implications, e.g., for controlling the covering numbers of polytopes. But more broadly, the proof technique is quite influential and is at the root of other important developments, e.g., the existence of neural networks of small width which approximate functions in the Barron class [Bar93].

Now comes the punchline: Franke–Wolfe renders the approximate Carathéodory theorem constructive. Indeed, suppose that the LOO always outputs a vertex. After $N-1$ iterations of FW starting from a vertex, the iterate x_{N-1} is a convex combination of at most N vertices. At the same time, if we apply Theorem 7.3 to the 2-smooth function $f : z \mapsto \|x - z\|^2$, where $x \in \mathcal{C}$ and $f_\star = 0$, we see that $\|x_{N-1} - x\|^2 \le 4D^2/N$.

The full statement of Theorem 7.3 can therefore be seen as a generalization of the approximate Carathéodory principle: the iterate of FW is a sparse combination of vertices which is approximately optimal. We next demonstrate an example in which this sparsity property is crucial.

Example 7.6 (low-rank estimation)**.** Consider the nuclear norm ball

$$\mathcal{C} = \left\{X \in \mathbb{R}^{d\times d} : \|X\|_* = \sum_{i=1}^{d} \sigma_i(X) \le 1\right\}.$$

This constraint set often arises in low-rank matrix recovery as a convex relaxation of a rank constraint. Projection onto the set $\mathcal{C}$ requires projecting the singular values onto the simplex; this requires computing a full SVD, which uses $O(d^3)$ arithmetic operations. On the other hand, since

$$\mathcal{C} = \operatorname{conv}\{uv^\top : u, v \in \mathbb{R}^d\,,\ \|u\| = \|v\| = 1\}\,,$$

the LOO for $\mathcal{C}$ involves solving, for any $P \in \mathbb{R}^{d\times d}$,

$$\operatorname*{arg\,min}_{X\in\mathcal{C}} \langle P, X\rangle = \operatorname{arg\,min}\{\langle P, uv^\top\rangle : u, v \in \mathbb{R}^d\,,\ \|u\| = \|v\| = 1\}\,.$$

Solving this amounts to computing the top singular vector of P, which is often implemented via power iteration at cost $O(d^2)$ per step. Moreover, FW yields an ε-accurate solution with rank $O(1/\varepsilon)$.

Exercises

Exercise 7.1. Show that FW is affine-invariant in the following sense. Let $A \in \mathbb{R}^{d\times d}$ be an invertible matrix. Show that the iterates $\{\hat{x}_n\}_{n\in\mathbb{N}}$ of FW applied to the problem of minimizing $\hat{x} \mapsto f(A\hat{x})$ over the set $A^{-1}\mathcal{C}$ are related to the iterates $\{x_n\}_{n\in\mathbb{N}}$ of FW on the original problem via $\hat{x}_n = Ax_n$.

8 [2/20] Proximal methods

Can we solve non-smooth problems at the same rate as smooth problems? The black-box lower bounds say *no* in general, but if the non-smooth part is "simple" in the sense that it admits an implementable proximal oracle, the answer becomes yes.

Definition 8.1. Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. The **proximal oracle** for f is the mapping $\operatorname{prox}_f : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$\operatorname{prox}_f(y) \coloneqq \operatorname*{arg\,min}_{x \in \mathbb{R}^d} \Big\{ f(x) + \frac{1}{2} \|y - x\|^2 \Big\} .$$

If f is a regular convex function, then the optimization problem defining the proximal oracle is strongly convex, so it admits a unique minimizer by Lemma 1.10 and Lemma 6.6. Note also that

$$\operatorname{prox}_{hf}(y) = \operatorname*{arg\,min}_{x \in \mathbb{R}^d} \Big\{ hf(x) + \frac{1}{2} \|y - x\|^2 \Big\} = \operatorname*{arg\,min}_{x \in \mathbb{R}^d} \Big\{ f(x) + \frac{1}{2h} \|y - x\|^2 \Big\} ,$$

where $h > 0$ plays the role of a step size.

The value of the optimization problem defining prox_f also has a name.

Definition 8.2. Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$. The **Moreau–Yosida envelope** of f with parameter $h > 0$ is the mapping $f_h : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ given by

$$f_h(y) \coloneqq \inf_{x \in \mathbb{R}^d} \Big\{ f(x) + \frac{1}{2h} \|y - x\|^2 \Big\} .$$

8.1 Algorithms and examples

The proximal oracle is a regularized version of the original optimization problem. Assuming for the moment that we can compute the proximal oracle easily, let us explore its uses for algorithm design.

The simplest algorithm is to repeatedly iterate the proximal mapping. This is known as the *proximal point method*.

$$x_{n+1} \coloneqq \operatorname{prox}_{hf}(x_n) . \tag{PPM}$$

Assume for the moment that f is smooth and that the next point x_{n+1} can be obtained from the first-order optimality condition for prox_{hf}. This leads to

$$0 = \nabla f(x_{n+1}) + \frac{1}{h} (x_{n+1} - x_n) \iff x_{n+1} = x_n - h \nabla f(x_{n+1}) .$$

Note that this is similar to the GD update, except that the gradient is evaluated at the subsequent point x_{n+1}. In numerical analysis, we say that GD is an *explicit* discretization of the gradient flow, whereas PPM is an *implicit* discretization. The advantage of an explicit method is easy of implementation; it does not require solving a (non-linear) system in order to perform an update. The advantage of an implicit method is stability.

Recall that the results in §2 for GF do not require smoothness of f, whereas the results in §3 for GD do. (We studied the non-smooth case for GD in §6.2, but it requires decreasing step sizes and averaging.) Shortly, we shall see that PPM is similar to GF, in that it also does not require smoothness.

The most powerful results using the proximal oracle, however, are for the problem of *composite optimization*. Here, the goal is to minimize a sum of functions:

$$\text{minimize} \qquad F := f + g\,.$$

We assume that f is smooth and that g is non-smooth.

Example 8.3 (LASSO as composite optimization)**.** The computation of the LASSO estimator from Example 1.3 is the canonical example of composite optimization, where

$$f : \theta \mapsto \frac{1}{2n} \sum_{i=1}^{n} (Y_i - \langle \theta, X_i \rangle)^2\,, \qquad g : \theta \mapsto \lambda \, \|\theta\|_1\,.$$

In this example, the non-smooth part is particularly simple, so we can compute its proximal oracle in closed form. First, note that it is coordinate-wise decomposable:

$$\begin{aligned} \operatorname{prox}_{\lambda\,\|\cdot\|_1}(y) &= \operatorname*{arg\,min}_{x \in \mathbb{R}^d} \Bigl\{\lambda \, \|x\|_1 + \frac{1}{2}\,\|y - x\|^2\Bigr\} \\ &= \sum_{i=1}^{d} \Bigl(\operatorname*{arg\,min}_{x[i] \in \mathbb{R}} \Bigl\{\lambda \, |x[i]| + \frac{1}{2}\,(y[i] - x[i])^2\Bigr\}\Bigr)\, e_i\,. \end{aligned}$$

Therefore, it suffices to solve the problem in dimension one. A direct computation (see Exercise 8.1) then yields

$$\operatorname{prox}_{\lambda\,|\cdot|}(y) = (|y| - \lambda)_+ \operatorname{sgn} y =: \mathsf{thresh}_\lambda(y)$$

where $(\cdot)_+ := \max\{0, \cdot\}$ denotes the positive part. The operator thesh_λ, known as the *soft thresholding operator*, reduces the magnitude of its input by λ, or to 0 if the original magnitude is less than λ. The proximal operator for $\lambda \, \|\cdot\|_1$ simply applies thresh_λ to each coordinate.

Example 8.4 (constrained optimization as composite optimization)**.** Consider the problem of minimizing a smooth function f over a closed convex set $\mathcal{C}$. We can also treat this as composite optimization with

$$g = \chi_{\mathcal{C}}\,.$$

(Recall the convex indicator defined in (6.1).) In this case, the proximal oracle for g is

$$\operatorname{prox}_{h\chi_{\mathcal{C}}}(y) = \operatorname*{arg\,min}_{x\in\mathbb{R}^d} \Big\{\chi_{\mathcal{C}}(x) + \frac{1}{2h}\,\|y-x\|^2\Big\} = \operatorname*{arg\,min}_{x\in\mathcal{C}} \Big\{\frac{1}{2h}\,\|y-x\|^2\Big\} = \pi_{\mathcal{C}}(y)\,.$$

So, the proximal oracle for $\chi_{\mathcal{C}}$ is the projection oracle for $\mathcal{C}$.

The above examples motivate the assumption that we have access to the proximal oracle for the non-smooth part g. Further examples of computable proximal oracles can be found on the website proximity-operator.net.

The algorithm we consider in this context is known as *proximal gradient descent.*

$$x_{n+1} \coloneqq \operatorname*{arg\,min}_{x\in\mathbb{R}^d} \Big\{f(x_n) + \langle \nabla f(x_n), x - x_n\rangle + g(x) + \frac{1}{2h}\,\|x-x_n\|^2\Big\}\,. \tag{PGD}$$

In other words, we take the objective function $F = f + g$ and linearize only the smooth part. The update can be rewritten as follows. By completing the square,

$$x_{n+1} = \operatorname*{arg\,min}_{x\in\mathbb{R}^d} \Big\{g(x) + \frac{1}{2h}\,\|x - x_n + h\,\nabla f(x_n)\|^2\Big\} = \operatorname{prox}_{hg}(x_n - h\,\nabla f(x_n))\,.$$

This corresponds to taking an explicit step on f, followed by an implicit step on g. It is not obvious that this algorithm converges to $x_\star$, the minimizer of $F = f + g$. However, note that if g is differentiable, then

$$x_{n+1} = x_n - h\,\nabla f(x_n) - h\,\nabla g(x_{n+1})\,.$$

If $x_n = x_\star$, then $x_{n+1} = x_\star$ is the solution since $0 = \nabla F(x_\star) = \nabla f(x_\star) + \nabla g(x_\star)$. Thus, provided that f and g are convex and differentiable, $x_\star$ is the unique fixed point.

For the LASSO problem, the iteration reads

$$x_{n+1} = \operatorname{thresh}_{\lambda h}(x_n - h\,\nabla f(x_n))\,.$$

In the literature, this is known as the *iterative shrinking-thresholding algorithm* (ISTA). For constrained optimization, proximal gradient descent is projected gradient descent.

8.2 Convergence analysis

We study the convergence of PGD, since it includes PPM as a special case (take $f = 0$).

Theorem 8.5 (convergence of PGD)**.** Let f be α_f-convex and β_f-smooth, and let g be α_g-convex. Let the step size h satisfy $h \le 1/\beta$, let x^+ denote the next iterate of PGD started from x, and let $y \in \mathbb{R}^d$. Then,

$$(1 + \alpha_g h) \|y - x^+\|^2 \le (1 - \alpha_f h) \|y - x\|^2 - 2h \left(F(x^+) - F(y)\right). \tag{8.1}$$

In particular, if we set $y = x_\star$ and iterate, it yields

$$F(x_N) - F_\star \le \frac{\alpha_f + \alpha_g}{2\,(\lambda_h^{-N} - 1)} \|x_0 - x_\star\|^2 ,$$

where $\lambda_h := (1 - \alpha_f h)/(1 + \alpha_g h)$.

Proof. Let ψ_x denote the objective function in the definition of PGD. Then, ψ_x is $(\alpha_g + 1/h)$-strongly convex with minimizer x^+, so by the quadratic growth inequality,

$$\psi_x(y) \ge \psi_x(x^+) + \frac{\alpha_g + 1/h}{2} \|y - x^+\|^2 .$$

On one hand, by α_f-convexity,

$$\psi_x(y) + f(x) = f(x) + \langle \nabla f(x), y - x\rangle + g(y) + \frac{1}{2h} \|y - x\|^2 \le F(y) + \frac{1/h - \alpha_f}{2} \|y - x\|^2 .$$

On the other hand, by β_f-smoothness,

$$\begin{aligned} \psi_x(x^+) + f(x) &= f(x) + \langle \nabla f(x), x^+ - x\rangle + g(x^+) + \frac{1}{2h} \|x^+ - x\|^2 \\ &\ge F(x^+) + \frac{1/h - \beta_f}{2} \|x^+ - x\|^2 \ge F(x^+) . \end{aligned}$$

Combining these inequalities and rearranging,

$$(1 + \alpha_g h) \|y - x^+\|^2 \le (1 - \alpha_f h) \|y - x\|^2 - 2h \left(F(x^+) - F(y)\right).$$

Note that by taking $y = x$, it yields the descent property

$$F(x^+) - F(x) \le -\frac{1 + \alpha_g h}{2h} \|x - x^+\|^2 = -\frac{(1 + \alpha_g h)\, h}{2} \|\nabla f(x) + \nabla g(x^+)\|^2 \le 0 .$$

The final bound follows from Lemma 3.5 and algebra. □

The key feature of Theorem 8.5 is that it essentially recovers the *smooth* rate for GD despite the presence of non-smoothness in the objective. Thus, for the LASSO problem (Example 8.3), we can solve it as quickly as if it were a smooth problem via ISTA.

Moreover, the one-step inequality (8.1) is the PGD analogue of the inequality (3.3) which holds for GD, and in turn, (3.3) is the only property of GD which plays a role in the proof of Nesterov acceleration (Theorem 5.10); the remainder of the proof is purely algebraic. This naturally leads to an accelerated algorithm for composite optimization.

Starting with $x_{-1} = x_0$, consider

$$x_{n+1} \coloneqq x_n + \theta_n\,(x_n - x_{n-1}) - \frac{1}{\beta}\,\mathrm{PGD}_F(x_n + \theta_n\,(x_n - x_{n-1}))\,, \tag{APGD}$$

where PGD_F denotes one step of PGD on $F = f + g$.

Theorem 8.6 (convergence of APGD)**.** Let f be convex and β-smooth, and let g be convex. Define the sequence: $\lambda_0 \coloneqq 0$ and $\lambda_{n+1} \coloneqq \frac{1}{2}\,(1 + \sqrt{1 + 4\lambda_n^2})$ for $n \in \mathbb{N}$. Set $\theta_n \coloneqq (\lambda_n - 1)/\lambda_{n+1}$. Then, APGD satisfies

$$F(x_N) - F_\star \le \frac{2\beta\,\|x_0 - x_\star\|^2}{N^2}\,.$$

When applied to LASSO, this algorithm is known as *fast ISTA* or *FISTA*. Rates in the strongly convex setting can be obtained from the reduction in Lemma 4.1.

Exercises

Exercise 8.1. Verify the computation of $\mathrm{prox}_{\lambda\,|\cdot|}$ in Example 8.3.

References

[AP24a] J. M. Altschuler and P. A. Parrilo. “Acceleration by stepsize hedging: multi-step descent and the silver stepsize schedule”. In: *J. ACM* (Dec. 2024).

[AP24b] J. M. Altschuler and P. A. Parrilo. “Acceleration by stepsize hedging: silver stepsize schedule for smooth convex optimization”. In: *Mathematical Programming* (2024).

[Bar93] A. R. Barron. “Universal approximation bounds for superpositions of a sigmoidal function”. In: *IEEE Trans. Inform. Theory* 39.3 (1993), pp. 930–945.

[Bub15] S. Bubeck. “Convex optimization: algorithms and complexity”. In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357.

[Che25] S. Chewi. *Log-concave sampling*. Available online at chewisinho.github.io. Forthcoming, 2025.

[CLM24] H. Chardon, M. Lerasle, and J. Mourtada. “Finite-sample performance of the maximum likelihood estimator in logistic regression”. In: *arXiv preprint 2411.02137* (2024).

[KNS16] H. Karimi, J. Nutini, and M. Schmidt. “Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition”. In: *European Conference on Machine Learning and Knowledge Discovery in Databases—Volume 9851*. ECML PKDD 2016. Riva del Garda, Italy: Springer-Verlag, 2016, pp. 795–811.

[LMW24] J. Liang, S. Mitra, and A. Wibisono. “On independent samples along the Langevin diffusion and the unadjusted Langevin algorithm”. In: *arXiv preprint 2402.17067* (2024).

[Łoj63] S. Łojasiewicz. “Une propriété topologique des sous-ensembles analytiques réels”. In: *Les Équations aux Dérivées Partielles (Paris, 1962)*. Éditions du Centre National de la Recherche Scientifique (CNRS), Paris, 1963, pp. 87–89.

[LRP16] L. Lessard, B. Recht, and A. Packard. “Analysis and design of optimization algorithms via integral quadratic constraints”. In: *SIAM J. Optim.* 26.1 (2016), pp. 57–95.

[Nes18] Y. Nesterov. *Lectures on convex optimization*. Vol. 137. Springer Optimization and Its Applications. Springer, 2018, pp. xxiii+589.

[NY83] A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. Translated from the Russian and with a preface by E. R. Dawson. John Wiley & Sons, Inc., New York, 1983, pp. xv+388.

[OV00] F. Otto and C. Villani. “Generalization of an inequality by Talagrand and links with the logarithmic Sobolev inequality”. In: *J. Funct. Anal.* 173.2 (2000), pp. 361–400.

[OV01] F. Otto and C. Villani. “Comment on: “Hypercontractivity of Hamilton–Jacobi equations” [J. Math. Pures Appl. (9) **80** (2001), no. 7, 669–696] by S. G. Bobkov, I. Gentil and M. Ledoux”. In: *J. Math. Pures Appl. (9)* 80.7 (2001), pp. 697–700.

[Pol63] B. T. Polyak. “Gradient methods for minimizing functionals”. In: *Ž. Vyčisl. Mat i Mat. Fiz.* 3 (1963), pp. 643–653.

[SBC16] W. Su, S. Boyd, and E. J. Candès. "A differential equation for modeling Nesterov's accelerated gradient method: theory and insights". In: *J. Mach. Learn. Res.* 17 (2016), Paper No. 153, 43.

[Ver18] R. Vershynin. *High-dimensional probability*. Vol. 47. Cambridge Series in Statistical and Probabilistic Mathematics. An introduction with applications in data science, With a foreword by Sara van de Geer. Cambridge University Press, Cambridge, 2018, pp. xiv+284.

[Vis12] N. K. Vishnoi. "$Lx = b$ Laplacian solvers and their algorithmic applications". In: *Found. Trends Theor. Comput. Sci.* 8.1-2 (2012), front matter, 1–141.