Optimization for machine learning

Irène Waldspurger
waldspurger@ceremade.dauphine.fr

September 30, 2024
October 7, 2024

Chapter 1

Gradient descent

In the whole lecture, we imagine that we want to find a minimizer of a function $f : \mathbb{R}^n \to \mathbb{R}$:

$$\text{find } x_* \text{ such that } f(x_*) = \min_{x \in \mathbb{R}^n} f(x). \tag{1.1}$$

We assume that at least one minimizer exists (which is for example guaranteed if f is continuous and coercive[1]) and denote one of them x_*.

Throughout the lecture, we will assume that f is differentiable. Minimizing non-differentiable functions is called *non-smooth optimization*. It is of course also of interest, but requires a specific theory, which we will not have time to cover here.

In the previous lectures, you have introduced gradient descent, and analyzed its convergence rate when the objective function is quadratic. The goal of this lecture is first to extend this analysis to general convex or strongly convex functions, and to discuss the choice of the stepsize. In the second part, we will present variants of gradient descent which achieve faster convergence rates through the introduction of a so-called *momentum term*.

[1] f is said to be *coercive* if $f(x) \to +\infty$ when $||x|| \to +\infty$

1.1 Classical theory of gradient descent

1.1.1 Reminders

Definition 1.1.1

For any x, the gradient of f at x is

$$\nabla f(x) \overset{def}{=} \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right) \in \mathbb{R}^n.$$

(It exists, because we have assumed that f is differentiable.)
If f is twice differentiable, we also define its Hessian at any point x as

$$\operatorname{Hess} f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{1 \le i,j \le n} \in \mathbb{R}^{n \times n}.$$

As explained in a previous lecture, the gradient at a point $x \in \mathbb{R}^n$ provides a linear approximation of f in a neighborhood of f: informally,

$$\forall y \text{ close to } x, \qquad f(y) \approx f(x) + \langle \nabla f(x), y - x \rangle . \tag{1.2}$$

Consequently, $-\nabla f(x)$ is the direction along which f decays the most around x. This motivates the definition of gradient descent: starting at any $x_0 \in \mathbb{R}^n$, we define $(x_t)_{t \in \mathbb{N}}$ by

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t), \qquad \forall t \in \mathbb{N}.$$

Here α_t is a positive number, called the *stepsize*. In this lecture, we will restrict ourselves to constant stepsizes, except in Subsection 1.1.5, where we discuss better ways to choose the stepsize.

Input: A starting point x_0, a number of iterations T, a sequence of stepsizes $(\alpha_t)_{0 \le t \le T-1}$
for $t = 0, \dots, T-1$ **do**
  Define $x_{t+1} = x_t - \alpha_t \nabla f(x_t)$.
end
Output: x_T

Algorithm 1: Gradient descent

Since our goal is to find a minimizer of f, we hope that

$$x_t \overset{t\to+\infty}{\to} x_*$$

or, at least,

$$f(x_t) \overset{t\to+\infty}{\to} f(x_*)$$

The goal of today's lecture in to understand under which assumptions on f we can guarantee that this happens, and, when it does, what is the convergence rate.

Before stating the main results, let us review what you have seen in the previous lectures about the convergence of gradient descent when f is quadratic.

Let $n > 0$ be an integer, C a symmetric $n \times n$ matrix, and $b \in \mathbb{R}^n$ a vector. Let f be defined as

$$\forall x \in \mathbb{R}^n, \quad f(x) = \frac{1}{2} \langle x, Cx \rangle + \langle x, b \rangle .$$

We assume that f is convex, which is equivalent to

$$C \succeq 0.$$

In this case, you have seen that, when $\lambda_{\min}(C) > 0$, gradient descent converges to a minimizer and the convergence rate is geometric (that is, fast). When $\lambda_{\min}(C) = 0$, this may not be true. You could nevertheless have shown that $(f(x_t))_{t\in\mathbb{N}}$ converges to $(f(x_*))$, with convergence rate at least $O(1/t)$. This is what the following theorem says.

Theorem 1.1.2

Let us consider the sequence of iterates $(x_t)_{t\in\mathbb{N}}$ generated by gradient descent with constant stepsize $\alpha < \frac{2}{\lambda_{\max}(C)}$.

- If $\lambda_{\min}(C) > 0$, it holds for any t that

  $$f(x_t) - f(x_*) \leq \rho^t (f(x_0) - f(x_*))$$

  for some $\rho \in]0; 1[$.

  (Even more, the sequence of iterates $(x_t)_{t\in\mathbb{N}}$ converges geometrically to x_*.)

- Even if $\lambda_{\min}(C) = 0$, it holds for any t that

$$f(x_t) - f(x_*) \leq \frac{||x_0 - x_*||}{4\tau t}.$$

1.1.2 Convergence guarantees for general functions

The goal of this lecture is to extend to general convex functions the results stated in the quadratic case. More precisely, we will show the following guarantees.

- When f is convex and ∇f is Lipschitz, $(f(x_t))_{t\in\mathbb{N}}$ goes to $f(x_*)$ at speed $O\left(\frac{1}{t}\right)$ (Theorem 1.1.11). This result generalizes the situation where f is quadratic and $\lambda_{\min}(C)$ may be zero.

- When f is strongly convex and ∇f is Lipschitz, $(f(x_t))_{t\in\mathbb{N}}$ goes to $f(x_*)$ at a geometric rate (Theorem 1.1.14). This result generalizes the situation where f is quadratic and $\lambda_{\min}(C) > 0$.

Smooth functions

Let us first see what we can say of the behavior of gradient descent without assuming that f is convex. Consequently, we let f be a general differentiable function, and make only one hypothesis: f has some amount of regularity. More precisely, we assume that ∇f is Lipschitz.

Definition 1.1.3 : smoothness

For any $L > 0$, we say that f is L-smooth if ∇f is L-Lipschitz, that is

$$\forall x, y \in \mathbb{R}^n, \quad ||\nabla f(x) - \nabla f(y)|| \leq L||x - y||.$$

Remark

For any $L > 0$, when f is twice differentiable, it is L-smooth if and only if, for any $x \in \mathbb{R}^n$,

$$|||\text{Hess}\, f(x)||| \leq L.$$

[The notation $|||.|||$ stands for the operator norm: for any symmetric matrix C, $|||C||| = \sup_{||u||_2=1} ||Cu||_2 = \max(|\lambda_{\min}(C)|, |\lambda_{\max}(C)|)$.]

Proof. Let us assume f to be twice differentiable.

If f is L-smooth, then, for any $x \in \mathbb{R}^n$, it holds for any $h \in \mathbb{R}^n$ that

$$\begin{aligned} |\langle \text{Hess}\, f(x)h, h\rangle| &= \left|\lim_{\epsilon\to 0} \frac{1}{\epsilon}\langle \nabla f(x+\epsilon h) - \nabla f(x), h\rangle\right| \\ &\leq ||h|| \limsup_{\epsilon\to 0} \frac{||\nabla f(x+\epsilon h) - \nabla f(x)||}{\epsilon} \\ &\leq L||h||^2, \end{aligned}$$

which implies that $|||\text{Hess}\, f(x)||| \leq L$.

Conversely, if $|||\text{Hess}\, f(x)||| \leq L$ for any $x \in \mathbb{R}^n$, it holds for any $x, y \in \mathbb{R}^n$ that

$$\begin{aligned} ||\nabla f(x) - \nabla f(y)|| &= \left|\left|\int_0^1 \text{Hess}\, f(x + t(y-x))(y-x)dt\right|\right| \\ &\leq \int_0^1 |||\text{Hess}\, f(x+t(y-x))|||\, ||y-x||dt \\ &\leq L||x-y|| \int_0^1 1dt \\ &= L||x-y||. \end{aligned}$$

□

Example 1.1.4

For any L, our quadratic function $f : x \to \frac{1}{2}\langle x, Cx\rangle + \langle x, b\rangle$ is L-smooth if and only if

$$|||C||| \leq L,$$

that is $-L \leq \lambda_{\min}(C) \leq \lambda_{\max}(C) \leq L$.

When f is smooth, the main two statements about gradient descent (with suitable constant stepsize) are given by Corollary 1.1.7.

- $(f(x_t))_{t\in\mathbb{N}}$ is nonincreasing (in particular, it converges);

- $(\nabla f(x_t))_{t\in\mathbb{N}}$ goes to 0.

Let us state and prove these results.

Lemma 1.1.5

Let $L > 0$ be fixed. If f is L-smooth, then, for any $x, y \in \mathbb{R}^n$,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}||y - x||^2.$$

Proof. For any $x, y \in \mathbb{R}^n$,

$$\begin{aligned}
f(y) &= f(x) + \int_0^1 \langle \nabla f(x + t(y - x)), y - x\rangle \, dt \\
&= f(x) + \langle \nabla f(x), y - x\rangle + \int_0^1 \langle \nabla f(x + t(y - x)) - \nabla f(x), y - x\rangle \, dt \\
&\leq f(x) + \langle \nabla f(x), y - x\rangle + \int_0^1 ||\nabla f(x + t(y - x)) - \nabla f(x)|| \, ||y - x|| dt \\
&\leq f(x) + \langle \nabla f(x), y - x\rangle + \int_0^1 Lt||y - x||^2 dt \\
&= f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}||y - x||^2.
\end{aligned}$$

□

Corollary 1.1.6

Let f be L-smooth, for some $L > 0$.
We consider gradient descent with constant stepsize: $\alpha_t = \frac{1}{L}$ for all t.
Then, for any t,

$$f(x_{t+1}) \leq f(x_t) - \frac{1}{2L}||\nabla f(x_t)||^2.$$

Corollary 1.1.7

With the same hypotheses as in the previous corollary, and additionally assuming that f is lower bounded,

1. $(f(x_t))_{t\in\mathbb{N}}$ converges to a finite value;

2. $||\nabla f(x_t)|| \overset{t\to+\infty}{\to} 0$.

Proof. The first property holds because, from Corollary 1.1.6, $(f(x_t))_{t\in\mathbb{N}}$ is a non-increasing sequence, which is lower bounded because f is. The second one is because, from the same corollary,

$$\forall t \in \mathbb{N}, \quad ||\nabla f(x_t)||^2 \leq 2L\left(f(x_t) - f(x_{t+1})\right).$$

Therefore, for any $T \in \mathbb{N}$,

$$\sum_{t=0}^{T-1} ||\nabla f(x_t)||^2 \leq 2L\left(f(x_0) - f(x_T)\right) \leq 2L(f(x_0) - \inf f).$$

Therefore, the sum $\sum_{t\geq 0} ||\nabla f(x_t)||^2$ converges, and $(||\nabla f(x_t)||)_{t\in\mathbb{N}}$ must go to zero. □

The guarantee that $||\nabla f(x_t)|| \to 0$ when $t \to +\infty$ is quite weak (although useful in some settings, as we will see in the lecture on non-convex optimization). In particular, it does not imply that $(f(x_t))_{t\in\mathbb{N}}$ converges to $f(x_*)$. To guarantee convergence to $f(x_*)$, we need stronger assumptios on f. This is where convexity comes into play.

1.1.3 Smooth convex functions

Definition 1.1.8

We say that f is convex if

$$\forall x, y \in \mathbb{R}^n, t \in [0;1], \quad f((1-t)x + ty) \leq (1-t)f(x) + tf(y). \quad (1.3)$$

Proposition 1.1.9

When f is differentiable, it is convex if and only if

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle. \quad (1.4)$$

Convexity is a strong structural property. From Equations (1.3) and (1.4), if we have access to the value of f and ∇f at a few points, then we have upper and lower bounds for the value of f at many other points. This

allows to precisely estimate the minimum and minimizer of f from only a few values. This is why optimization is possible for convex functions, while it is quite difficult for non-convex ones.

Remark

When f is twice differentiable, it is convex if and only if, for any $x \in \mathbb{R}^n$,

$$\text{Hess } f(x) \succeq 0.$$

Example 1.1.10

The quadratic function $f : x \to \frac{1}{2}\langle x, Cx\rangle + \langle x, b\rangle$ is convex if and only if $C \succeq 0$.

As announced, if we assume that f, in addition to being smooth, is convex, we can prove that $(f(x_t))_{t\in\mathbb{N}}$ converges to $f(x_*)$. Moreover, we have guarantees on the convergence rate, as described by the following theorem.

Theorem 1.1.11

Let f be convex and L-smooth, for some $L > 0$.
We consider gradient descent with constant stepsize: $\alpha_t = \frac{1}{L}$ for all t.
Then, for any $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq \frac{2L||x_0 - x_*||^2}{t+4}.$$

Proof. First step: We show that the sequence of iterates gets closer to the minimizer x_* at each step: For any $t \in \mathbb{N}$,[2]

$$||x_* - x_{t+1}|| \leq ||x_* - x_t||.$$

Let t be fixed. We find upper and lower bounds for $f(x_*)$ using the convexity and L-smoothness of f. First, by convexity,

$$f(x_*) \geq f(x_t) + \langle \nabla f(x_t), x_* - x_t\rangle = f(x_t) + L\langle x_t - x_{t+1}, x_* - x_t\rangle.$$

[2] We do not need it for our proof, but a stronger inequality actually holds: $\forall t \in \mathbb{N}, ||x_* - x_{t+1}||^2 \leq ||x_* - x_t||^2 - ||x_{t+1} - x_t||^2$.

Then, using L-smoothness through Corollary 1.1.6, and also the fact that x_* is a minimizer of f,

$$\begin{aligned} f(x_*) &\leq f(x_{t+1}) \\ &\leq f(x_t) - \frac{1}{2L}||\nabla f(x_t)||^2 \\ &= f(x_t) - \frac{L}{2}||x_{t+1} - x_t||^2. \end{aligned}$$

Combining the two bounds yields

$$\begin{aligned} f(x_t) + L\left\langle x_t - x_{t+1}, x_* - x_t\right\rangle \leq f(x_*) \leq f(x_t) - \frac{L}{2}||x_{t+1} - x_t||^2 \\ \Rightarrow \quad 2\left\langle x_t - x_{t+1}, x_* - x_t\right\rangle + ||x_{t+1} - x_t||^2 \leq 0 \\ \iff \quad ||x_* - x_{t+1}||^2 \leq ||x_* - x_t||^2. \end{aligned}$$

Second step: We can now find an inequality relating $f(x_{t+1}) - f(x_*)$ and $f(x_t) - f(x_*)$ which, applied iteratively, will prove the result. First, from corollary 1.1.6,

$$f(x_{t+1}) - f(x_*) \leq f(x_t) - f(x_*) - \frac{1}{2L}||\nabla f(x_t)||^2. \tag{1.5}$$

In addition, because f is convex, as we have already seen in the first part,

$$f(x_t) - f(x_*) \leq \left\langle \nabla f(x_t), x_t - x_*\right\rangle.$$

Using now Cauchy-Schwarz as well as the first step of the proof:

$$f(x_t) - f(x_*) \leq ||\nabla f(x_t)||\,||x_t - x_*|| \leq ||\nabla f(x_t)||\,||x_0 - x_*||.$$

In other words, $||\nabla f(x_t)|| \geq \frac{f(x_t) - f(x_*)}{||x_0 - x_*||}$. We plug this into Equation (1.5):

$$f(x_{t+1}) - f(x_*) \leq f(x_t) - f(x_*) - \frac{1}{2L}\frac{(f(x_t) - f(x_*))^2}{||x_0 - x_*||^2}.$$

Taking the inverse (and defining, by convention, $\frac{1}{0} = +\infty$), we get

$$\begin{aligned} \frac{1}{f(x_{t+1}) - f(x_*)} &\geq \frac{1}{f(x_t) - f(x_*)} \times \frac{1}{1 - \frac{1}{2L}\frac{f(x_t) - f(x_*)}{||x_0 - x_*||^2}} \\ &\geq \frac{1}{f(x_t) - f(x_*)}\left(1 + \frac{1}{2L}\frac{f(x_t) - f(x_*)}{||x_0 - x_*||^2}\right) \\ &= \frac{1}{f(x_t) - f(x_*)} + \frac{1}{2L||x_0 - x_*||^2}. \end{aligned}$$

For the second inequality, we have used the fact that $\frac{1}{1-x} \geq 1+x$ for any $x \in [0;1]$.

Consequently, by iteration, it holds for any $t \in \mathbb{N}$ that

$$\frac{1}{f(x_t)-f(x_*)} \geq \frac{1}{f(x_0)-f(x_*)} + \frac{t}{2L||x_0-x_*||^2}.$$

Corollary 1.1.6, together with the fact that $\nabla f(x_*) = 0$, ensures that

$$f(x_0)-f(x_*) \leq \frac{L}{2}||x_0-x_*||^2,$$

so for any $t \in \mathbb{N}$,

$$\begin{aligned}\frac{1}{f(x_t)-f(x_*)} &\geq \frac{2}{L||x_0-x_*||^2} + \frac{t}{2L||x_0-x_*||^2} \\ &= \frac{t+4}{2L||x_0-x_*||^2},\end{aligned}$$

that is

$$f(x_t)-f(x_*) \leq \frac{2L||x_0-x_*||^2}{t+4}.$$

□

Remark

A more general version of the theorem holds for stepsizes different from $\frac{1}{L}$. Namely, if $\alpha_t = \tau$ for all t, where $0 < \tau < \frac{2}{L}$, then it holds for all $t \in \mathbb{N}$ that

$$f(x_t)-f(x_*) \leq \frac{1}{\tau L(2-\tau L)}\frac{2L||x_0-x_*||^2}{t+4}.$$

Note that this result does not cover the case where $\tau = \frac{2}{L}$ and, indeed, gradient descent may not converge if $\tau = \frac{2}{L}$.

If we treat $||x_0-x_*||$ as a constant, the previous theorem guarantees that $f(x_t)-f(x_*) = O(1/t)$. Therefore, if we want to find an ϵ-approximate minimizer (that is, an x_t such that $f(x_t)-f(x_*) \leq \epsilon$), we can do so with $O(1/\epsilon)$ iterations of gradient descent. This is nice for problems where we do not need a high-precision solution, but when ϵ is very small, this is too much. Unfortunately, Theorem 1.1.11 is essentially optimal: There are smooth and convex functions f for which the inequality is an equality (up to minor changes in the constants).

1.1.4 Smooth strongly convex functions

We will now see a subclass of smooth convex functions for which gradient descent converges much faster than the slow $O(1/t)$ rate described in the last section: the class of smooth *strongly convex* functions. It generalizes the case of quadratic functions when the smallest eigenvalue is strictly positive (see Example 1.1.13).

Definition 1.1.12

Let $\mu > 0$ be fixed. If f is differentiable, we say that it is μ-strongly convex if, for any $x, y \in \mathbb{R}^n$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}||y - x||_2^2.$$

We observe that, if f is strongly convex, then it is convex. But strong convexity is a more powerful property than convexity: If we know the value and gradient at a point x of a strongly convex function, we know a quadratic lower bound for f (which, in particular, grows to $+\infty$ away from x) instead of a simple linear lower bound as for simply convex functions.

Remark

For any $\mu > 0$, a differentiable function f is μ-strongly convex if and only if the function $f_\mu : x \to f(x) - \frac{\mu}{2}||x||^2$ is convex.

Proof. The function f_μ is convex if and only if, for any $x, y \in \mathbb{R}^n$,

$$\begin{aligned}
& f_\mu(y) \geq f_\mu(x) + \langle \nabla f_\mu(x), y - x \rangle \,; \\
\iff \quad & f(y) - \frac{\mu}{2}||y||_2^2 \geq f(x) - \frac{\mu}{2}||x||_2^2 + \langle \nabla f(x) - \mu x, y - x \rangle \,; \\
\iff \quad & f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\left(||y||_2^2 - 2\langle x, y - x \rangle - ||x||_2^2\right) \,; \\
\iff \quad & f(y) \geq f(x) + \langle \nabla f(x) - \mu x, y - x \rangle + \frac{\mu}{2}||y - x||_2^2.
\end{aligned}$$

□

Remark

As a consequence from the previous remark, as well as the one following Definition 1.1.8, a twice differentiable function f is μ-strongly convex if and only if, for any $x \in \mathbb{R}^n$,

$$\text{Hess}\, f(x) - \mu \text{Id} \succeq 0,$$

or, in other words, all eigenvalues of Hess $f(x)$ are larger than μ.

Example 1.1.13

We consider again the quadratic function $f : x \in \mathbb{R}^n \to \frac{1}{2}\langle x, Cx\rangle + \langle x, b\rangle$. Its Hessian at any point is C. We denote $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ the ordered eigenvalues of C. From the previous remark, if $\lambda_n > 0$, f is λ_n-strongly convex. If $\lambda_n \leq 0$, f is not μ-strongly convex, whatever the value of $\mu > 0$.

Theorem 1.1.14

Let $0 < \mu < L$ be fixed. Let f be L-smooth and μ-strongly convex. We consider gradient descent with constant stepsize: $\alpha_t = \frac{1}{L}$ for all t. Then, for any $t \in \mathbb{N}$,

$$||x_t - x_*||_2 \leq \left(1 - \frac{\mu}{L}\right)^t ||x_0 - x_*||_2; \tag{1.6}$$
$$f(x_t) - f(x_*) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^{2t} ||x_0 - x_*||_2^2.$$

Proof. It is enough to prove Equation (1.6). Indeed, if this equation holds, it implies (from Lemma 1.1.5 and because $\nabla f(x_*) = 0$),

$$f(x_t) \leq f(x_*) + \frac{L}{2}||x_t - x_*||_2^2. \Rightarrow \quad f(x_t) - f(x_*) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^{2t} ||x_* - x_0||_2^2.$$

To prove Equation (1.6), it suffices to prove that, for any $t \in \mathbb{N}$,

$$||x_{t+1} - x_*||_2 \leq \left(1 - \frac{\mu}{L}\right) ||x_t - x_*||_2.$$

Let us fix $t \in \mathbb{N}$ and establish this inequality.

Given that $x_{t+1} = x_t - \frac{1}{L}\nabla f(x_t)$, we must simply upper bound

$$||x_{t+1} - x_*||_2 = \frac{1}{L}\,||\nabla f(x_t) - L(x_t - x_*)||_2$$

with a multiple of $||x_t - x_*||_2$.

We must therefore establish an inequality involving only x_t, x_* and $\nabla f(x_t)$. For this, we first look at which inequlities we can write on these quantities. In particular, we consider the inequality defining μ-strong convexity (Definition 1.1.12), at $x = x_t$ or $x = x_*$: for all $y \in \mathbb{R}^n$,

$$f(y) \geq f(x_t) + \langle \nabla f(x_t), y - x_t\rangle + \frac{\mu}{2}||y - x_t||_2^2; \tag{1.7a}$$

$$f(y) \geq f(x_*) + \frac{\mu}{2}||y - x_*||_2^2. \tag{1.7b}$$

And considering also the inequality of Lemma 1.1.5, we have, for all $y \in \mathbb{R}^n$,

$$f(y) \leq f(x_t) + \langle \nabla f(x_t), y - x_t\rangle + \frac{L}{2}||y - x_t||_2^2; \tag{1.8a}$$

$$f(y) \leq f(x_*) + \frac{L}{2}||y - x_*||_2^2. \tag{1.8b}$$

In particular, for all $y \in \mathbb{R}^n$, combining (1.7a) and (1.8b), it holds that

$$f(x_*) + \frac{L}{2}||y - x_*||_2^2 - f(x_t) - \langle \nabla f(x_t), y - x_t\rangle - \frac{\mu}{2}||y - x_t||_2^2 \geq 0.$$

The minimum of this expression is reached at $y = \frac{Lx_* - \mu x_t + \nabla f(x_t)}{L-\mu}$, and its value is

$$f(x_*) - f(x_t) - \frac{||\nabla f(x_t)||_2^2}{2(L-\mu)} - \left\langle \nabla f(x_t), \frac{L(x_* - x_t)}{L-\mu}\right\rangle - \frac{L\mu}{2(L-\mu)}||x_t - x_*||_2^2 \geq 0.$$

Similarly, combining (1.7b) and (1.8a), we get for all $y \in \mathbb{R}^n$

$$f(x_t) + \langle \nabla f(x_t), y - x_t\rangle + \frac{L}{2}||y - x_t||_2^2 - f(x_*) - \frac{\mu}{2}||y - x_*||_2^2 \geq 0.$$

The minimum of this expression is reached at $y = \frac{Lx_t - \mu x_* - \nabla f(x_t)}{L-\mu}$, and its value is

$$f(x_t) - f(x_*) - \frac{||\nabla f(x_t)||_2^2}{2(L-\mu)} + \left\langle \nabla f(x_t), \frac{\mu(x_t - x_*)}{L-\mu}\right\rangle - \frac{L\mu}{2(L-\mu)}||x_t - x_*||_2^2 \geq 0.$$

If we combine the two minima, we get

$$\begin{aligned}
&(L+\mu)\langle \nabla f(x_t), x_t - x_* \rangle \geq ||\nabla f(x_t)||_2^2 + L\mu ||x_t - x_*||_2^2 \\
\iff \quad &\left|\left| \nabla f(x_t) - \frac{L+\mu}{2}(x_t - x_*) \right|\right|_2 \leq \frac{L-\mu}{2}||x_t - x_*||_2.
\end{aligned}$$

Together with the triangular inequality, this proves the result:

$$\begin{aligned}
&\frac{1}{L}||\nabla f(x_t) - L(x_t - x_*)||_2 \\
&\quad \leq \frac{1}{L}\left|\left| \nabla f(x_t) - \frac{L+\mu}{2}(x_t - x_*) \right|\right|_2 + \frac{1}{L}\left|\left| \frac{L+\mu}{2}(x_t - x_*) - L(x_t - x_*) \right|\right|_2 \\
&\quad \leq \frac{L-\mu}{2L}||x_t - x_*||_2 + \frac{L-\mu}{2L}||x_t - x_*||_2 \\
&\quad = \left(1 - \frac{\mu}{L}\right)||x_t - x_*||_2.
\end{aligned}$$

□

Hence, when f is smooth and strongly convex, $(f(x_t) - f(x_*))_{t\in\mathbb{N}}$ decays geometrically, with rate at least $\left(1 - \frac{\mu}{L}\right)^2$. An ϵ-approximate minimizer can be found in $O((\log \epsilon)/\log(1 - \mu/L))$ gradient descent iterations, much less than the $O(\epsilon)$ obtained without the strong convexity assumption.

We call $\frac{L}{\mu} \geq 1$ the *condition number* of f. The closer to 1 it is, the faster the convergence.

Remark

The rate $\left(1 - \frac{\mu}{L}\right)^2$ in the previous theorem is tight, in the sense that it is not possible to establish the same theorem for a strictly smaller convergence rate. Indeed, when applied to a μ-strongly convex and L-smooth *quadratic* function, the gradient descent iterates go to zero at this exact rate.

Remark : other constant stepsizes

A more general theorem holds for stepsizes different from $\frac{1}{L}$. More precisely, if $\alpha_t = \tau$ for all t, where $0 < \tau < \frac{2}{L}$, then it holds for any

$t \in \mathbb{N}$ that

$$||x_t - x_*||_2 \leq \max\left(|1-\tau\mu|, |1-\tau L|\right)^t ||x_0 - x_*||_2$$

In this expression, the right-hand side is minimal when $\tau = \frac{2}{\mu+L}$. This value is the optimal stepsize for gradient descent on strongly convex functions.

1.1.5 Choice of stepsizes

Properly choosing the stepsizes $(\alpha_t)_{t\in\mathbb{N}}$ is crucial: if they are too large, then x_{t+1} is outside the domain where the approximation (1.2) holds, and the algorithm may diverge. On the contrary, if they are too small, x_t needs many time steps to move away from x_0, and convergence can be slow.

What a good stepsize choice is depends on the properties of f. Let us however mention some common strategies:

1. *Fixed schedule*: the stepsizes are chosen in advance; α_t generally depends on t through a simple equation, like

$$\forall t, \quad \alpha_t = \eta, \text{ for some } \eta > 0, \qquad \text{(Constant stepsize)}$$

$$\text{or} \quad \forall t, \quad \alpha_t = \frac{1}{t+1}. \qquad \text{(Monotonically decreasing stepsize)}$$

2. *Exact line search*: for any t, choose α_t such that

$$f(x_t - \alpha_t \nabla f(x_t)) = \min_{a\in\mathbb{R}} f(x_t - a\nabla f(x_t)).$$

3. *Backtracking line search*: unless f has very particular properties, it is a priori difficult to minimize f on a line. The exact line search strategy is therefore difficult to implement. Instead, one can simply choose α_t such that $f(x_t - \alpha_t\nabla f(x_t))$ is "sufficiently smaller than $f(x_t)$" The approximation (1.2) implies, for α_t small enough,

$$f(x_t - \alpha_t \nabla f(x_t)) \approx f(x_t) - \alpha_t ||\nabla f(x_t)||^2.$$

If we consider that "being sufficiently smaller than $f(x_t)$" means that the previous approximation holds, up to the introduction of a multiplicative constant (which is known as *Armijo's condition*), the following algorithm describes a way to find a suitable α_t.

Input: Parameters $c, \tau \in]0;1[$, maximal stepsize value a_{max}
Define $\alpha_t = a_{max}$.
while $f(x_t - \alpha_t \nabla f(x_t)) > f(x_t) - c\alpha_t ||\nabla f(x_t)||^2$ **do**
| Set $\alpha_t = \tau \alpha_t$.
end
Output: α_t

Algorithm 2: Backtracking line search

1.1.6 Exercise

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function. We assume that

1. f is convex;
2. f has a global minimizer x_*;
3. f is differentiable and, for any $x \in \mathbb{R}^d$,

$$||\nabla f(x)||_2 \leq 1.$$

We fix a starting point x_0 and run gradient descent from this point, with a sequence of positive stepsizes $(h_k)_{k \in \mathbb{N}}$:

$$x_{k+1} = x_k - h_k \nabla f(x_k).$$

1. a) Show that, for any $k \in \mathbb{N}$,

$$f(x_k) - f(x_*) \leq \langle \nabla f(x_k), x_k - x_* \rangle .$$

b) Show that, for any $k \in \mathbb{N}$,

$$||x_{k+1} - x_*||_2^2 \leq ||x_k - x_*||_2^2 - 2h_k(f(x_k) - f(x_*)) + h_k^2 ||\nabla f(x_k)||_2^2.$$

c) Show that, for any $n \in \mathbb{N}$,

$$2\sum_{k=0}^{n} h_k(f(x_k) - f(x_*)) \leq ||x_0 - x_*||_2^2 - ||x_{n+1} - x_*||_2^2 + \sum_{k=0}^{n} h_k^2 ||\nabla f(x_k)||_2^2.$$

d) For any n, let $k_n \in \{0, \dots, n\}$ be such that

$$f(x_{k_n}) = \min_{k=0,\dots,n} f(x_k).$$

Show that, for any n,

$$2(f(x_{k_n}) - f(x_*)) \left(\sum_{k=0}^{n} h_k \right) \leq ||x_0 - x_*||_2^2 - ||x_{n+1} - x_*||_2^2 + \sum_{k=0}^{n} h_k^2 ||\nabla f(x_k)||_2^2.$$

e) Show that, for any n,

$$2(f(x_{k_n}) - f(x_*)) \left(\sum_{k=0}^{n} h_k \right) \leq ||x_0 - x_*||_2^2 + \sum_{k=0}^{n} h_k^2.$$

2. In this question, we assume that, for any k, $h_k = \frac{1}{\sqrt{k+1}}$. Show that, for any n,

$$f(x_{k_n}) - f(x_*) \leq \frac{||x_0 - x_*||_2^2 + 2 + \log(n)}{\sqrt{n+2}}.$$

Hint: You can use the fact that, for any n,

$$\sum_{k=1}^{n+1} \frac{1}{k} \leq 2 + \log(n) \quad \text{and} \quad \sum_{k=1}^{n+1} \frac{1}{\sqrt{k}} \geq \frac{\sqrt{n+2}}{2}.$$

3. In this question, we assume that the sequence of stepsizes is constant: there exists $\eta > 0$ such that, for any $k \in \mathbb{N}$, $h_k = \eta$.

Give an example of a function f satisfying properties 1, 2, 3, and a starting point x_0 such that

$$f(x_{k_n}) - f(x_*) \overset{n \to +\infty}{\not\to} 0.$$

Hint: Define

$$f : x \in \mathbb{R} \quad \to \quad \begin{array}{ll} |x| - \frac{\epsilon}{2} & \text{if } |x| \geq \epsilon; \\ \frac{x^2}{2\epsilon} & \text{if } |x| \leq \epsilon, \end{array}$$

for some $\epsilon > 0$ small enough.

Solution

1. a) Let k be fixed. We apply the characterization of convexity for differentiable functions: at x_*, f is above its tangent at x_k, that is

$$f(x_*) \geq f(x_k) + \langle \nabla f(x_k), x_* - x_k \rangle ,$$

which is equivalent to the desired inequality.
b) For any k,

$$\begin{aligned} ||x_{k+1} - x_*||_2^2 &= ||x_k - x_* - h_k \nabla f(x_k)||_2^2 \\ &= ||x_k - x_*||_2^2 - 2h_k \langle \nabla f(x_k), x_k - x_* \rangle + h_k^2 ||\nabla f(x_k)||_2^2 \\ &\overset{1.a)}{\leq} ||x_k - x_*||_2^2 - 2h_k(f(x_k) - f(x_*)) + h_k^2 ||\nabla f(x_k)||_2^2. \end{aligned}$$

c) We deduce from the previous question that, for any $k \in \mathbb{N}$,

$$2h_k(f(x_k) - f(x_*)) \leq ||x_k - x_*||_2^2 - ||x_{k+1} - x_*||_2^2 + h_k^2 ||\nabla f(x_k)||_2^2.$$

Therefore, for any $n \in \mathbb{N}$,

$$\begin{aligned} 2\sum_{k=0}^{n} h_k(f(x_k) - f(x_*)) &\leq \sum_{k=0}^{n} (||x_k - x_*||_2^2 - ||x_{k+1} - x_*||_2^2) + \sum_{k=0}^{n} h_k^2 ||\nabla f(x_k)||_2^2 \\ &= ||x_0 - x_*||_2^2 - ||x_{n+1} - x_*||^2 + \sum_{k=0}^{n} h_k^2 ||\nabla f(x_k)||_2^2. \end{aligned}$$

d) Let n be fixed. For any $k \leq n$, we have, from the definition of k_n, $f(x_{k_n}) \leq f(x_k)$. As a consequence, for any $k \leq n$,

$$2h_k(f(x_{k_n}) - f(x_*)) \leq 2h_k(f(x_k) - f(x_k)).$$

and

$$\begin{aligned} 2(f(x_{k_n}) - f(x_*)) \left(\sum_{k=0}^{n} h_k \right) &\leq 2 \sum_{k=0}^{n} h_k(f(x_k) - f(x_k)) \\ &\overset{1.c)}{\leq} ||x_0 - x_*||_2^2 - ||x_{n+1} - x_*||^2 + \sum_{k=0}^{n} h_k^2 ||\nabla f(x_k)||_2^2. \end{aligned}$$

e) From our third assumption on f, $||\nabla f(x_k)||_2 \leq 1$ for any $k \in \mathbb{N}$. Therefore, for any $n \in \mathbb{N}$,

$$\sum_{k=0}^{n} h_k^2 ||\nabla f(x_k)||_2^2 \leq \sum_{k=0}^{n} h_k^2.$$

Since, in addition, $-||x_{n+1} - x_*||_2^2 \leq 0$, we deduce from question 1.d) that

$$2(f(x_{k_n}) - f(x_*)) \left(\sum_{k=0}^{n} h_k \right) \leq ||x_0 - x_*||_2^2 + \sum_{k=0}^{n} h_k^2.$$

2. For any $n \in \mathbb{N}$,

$$\sum_{k=0}^{n} h_k^2 = \sum_{k=1}^{n+1} \frac{1}{k} \leq 2 + \log(n);$$
$$\sum_{k=0}^{n} h_k = \sum_{k=1}^{n+1} \frac{1}{\sqrt{k}} \geq \frac{\sqrt{n+2}}{2}.$$

Plugging these inequalities into the one established at question 1.e) yields

$$\begin{aligned}(f(x_{k_n}) - f(x_*))\sqrt{n+2} \leq 2(f(x_{k_n}) - f(x_*)) \left(\sum_{k=0}^{n} h_k \right) &\leq ||x_0 - x_*||_2^2 + \sum_{k=0}^{n} h_k^2 \\ &\leq ||x_0 - x_*||_2^2 + 2 + \log(n).\end{aligned}$$

Therefore,

$$f(x_{k_n}) - f(x_*) \leq \frac{||x_0 - x_*||_2^2 + 2 + \log(n)}{\sqrt{n+2}}.$$

3. We set $\epsilon = \frac{\eta}{2}$ and define f as suggested:

$$f : x \in \mathbb{R} \quad \rightarrow \quad \begin{cases} |x| - \dfrac{\epsilon}{2} & \text{if } |x| \geq \epsilon; \\ \dfrac{x^2}{2\epsilon} & \text{if } |x| \leq \epsilon, \end{cases}$$

Let us show that f satisfies properties 1, 2, 3.

We start with property 2. For any $x \in \mathbb{R}$ such that $|x| \geq \epsilon$,

$$f(x) \geq \epsilon - \frac{\epsilon}{2} > 0.$$

For any $x \in \mathbb{R}$ such that $|x| < \epsilon$,

$$f(x) = \frac{x^2}{2\epsilon} \geq 0.$$

Therefore, f is nonnegative over $\mathbb{R}$. Since $f(0) = 0$, it implies that $x_* = 0$ is a global minimizer of f.

Let us now show that f is differentiable and compute its derivative. The function $|.|$ is differentiable over $\mathbb{R} - \{0\}$ so f is differentiable over $]-\infty; -\epsilon] \cup [\epsilon; +\infty[$, with derivative

$$\begin{aligned} f'(x) &= -1 \quad \forall x \in]-\infty; -\epsilon]; \\ f'(x) &= 1 \quad \forall x \in [\epsilon; +\infty[. \end{aligned}$$

(The derivative is only a left derivative when $x = -\epsilon$ and a right derivative when $x = \epsilon$.)

The square function is differentiable over $\mathbb{R}$ so f is differentiable over $[-\epsilon; \epsilon]$, with derivative

$$f'(x) = \frac{x}{\epsilon} \quad \forall x \in [-\epsilon; \epsilon].$$

(The derivative is only a right derivative when $x = -\epsilon$ and a left derivative when $x = \epsilon$.)

Since the left and right derivatives coincide in $x = -\epsilon$ and $x = \epsilon$, the function f is differentiable at $-\epsilon$ and ϵ and therefore differentiable over $\mathbb{R}$.

For any x such that $|x| \geq \epsilon$, we have $|f'(x)| = 1$ and, for any x such that $|x| \leq \epsilon$, we have $|f'(x)| = \frac{|x|}{\epsilon} \leq 1$. As a consequence, the norm of the gradient (that is, in this case, the derivative), is always at most 1 and Property 3 holds.

Now that we have computed the derivative, we can easily show that f is convex: its derivative is continuous, nondecreasing (actually constant) over $]-\infty; -\epsilon]$, increasing over $[-\epsilon; \epsilon]$, nondecreasing again over $[\epsilon; +\infty[$. Therefore, the derivative is nondecreasing over $\mathbb{R}$ and f is convex.

We consider the starting point $x_0 = \frac{\eta}{2} = \epsilon$. With this definition,

$$\begin{aligned} x_1 &= x_0 - h_0 f'(x_0) \\ &= \epsilon - \eta \times 1 \\ &= -\epsilon \end{aligned}$$

and

$$\begin{aligned} x_2 &= x_1 - h_0 f'(x_1) \\ &= -\epsilon - \eta \times (-1) \\ &= \epsilon. \end{aligned}$$

We can iteratively reapply this result and we obtain that $x_k = -\epsilon$ for all odd k and $x_k = \epsilon$ for all even k. In particular, $x_k \not\to x_* = 0$ when $k \to +\infty$.

1.2 Gradient descent with momentum

Gradient descent is by far the most well-known optimization algorithm. Because of its simplicity and flexibility, it is a method of choice for many problems. However, it is oftentimes unconveniently slow. In this lecture, we will see that it is possible to speed up gradient descent by incorporating in it a term called *momentum*. We will present two forms of momentum, leading to the following two algorithms:

- heavy ball, which is the simplest form of gradient descent with momentum, and already provides significant speed-ups,
- Nesterov's method, which is slightly more complex, but performs much better than gradient descent on a larger range of problems than heavy ball.

1.2.1 Motivation of momentum

In this section, we motivate the introduction of momentum: we consider a simple function f for which gradient descent converges slowly, explain why convergence is slow, and why momentum can speed it up.

Let f be a simple quadratic function over $\mathbb{R}^2$:

$$\forall (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = \frac{1}{2}\left(\lambda_1 x_1^2 + \lambda_2 x_2^2\right),$$

for parameters $0 < \lambda_1 < \lambda_2$. The unique minimizer of f is

$$x_* = (0, 0).$$

The gradient of f is

$$\forall (x_1, x_2) \in \mathbb{R}^2, \quad \nabla f(x_1, x_2) = (\lambda_1 x_1, \lambda_2 x_2).$$

If we run gradient descent with constant stepsize $\alpha > 0$, the relation between iterates $x_t = (x_{t,1}, x_{t,2})$ and $x_{t+1} = (x_{t+1,1}, x_{t+1,2})$ is

$$\begin{aligned}(x_{t+1,1}, x_{t+1,2}) &= x_t - \alpha \nabla f(x_t) \\ &= (x_{t,1}, x_{t,2}) - \alpha(\lambda_1 x_{t,1}, \lambda_2 x_{t,2}) \\ &= ((1 - \alpha\lambda_1)x_{t,1}, (1 - \alpha\lambda_2)x_{t,2}) \,.\end{aligned}$$

Since we want the iterates to go as fast as possible to zero, we would like to choose α such that

$$|1 - \alpha\lambda_1| \ll 1 \quad \text{and} \quad |1 - \alpha\lambda_2| \ll 1.$$

If λ_1 and λ_2 are of the same order, this is fine: it suffices to pick α of the order of $\frac{1}{\lambda_1} \sim \frac{1}{\lambda_2}$.

But if λ_1 is much smaller than λ_2 (that is, the problem is *ill-conditioned*), there is no good choice of α. If we set $\alpha \approx \frac{1}{\lambda_1}$, then

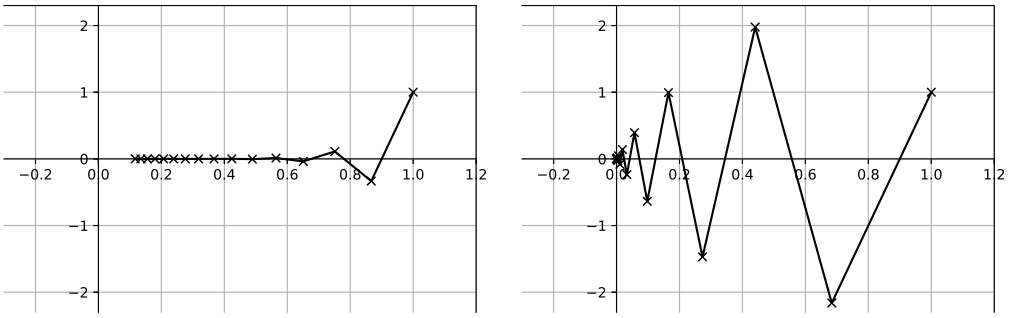
$$1 - \alpha\lambda_2 = 1 - \frac{\lambda_2}{\lambda_1} < -1$$

and the second coordinate of the iterates, $x_{t,2}$, diverges when $t \to \infty$. If, on the other hand, we set $\alpha \approx \frac{1}{\lambda_2}$, then the second coordinate goes to 0, and fast, but the first one converges very slowly:

$$1 - \alpha\lambda_1 = 1 - \frac{\lambda_1}{\lambda_2} \approx 1.$$

In this situation, gradient descent is slow. Figure 1.1a displays the first fifteen iterates in the case where $\lambda_1 = 0.1$ and $\lambda_2 = 1$, for $\alpha = 4/3$ (that is, of the order of $\frac{1}{\lambda_2}$). As expected, the second coordinate goes fast to zero, but the first one decays only slowly.

A possible remedy to this slow convergence is to use the information given by the past gradients when we define x_{t+1} from x_t: instead of moving in the direction given by $-\nabla f(x_t)$, we move in a direction m_{t+1} which is a (weighted) average between $-\nabla f(x_t)$ and the previous gradients $-\nabla f(x_0)$, ..., $-\nabla f(x_{t-1})$. Concretely, this yields the following iteration formula:



(a) Standard gradient descent (b) Gradient descent with momentum

Figure 1.1: First 15 iterates of gradient descent, for $\lambda_1 = 0.1, \lambda_2 = 1$

$$\begin{aligned} m_{t+1} &= \gamma_t m_t + (1-\gamma_t)\nabla f(x_t), \\ x_{t+1} &= x_t - \alpha_t m_{t+1}. \end{aligned}$$

Here, γ_t and α_t are respectively the momentum and stepsize parameters. The quantity m_t, which is the average of all gradients until step t, is called *momentum*.

Remark

An equivalent iteration formula is

$$x_{t+1} = x_t - \tilde{\alpha}_t \nabla f(x_t) + \tilde{\beta}_t (x_t - x_{t-1}), \tag{1.10}$$

with $\tilde{\alpha}_t = \alpha_t(1-\gamma_t)$ and $\tilde{\beta}_t = \frac{\alpha_t \gamma_t}{\alpha_{t-1}}$.

Proof of the remark. From the second equation in the iteration formula:

$$\begin{aligned} &\forall t \in \mathbb{N}, \quad m_{t+1} = \frac{x_t - x_{t+1}}{\alpha_t}, \\ \Rightarrow \quad &\forall t \in \mathbb{N} - \{0\}, \quad m_t = \frac{x_{t-1} - x_t}{\alpha_{t-1}}. \end{aligned}$$

We plug these equalities into the first iteration formula:

$$\forall t \in \mathbb{N} - \{0\}, \quad \frac{x_t - x_{t+1}}{\alpha_t} = \gamma_t \left(\frac{x_{t-1} - x_t}{\alpha_{t-1}} \right) + (1 - \gamma_t) \nabla f(x_t),$$
$$\Rightarrow \quad \forall t \in \mathbb{N} - \{0\}, \quad x_{t+1} = x_t - \alpha_t (1 - \gamma_t) \nabla f(x_t) + \frac{\alpha_t \gamma_t}{\alpha_{t-1}} (x_t - x_{t-1}) .$$

□

Using momentum instead of plain gradient in the iteration formula allows to use a larger stepsize. Indeed, for large stepsizes, $\alpha_t \nabla f(x_t)$ diverges when t grows, which causes the divergence of plain gradient descent. But it is possible that $\alpha_t m_t$ stays bounded, in which case gradient descent with momentum does not diverge: $\alpha_t m_t$ is an average of potentially large gradients pointing to different directions, which may therefore compensate each other. This can be seen in Figure 1.1b: compared to Figure 1.1a, the stepsize is larger; consequently, the first coordinate converges faster towards zero, but the second coordinate does not diverge.

1.2.2 Heavy ball

The simplest version of gradient descent with momentum is when the momentum and stepsize parameters are constant. It is due to Polyak, and often called *heavy ball*[3].

[3]The name comes from the fact that the momentum term can be seen as an inertia term, which reminds of the movement of a heavy ball falling down a mountain towards a valley.

Input: Starting point x_0, number of iterations T, stepsize α, momentum parameter γ.
Set $m_0 = \nabla f(x_0)$;
for $t = 0, \dots, T-1$ **do**
  define

$$m_{t+1} = \gamma m_t + (1-\gamma)\nabla f(x_t);$$
$$x_{t+1} = x_t - \alpha m_{t+1}.$$

end
return x_T

Algorithm 3: Heavy ball

For proper choices of parameters, heavy ball exhibits a faster convergence rate than plain gradient descent on many natural problems. We will prove this fact for quadratic strongly convex functions.

Theorem 1.2.1 : heavy ball - quadratic case

Let $0 < \mu < L$ be fixed. Let f be a quadratic function, which is L-smooth and μ-strongly convex. We set

$$\alpha = \frac{1}{\sqrt{\mu L}}, \quad \gamma = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\right)^2.$$

There exists a constant $C_{\mu,L} > 0$ such that, for any $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq C_{\mu,L} t^2 \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\right)^{2t} ||x_0 - x_*||^2.$$

Before proving the theorem, let us compare the convergence rate with gradient descent. From Theorem 1.1.14, gradient descent converges geometrically, with decay rate

$$1 - \frac{\mu}{L}.$$

Theorem 1.2.1, on the other hand, guarantees for heavy ball a convergence

with decay rate

$$\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2 \approx 1 - 4\sqrt{\frac{\mu}{L}} \quad \text{when } \mu \ll L.$$

For ill-conditioned problems, $\sqrt{\frac{\mu}{L}}$ is much larger than $\frac{\mu}{L}$, resulting in a significant speed-up. As an example, if $\frac{\mu}{L} = 0.01$, dividing $f(x_t) - f(x_*)$ by a factor 10 necessitates around

$$\frac{\ln(10)}{-\ln\left(1-\frac{\mu}{L}\right)} \approx 230$$

iterations with gradient descent, and only

$$\frac{\ln(10)}{-\ln\left(\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2\right)} \approx 6$$

with heavy ball.

Proof of Theorem 1.2.1. Up to a change of coordinates, we can assume that f is of the form

$$f(x_1, \dots, x_n) = \frac{1}{2}\left(\lambda_1 x_1^2 + \cdots + \lambda_n x_n^2\right),$$

where

$$L \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq \mu > 0$$

are the eigenvalues of the matrix representing f.

Denoting $x_t = (x_{t,1}, x_{t,2}, \dots, x_{t,n})$, we have, for each t,

$$\nabla f(x_t) = (\lambda_1 x_{t,1}, \dots, \lambda_n x_{t,n}),$$

hence the evolution equation of heavy ball is, for each $t \in \mathbb{N}$,

$$\begin{aligned} \forall k \leq n, \quad & m_{t+1,k} = \gamma m_{t,k} + (1-\gamma)\lambda_k x_{t,k}; \\ & x_{t+1,k} = x_{t,k} - \alpha m_{t+1,k} = (1 - \alpha(1-\gamma)\lambda_k) x_{t,k} - \alpha\gamma m_{t,k}. \end{aligned}$$

This can be written in matricial form: for each $t \in \mathbb{N}, k \in \{1, \dots, n\}$,

$$\begin{pmatrix} m_{t+1,k} \\ x_{t+1,k} \end{pmatrix} = M_k \begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix}, \quad \text{with } M_k = \begin{pmatrix} \gamma & (1-\gamma)\lambda_k \\ -\alpha\gamma & 1 - \alpha(1-\gamma)\lambda_k \end{pmatrix}$$

$$\Rightarrow \quad \begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix} = M_k^t \begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix}.$$

For any k, the matrix M_k can be triangularized in a (complex) orthonormal basis: for some unitary matrix G_k, we can write it under the form

$$M_k = G_k \begin{pmatrix} \sigma_k^{(1)} & g_k \\ 0 & \sigma_k^{(2)} \end{pmatrix} G_k^{-1}.$$

For all $t \in \mathbb{N}$,

$$\begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix} = G_k \begin{pmatrix} (\sigma_k^{(1)})^t & g_{t,k} \\ 0 & (\sigma_k^{(2)})^t \end{pmatrix} G_k^{-1} \begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix},$$
$$\text{with} \quad g_{t,k} = ((\sigma_k^{(1)})^{t-1} + (\sigma_k^{(1)})^{t-2}\sigma_k^{(2)} + \cdots + (\sigma_k^{(2)})^{t-1})g_k.$$

As G_k is unitary, it does not change the norm:

$$\left\| \begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix} \right\| \leq \left\| \left\| \begin{pmatrix} (\sigma_k^{(1)})^t & g_{k,t} \\ 0 & (\sigma_k^{(2)})^t \end{pmatrix} \right\| \right\| \left\| \begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix} \right\|.$$

(The triple bar denotes the spectral norm.)

For some constants $C, C' > 0$, the spectral norm can be upper bounded by

$$\begin{aligned} \left\| \left\| \begin{pmatrix} (\sigma_k^{(1)})^t & g_{k,t} \\ 0 & (\sigma_k^{(2)})^t \end{pmatrix} \right\| \right\| &\leq C \max\left(|\sigma_k^{(1)}|^t, |\sigma_k^{(2)}|^t, |g_{k,t}| \right) \\ &\leq C' t \max\left(|\sigma_k^{(1)}|, |\sigma_k^{(2)}| \right)^t. \end{aligned}$$

We must compute $\max\left(|\sigma_k^{(1)}|, |\sigma_k^{(2)}| \right)$, where we recall that $\sigma_k^{(1)}, \sigma_k^{(2)}$ are the eigenvalues of M_k. These eigenvalues are the roots of the characteristic polynomial of M_k. A (slightly tedious) computation shows that the polynomial has a negative discriminant. The eigenvalues are therefore complex and conjugate one from each other:

$$|\sigma_k^{(1)}|^2 = |\sigma_k^{(2)}|^2 = \sigma_k^{(1)}\sigma_k^{(2)} = \det(M_k) = \gamma.$$

In particular, $\max\left(|\sigma_k^{(1)}|, |\sigma_k^{(2)}|\right) = \sqrt{\gamma}$, and we get

$$\begin{aligned}
&\forall k, \quad \left|\left|\begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix}\right|\right| \leq C' t \gamma^{t/2} \left|\left|\begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix}\right|\right| \\
\Rightarrow \quad &|x_{t,k}| \leq C' t \gamma^{t/2} \sqrt{x_{0,k}^2 + m_{0,k}^2} \leq C' t \gamma^{t/2} \sqrt{1+L^2} |x_{0,k}| \\
\Rightarrow &f(x_t) - f(x_*) = \sum_{k=1}^{n} \lambda_k x_{t,k}^2 \leq L(1+L^2) C'^2 t^2 \gamma^t ||x_0||^2.
\end{aligned}$$

If we set $C_{\mu,L} = L(1+L^2)C'^2$ and recall that

$$\gamma = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\right)^2,$$

we get the announced result:

$$f(x_t) - f(x_*) \leq C_{\mu,L} t^2 \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\right)^{2t} ||x_0 - x_*||^2.$$

□

The theorem we just proved does not extend from strongly convex quadratic functions to general strongly convex functions. Indeed, there are unfavorable strongly convex functions, on which gradient descent with momentum is not faster than its standard version (or even where it diverges whereas plain gradient descent converges). Fortunately, many "interesting" functions are either quadratic or, more frequently, approximately quadratic in the neighborhood of a minimizer. For these functions, heavy ball is usually better than plain gradient descent.

1.2.3 Nesterov's method

In the previous section, we have said that heavy ball has a faster convergence rate than gradient descent for quadratic problems, but not for all strongly convex problems. In addition, it does not apply when the objective function is not strongly convex. In this final section, we present an algorithm which solves both these issues. As it has been found by Yurii Nesterov, it is often called "Nesterov's method".

The iteration formula for this algorithm is

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t + \beta_t(x_t - x_{t-1})) + \beta_t(x_t - x_{t-1}), \tag{1.11}$$

for a proper choice of parameters α_t, β_t. We see that it is very similar to the general form of gradient descent with momentum, as described in Equation (1.10), with the (important) difference that the gradient is not evaluated at point x_t, but at $x_t + \beta_t(x_t - x_{t-1})$.

If f is assumed to be L-smooth and μ-strongly convex, a simple choice is possible for coefficients α_t, β_t:

$$\forall t, \quad \alpha_t = \frac{1}{L} \quad \text{and} \quad \beta_t = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}.$$

This yields the following algorithm.

Input: Starting point x_0, number of iterations T, smoothness parameter L, strong convexity parameter μ.
Set $x_{-1} = x_0, \alpha = \frac{1}{L}, \beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$;
for $t = 0, \ldots, T-1$ **do**
  define

$$x_{t+1} = x_t - \alpha \nabla f\left(x_t + \beta(x_t - x_{t-1})\right) + \beta(x_t - x_{t-1}).$$

end
return x_T
Algorithm 4: Nesterov's algorithm with constant parameters

With this choice, Nesterov's method converges to the minimizer linearly, with decay rate

$$1 - \sqrt{\frac{\mu}{L}},$$

which is similar to the convergence rate of heavy ball, but true for all strongly convex functions, not only quadratic ones!

Theorem 1.2.2 : Nesterov's method: smooth strongly convex case

Let $0 < \mu < L$ be fixed. Let f be an L-smooth and μ-strongly convex function.
Let $(x_t)_{t\in\mathbb{N}}$ be the sequence computed by Algorithm 4. For all $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq 2\left(1 - \sqrt{\frac{\mu}{L}}\right)^t (f(x_0) - f(x_*)).$$

When f is not strongly convex, it is not possible to set parameters α_t and β_t to constant values. A more complicated (and admittedly mysterious, at first sight) definition must be used, described in the following algorithm.

Input: Starting point x_0, number of iterations T, smoothness parameter L.
Set $x_{-1} = x_0, \alpha = \frac{1}{L}, \lambda_{-1} = 0$;
for $t = 0, \ldots, T-1$ **do**
define

$$\lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2};$$
$$\beta_t = \frac{\lambda_{t-1} - 1}{\lambda_t};$$
$$x_{t+1} = x_t - \alpha\nabla f\left(x_t + \beta_t(x_t - x_{t-1})\right) + \beta_t(x_t - x_{t-1}).$$

end
return x_T
Algorithm 5: Nesterov's algorithm with changing parameters

The convergence rate of this algorithm is given in the following theorem.

Theorem 1.2.3 : Nesterov's method: smooth convex case

Let $L > 0$ be fixed. Let f be an L-smooth convex function.
Let $(x_t)_{t\in\mathbb{N}}$ be the sequence computed by Algorithm 5. For all $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq \frac{2L}{(t+1)^2}||x_0 - x_*||^2.$$

Comparing the rates in Theorems 1.1.11 and 1.2.3 shows the superiority of Nesterov's method over gradient descent for smooth convex functions f:

$$\text{gradient descent rate: } O\left(\frac{1}{t}\right);$$
$$\text{Nesterov's method rate: } O\left(\frac{1}{t^2}\right).$$

Actually, it is possible to show that Nesterov's method is *optimal* for smooth convex functions among all first-order algorithms. In other words, for any first-order algorithm (that is, an algorithm which only exploits gradient information about f), there exists an "adversarial" objective function f, which is L-smooth and convex, such that, after t steps,

$$f(x_t) - f(x_*) \geq \frac{3L}{32(t+1)^2}||x_0 - x_*||^2.$$

This means that, up to the constant, no first-order algorithm can achieve a better convergence rate than the one in Theorem 1.2.3.

Nesterov's method is also optimal for smooth strongly convex functions among all first-order algorithms: no first-order algorithm can achieve a better convergence rate, for L-smooth and μ-strongly convex functions, than the one guaranteed by Theorem 1.2.2.

1.3 References

The main references used to prepare these notes are the original article where Polyak introduced the heavy ball algorithm,

- *Some methods of speeding up the convergence of iteration methods*, by B. T. Polyak, Ussr computational mathematics and mathematical physics, volume 4(5), pages 1-17 (1964),

four classical books on optimization,

- *Introduction to optimization*, by B.T. Polyak, Optimization Software (1987),
- *Introductory lectures on convex optimization: a basic course*, by Y. Nesterov, Springer Science & Business Media, volume 87 (2003),

- *Convex optimization*, by S. Boyd and L. Vandenberghe, Cambridge University Press (2004),
- *Optimization for data analysis*, by S. J. Wright and B. Recht, Cambridge University Press (2022).

and two blog posts by S. Bubek on Nesterov's method for smooth convex functions,

- http://blogs.princeton.edu/imabandit/2013/04/01/acceleratedgradientdescent/,
- http://blogs.princeton.edu/imabandit/2018/11/21/a-short-proof-for-nesterovs-momentum/.

Interested readers can read the following research article for more information on the convergence issues of Heavy Ball on non-quadratic functions:

- *Provable non-accelerations of the Heavy-Ball method*, de B. Goujaud, A. Taylor et A. Dieuleveut, arXiv preprint arXiv:2307.11291, 2023.

For another presentation of the advanced aspects of gradient descent, the reader can also refer to

- *Lecture notes on advanced gradient descent*, by C. Royer, https://www.lamsade.dauphine.fr/%7Ecroyer/ensdocs/GD/LectureNotesOML-GD.pdf (2021).

Chapter 2

Non-convex optimization

2.1 Introduction

Let us consider a general unconstrained minimization problem:

$$\text{find } x_* \text{ such that } f(x_*) = \min_{x \in \mathbb{R}^n} f(x),$$

for some $f : \mathbb{R}^n \to \mathbb{R}$. We assume that at least one minimizer, x_*, exists. We also assume througout the lecture that f is $\mathcal{C}^\infty$, to avoid regularity issues.

In the previous chapter, we have discussed how to find a good approximation of a minimizer, under the hypothesis that f is convex. In this chapter, we discuss the situation where f is not convex. The main messages to understand and remember will be as follows.

- Non-convex optimization is fundamentally more difficult than convex optimization.
- It is very rarely possible to certifiably find a global minimizer of a non-convex problem. At best, one can certifiably find a *second-order critical point*.
- Many simple and efficient algorithms can find approximate second-order critical points.
- There are problems for which all second-order critical points are global minimizers (although this is not the general situation). For these problems, the simple and efficient algorithms above therefore find an approximate global minimizer.

2.1.1 Why non-convex optimization is difficult

We first try to give an intuition of why non-convex optimization is much more difficult than convex optimization.

We consider the one-dimensional case, $n = 1$. Let us imagine that we run a first-order algorithm (that is, an algorithm which can access the value of f and ∇f at any desired point, and must return an approximate minimizer based on this information only). After some time, the algorithm has queried the values of f and ∇f at several points, for instance $\left\{-3, -1, -\frac{1}{2}, \frac{3}{2}, 3\right\}$. The gathered information is represented on Figure 2.1.

If f is convex, this already gives significant information on the minimum and minimizer of f. Indeed, the graph of f is above its tangents, and below its chords, which provides upper and lower bounds for f, as shown on Figure 2.2. One can use them to estimate the minimum and minimizer of f. For instance, from the upper and lower bounds of Figure 2.2, one can deduce that

1. the minimum of f is between $-3/8$ and $1/8$;
2. the minimizer(s) of f belong(s) to the interval $[-1/2; 5/6]$.

In particular, from this information, one knows[1] the value of $\min f$ with precision $\frac{1}{4}$ and the minimizer with precision $\frac{2}{3}$.

But if f is not convex, this information does not allow to distinguish, for instance, the two functions plotted in Figure 2.3.

The function represented on the left reaches its minimum at $1/2$, and this minimum is 0. The function on the right reaches its minimum at -2, and this minimum is -1. The difference between the minimums of these two functions is 1, and the difference between the minimizers is 2.5: one cannot produce estimations for the minimal value and minimizer of f with a precision comparable to the convex setting.

Intuitively, to compute a trustworthy approximation of $\min f$ or $\operatorname{argmin} f$ without the convexity assumption, one needs to sample f on a fine grid. As soon as there is a "hole" in the sampling set[2], one cannot know whether the function takes large or small values in this hole, hence one cannot compute a precise estimate of $\min f$ or $\operatorname{argmin} f$. In 1D, it may be possible to sample

[1] The minimum is in the interval $\left[-\frac{3}{8}; \frac{1}{8}\right]$. The middle point of this interval, $-\frac{1}{8}$, is therefore an approximation of $\min f$ which is at most $\frac{1}{4}$ away from the truth.

[2] The *sampling set* is the set of points at which the algorithm queries the values of f and ∇f. In our example, it is $\{-3, -1, -1/2, 3/2, 3\}$.

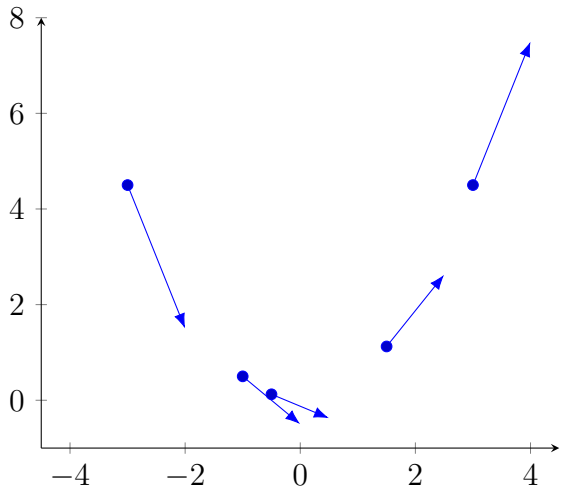


Figure 2.1: Values of f and ∇f at $-3, -1, -\frac{1}{2}, \frac{3}{2}, 3$.

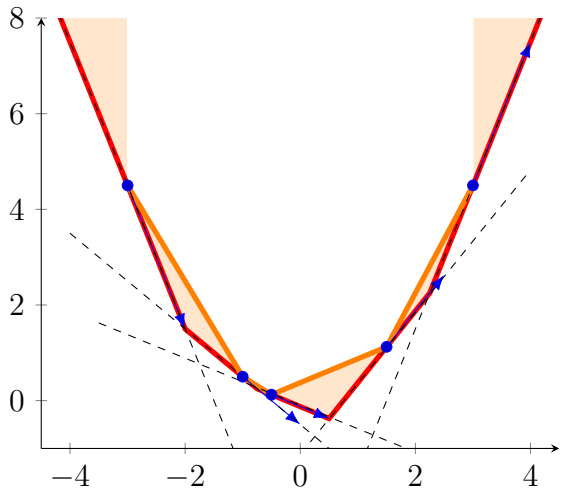


Figure 2.2: Upper and lower bounds on f, deduced from the information on Figure 2.1, under the assumption that f is convex; the graph of f must be entirely contained in the shaded zone.

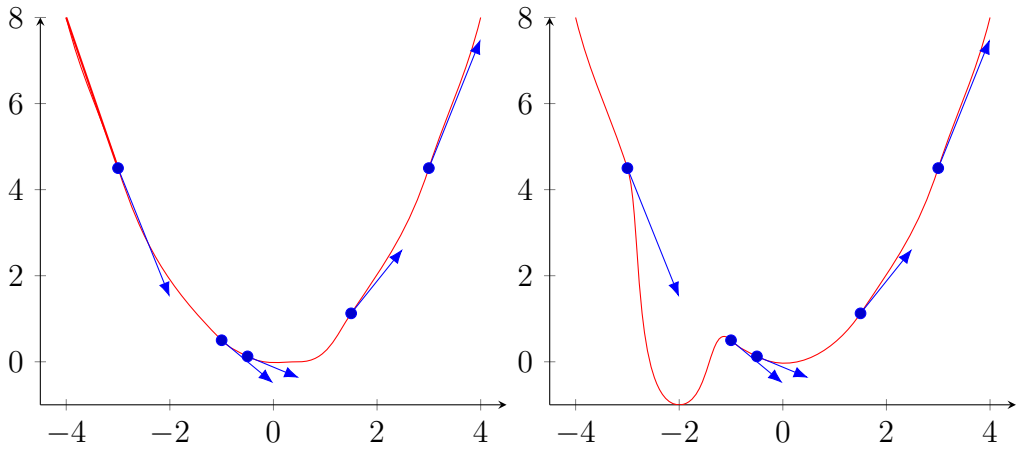


Figure 2.3: Two possible non-convex functions compatible with the information displayed on Figure 2.1.

f on a fine grid, but if n is large, this is out of question: the number of sampling points on a fine grid grows exponentially with the dimension.

As a consequence, if f is not convex, we must give up the idea of finding an approximate minimizer. In the rest of the lecture, we will see which kind of points we can hope to find, and how.

2.2 Critical points

A first idea is to look for a *local minimizer* instead of a global one. It turns out that this is also out of reach, at least for pathological functions.[3] Thus, we lower our expectations again: instead of looking for a local minimizer, we simply look for a point at which "the derivatives of f satisfy the same properties as at a local minimizer".

[3]A class of optimization problems for which finding a local minimizer is NP-hard is for instance presented in the article *On the complexity of finding a local minimizer of a quadratic function over a polytope*, by A A. Ahmadi and J. Zhang, Mathematical Programming, volume 195, pages 783-792 (2022)

Proposition 2.2.1

For any $x \in \mathbb{R}^n$, if x is a local minimizer of f, then

$$\nabla f(x) = 0 \text{ and } \operatorname{Hess} f(x) \succeq 0.$$

Almost conversely, if $\nabla f(x) = 0$ and $\operatorname{Hess} f(x) \succ 0$, then x is a local minimizer of f.

Definition 2.2.2

We say that an element x of $\mathbb{R}^n$ is

- a *first-order critical point* of f if $\nabla f(x) = 0$,
- a *second-order critical point* of f if $\nabla f(x) = 0$ and $\operatorname{Hess} f(x) \succeq 0$.

Example 2.2.3

We consider the map $f : (x_1, x_2) \in \mathbb{R}^2 \to x_1^2 - x_2^2 \in \mathbb{R}$.
Its gradient and Hessian have the following formulas:

$$\forall x = (x_1, x_2) \in \mathbb{R}^2, \quad \nabla f(x) = (2x_1, -2x_2) \quad \text{and} \quad \operatorname{Hess} f(x) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}.$$

Therefore, f has a single first-order critical point, which is $(0, 0)$. This point is not a second-order critical point, because $\begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$ is not positive semidefinite.

In the following two sections, we will see that, contrarily to global or local minimizers, second-order critical points can always be found with arbitrary precision by simple and relatively fast[4] algorithms.

Remark

At this point of the lecture, students often ask me: "Why do we bother discussing how to find second-order critical points? We want a global minimizer; we are not interested in second-order critical points." There

[4] *relatively fast* = whose running time is polynomial in the desired precision, the dimension of the problem and some basic smoothness parameters of the objective function

are two motivations.

- As discussed above, in the non-convex setting, the strongest property that we can prove about an optimization algorithm is essentially that it is able to find a second-order critical point. Therefore, if we want to have some rigorous theoretical criterion to assess the quality of an algorithm or to compare algorithms together, it is a reasonable choice.
- For most functions f encountered in practice, second-order critical points turn out to be local minimizers.[a] In various interesting situations (including the training of neural networks), it has moreover been observed that all second-order critical points of f are approximate global minimizers - or even exactly global minimizers. We will give an example in Section 2.5. In these situations, finding a second-order critical point therefore provides a global minimizer.

[a] Most, but not all functions! The map $(x \to x^3)$, for instance, has a second-order critical point at 0, but no local minimizer.

2.3 Convergence of gradient descent

Let us first consider the simplest first-order algorithm, gradient descent. In the previous lecture, we have seen that it successfully finds a global minimizer in the convex setting. In the non-convex setting, we will see that it always finds a first-order critical point and, "almost always", a second-order one. (The notion of "almost always" will of course be properly defined.)

We assume that f is L-smooth for some $L > 0$: For any $x, y \in \mathbb{R}^n$,

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||.$$

We consider gradient descent with constant stepsize, equal to $1/L$: starting from an arbitrary $x_0 \in \mathbb{R}^n$, we define a sequence $(x_t)_{t\in\mathbb{N}}$ by

$$x_{t+1} = x_t - \frac{1}{L}\nabla f(x_t).$$

2.3.1 Convergence to a first-order critical point

Theorem 2.3.1

Let $T \in \mathbb{N}$ be fixed. We consider the following algorithm:

1. Run T steps of gradient descent, which defines a sequence $(x_0, x_1, \dots, x_T)$.
2. Compute $T_{min} = \text{argmin}_{0 \leq t \leq T} ||\nabla f(x_t)||$ and define $\tilde{x}_T = x_{T_{min}}$.
3. Return $\tilde{x}_T$.

Then

$$||\nabla f(\tilde{x}_T)|| \leq \sqrt{\frac{2L(f(x_0) - f(x_*))}{T}}.$$

We say that $\tilde{x}_T$ is a $O(1/\sqrt{T})$-approximate first-order critical point.

Proof. As seen in the proof of Corollary 1.1.7,

$$\forall t \in \mathbb{N}, \quad f(x_{t+1}) \leq f(x_t) - \frac{1}{2L}||\nabla f(x_t)||^2,$$

which implies

$$\sum_{t=0}^{T-1} ||\nabla f(x_t)||^2 \leq 2L(f(x_0) - f(x_*)).$$

Since $||\nabla f(\tilde{x}_T)|| \leq ||\nabla f(x_t)||$ for any $t \leq T$,

$$T||\nabla f(\tilde{x}_T)||^2 \leq 2L(f(x_0) - f(x_*)),$$

which implies

$$||\nabla f(\tilde{x}_T)|| \leq \sqrt{\frac{2L(f(x_0) - f(x_*))}{T}}.$$

□

2.3.2 Convergence to a second-order critical point

The previous theorem shows that gradient descent always finds approximate first-order critical points. It even provides a convergence rate. For second-order critical points, the picture is more complicated.

For some choices of initial points x_0, it may happen that gradient descent does not get close to an approximate second-order critical point, even when run for an infinite number of steps. For instance, if x_0 is a first-order, but not second-order, critical point of f, then

$$x_0 = x_1 = x_2 = \dots,$$

because $\nabla f(x_0) = 0$, hence gradient descent stays stuck at x_0 and never reaches a second-order critical point.

The following theorem shows that this phenomenon is very rare: for "general" initializations, it does not happen, and gradient descent converges to a second-order critical point.

Theorem 2.3.2 : Lee, Simchowitz, Jordan, Recht (2016)

Let f be an L-smooth function. We assume that

- f has only a finite number of first-order critical points;
- f is coercive (i.e. $f(x) \to +\infty$ when $||x|| \to +\infty$).

We consider gradient descent with constant stepsize $\alpha \in]0; \frac{1}{L}[$.
For almost any x_0,[a] $(x_t)_{t\in\mathbb{N}}$ converges to a second-order critical point.

[a] that is, for all x_0 outside a zero-Lebesgue measure set

Intuition of proof. The finiteness of the critical set and the coercivity of f imply that $(x_t)_{t\in\mathbb{N}}$ converges to a first-order critical point whatever x_0. We admit this fact for simplicity.

We must show that, if x_{crit} is a first-order but not second-order critical point of f, then $(x_t)_{t\in\mathbb{N}}$ does not converge to x_{crit}, for almost any x_0. We consider such a critical point; up to translation, we can assume that it is 0.

We make the (very) simplifying hypothesis that f is quadratic in a ball centered at 0, whose radius we call r_0:

$$\forall x \in B(0, r_0), \quad f(x) = \frac{1}{2}\langle x, Mx\rangle + \langle x, b\rangle,$$

for some $n \times n$ symmetric matrix M.

For any $x \in B(0, r_0)$, $\nabla f(x) = Mx + b$. Since 0 is a first-order critical point, we necessarily have $b = 0$. In addition, $\operatorname{Hess} f(x) = M$ for any $x \in$

$B(0, r_0)$. The assumption that 0 is not a second-order critical point is then equivalent to the fact that $M \not\succeq 0$.

The matrix M can be diagonalized in an orthonormal basis:

$$M = U^T \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{pmatrix} U,$$

with $\lambda_1 \geq \cdots \geq \lambda_n$ the eigenvalues of M and U an orthonormal matrix. Up to a change of coordinates, we can assume $U = \mathrm{Id}$. Since $M \not\succeq 0$, the smallest eigenvalue of M is negative: $\lambda_n < 0$.

We proceed by contradiction, and assume that the sequence $(x_t)_{t\in\mathbb{N}}$ of gradient descent iterates converges to $x_{crit} = 0$. Then x_t belongs to $B(0, r_0)$ for any t large enough, in which case

$$\begin{aligned} x_{t+1} &= x_t - \alpha \nabla f(x_t) \\ &= x_t - \alpha M x_t \\ &= \begin{pmatrix} (1-\alpha\lambda_1)x_{t,1} \\ \vdots \\ (1-\alpha\lambda_n)x_{t,n} \end{pmatrix}. \end{aligned}$$

We fix t_0 such that this relation holds for any $t \geq t_0$. Then, for any $s \in \mathbb{N}$,

$$x_{t_0+s} = \begin{pmatrix} (1-\alpha\lambda_1)^s x_{t_0,1} \\ \vdots \\ (1-\alpha\lambda_n)^s x_{t_0,n} \end{pmatrix}.$$

If the sequence converges to 0, all the coordinates of x_{t_0+s} must go to 0 when s goes to $+\infty$ (for any fixed t), which means that

$$\forall k \in \{1, \dots, n\}, \quad (1 - \alpha\lambda_k)^s x_{t_0,k} \overset{s\to+\infty}{\to} 0. \tag{2.1}$$

We have said that $\lambda_n < 0$, hence $1 < 1 - \alpha\lambda_n$ and $(1 - \alpha\lambda_n)^s \not\to 0$ when $s \to +\infty$. In order for Property (2.1) to hold, we must therefore have

$$x_{t_0,n} = 0.$$

To summarize, we have shown that, if $(x_t)_{t\in\mathbb{N}}$ converges to 0, then, for some t_0,

$$x_{t_0} \in \mathcal{E} \overset{def}{=} \{z \in B(0, r_0) \text{ such that } z_n = 0\}.$$

As a consequence,

$$x_0 \in (\mathrm{Id} - \alpha\nabla f)^{-t_0}(\mathcal{E}).$$

(For any map $g : \mathbb{R}^n \to \mathbb{R}^n$, we define $g^{-t_0}(\mathcal{E})$ as the set of points x such that $g^{t_0}(x) = g \circ \overset{(t_0 \text{ times})}{\dots} \circ g(x) \in \mathcal{E}$.) Therefore, the set of initial points x_0 for which the gradient descent iterates may converge to 0 is included in

$$\bigcup_{t \in \mathbb{N}} (\mathrm{Id} - \alpha \nabla f)^{-t}(\mathcal{E}).$$

The set $\mathcal{E}$ has zero Lebesgue measure and one can check that $\mathrm{Id} - \alpha \nabla f$ is a diffeomorphism, hence $(\mathrm{Id} - \alpha \nabla f)^{-t}(\mathcal{E})$ has zero Lebesgue measure for any $t \in \mathbb{N}$, and the set of "problematic" initial points also has zero Lebesgue measure.

□

2.4 A second-order method

The theorem stated in the previous paragraph only states that gradient descent converges to a second-order critical point (for almost any initial point x_0). It does not say anything about the convergence rate. And it turns out that there are functions f for which convergence is terribly slow.

To overcome this possible slow convergence, several strategies are possible. One of them is to add "noise" to gradient iterates from time to time, to help them get away faster from first-order critical points. The interested reader will find a description in *How to escape saddle points efficiently*, by C. Jin, R. Ge, P. Netrapalli, S. Kakade and M. Jordan (ICML 2017)

Another one is to explicitely exploit the information provided by second-order derivatives. This yields the family of *second-order methods*. In this section, we briefly describe one member of this family: the trust-region method.

Second-order derivatives provide local quadratic approximations of f.

Proposition 2.4.1

For any $x \in \mathbb{R}^n$,

$$f(x+h) = f(x) + \langle h, \nabla f(x) \rangle + \frac{1}{2} \langle h, \mathrm{Hess}\, f(x) h \rangle + o(||h||^2). \quad (2.2)$$

To define x_{t+1} from x_t, it is therefore reasonable to set

$$h_t = \underset{||h|| \leq R_t}{\mathrm{argmin}} \left(f(x_t) + \langle h, \nabla f(x_t) \rangle + \frac{1}{2} \langle h, \mathrm{Hess}\, f(x_t) h \rangle \right)$$

and $x_{t+1} = x_t + h_t$. In the definition of h_t, R_t is a positive number, the *trust radius*. Intuitively, it represents the radius of the region over which the quadratic approximation (2.2) is valid. Choosing it properly is important for the good behavior of the algorithm.

We provide convergence guarantees for this algorithm under the assumption that Hess f is L_2-Lipschitz for some $L_2 > 0$:

$$\forall x, y, h \in \mathbb{R}^n, \quad ||(\text{Hess}\, f(x) - \text{Hess}\, f(y))h|| \leq L_2 ||x - y||\, ||h||.$$

Theorem 2.4.2

Let $\epsilon > 0$ be fixed.
We run the trust-region algorithm as described above, with $R_t = \frac{\sqrt{\epsilon}}{L_2}$ for any t. We stop the algorithm if

$$\frac{||\nabla f(x_t) + \text{Hess}\, f(x_t) h_t||}{||h_t||} \leq \sqrt{\epsilon}$$

and return x_{t+1}.
For any $x_0 \in \mathbb{R}^n$, the algorithm stops after at most $O\left(\frac{L_2^2 (f(x_0) - f(x_*))}{\epsilon^{3/2}}\right)$ iterations and the output x_{final} is an approximate second-order critical point, in the sense that

$$||\nabla f(x_{final})|| \lesssim \frac{\epsilon}{L_2} \quad \text{and} \quad \lambda_{\min}\left(\text{Hess}\, f(x_{final})\right) \gtrsim -\sqrt{\epsilon}.$$

(The notation "$\lesssim$" means "smaller up to a moderate multiplicative constant" and λ_{min} is the smallest eigenvalue.)

2.5 Example: phase retrieval

In the last part of this lecture, we give an example of a non-convex problem where it turns out that all second-order critical points are global minimizers and, moreover, it is possible to rigorously prove this fact. This example is *phase retrieval*.

In phase retrieval, one wants to recover an unknown vector $x_{true} \in \mathbb{C}^n$. Some linear maps $L_1, \ldots, L_m : \mathbb{C}^n \to \mathbb{C}$ are fixed and one has access to

$$y_1 = |L_1(x_{true})|, \ldots, y_m = |L_m(x_{true})|.$$

Here, the double bar, " $|.|$ " denotes the standard complex modulus. This problem is notably motivated by applications in imaging.

Since, for any $\alpha \in \mathbb{R}, k \leq m$, $|L_k(e^{i\alpha}x_{true})| = |e^{i\alpha}|\,|L_k(x_{true})| = |L_k(x_{true})|$, it is not possible to distinguish x_{true} from $e^{i\alpha}x_{true}$, knowing only $y_1, \ldots, y_m$. However, when $m \geq 4n$, it is possible to prove that, for almost all linear forms $L_1, \ldots, L_m$, x_{true} is uniquely determined by $y_1, \ldots, y_m$ up to multiplication by some unitary complex number $e^{i\alpha}$. In this case, which algorithm can recover x_{true}?

Recovering x_{true} is equivalent to finding $x \in \mathbb{C}^n$ such that

$$|L_1(x)| = y_1, \ldots, |L_m(x)| = y_m.$$

The modulus is non-differentiable, but its square is, so it is simpler to rewrite these equalities as

$$|L_1(x)|^2 = y_1^2, \ldots, |L_m(x)|^2 = y_m^2.$$

An intuitive idea to find such an x is to minimize the square-norm error between $(|L_1(x)|^2, \ldots, |L_m(x)|^2)$ and $(y_1^2, \ldots, y_m^2)$, that is

$$\mathcal{L}(x) = \sum_{k=1}^{m} \left(|L_k(x)|^2 - y_k^2\right)^2.$$

The function $\mathcal{L}$ is not convex. Therefore, attempting to minimize it with a first or second-order algorithm may fail: the algorithm will typically find a second-order critical point, but this critical point may not be the global minimizer x_{true}.

Numerically, it can indeed happen that the algorithm returns a point which is not close to x_{true}. However, when m is large enough compared to n and the linear maps $L_1, \ldots, L_m$ are sufficiently "incoherent" with each other, it empirically seems that "bad" critical points do not exist[5].

This fact can be rigorously established, although under strong assumptions on $L_1, \ldots, L_m$. Specifically, we assume that $L_1, \ldots, L_m$ are generated randomly and independently according to a normal distribution (that is, for each k, the coordinates of L_k in the canonical basis are independent realizations of complex Gaussian variables with unit variance). We also assume that

$$m \geq Cn\log^3(n).$$

[5] or, at least, are sufficiently rare so that a first or second-order algorithm does not find them

Theorem 2.5.1 : Sun, Qu, Wright (2018)

Under the above assumptions, the second-order critical points of $\mathcal{L}$ are exactly its global minimizers $\{e^{i\alpha}x_{true}, \alpha \in \mathbb{R}\}$, with probability at least $1 - \frac{1}{m}$.

As a consequence, in this setting, it is possible to recover x_{true} by simply running gradient descent on $\mathcal{L}$, since Theorem 2.3.2 guarantees that gradient descent converges to a second-order critical point for almost any initialization.

2.5.1 Exercise

In the exercise, for simplicity, we consider a *real* (and not *complex*, as presented above) phase retrieval problem:

$$\text{recover } x_* \in \mathbb{R}^n \text{ from } |\langle x_*, v_1\rangle|, \dots, |\langle x_*, v_m\rangle| \ ?$$

Here, $v_1, \dots, v_m$ are known vectors in $\mathbb{R}^n$, "$\langle ., . \rangle$" denotes the usual Euclidean scalar product and "$|.|$" is the absolue value.

Observe that $|\langle x_*, v_k\rangle| = |\langle -x_*, v_k\rangle|$ for any $k = 1, \dots, m$, hence recovery of x_* is at best possible up to sign. This is the real counterpart to the complex notion of "equality up to multiplication by a unitary complex number".

1. We define $y_k = |\langle x_*, v_k\rangle|$ for any $k = 1, \dots, m$ and

$$\begin{array}{rccc} \mathcal{L} & : & \mathbb{R}^n & \to & \mathbb{R} \\ & & x & \to & \sum_{k=1}^m (\langle x, v_k\rangle^2 - y_k^2)^2. \end{array}$$

   Show that a vector $x \in \mathbb{R}^n$ is a global minimizer of $\mathcal{L}$ if and only if

$$|\langle x, v_k\rangle| = |\langle x_*, v_k\rangle|, \quad \forall k = 1, \dots, m.$$

2. Show that $\mathcal{L}$ is C^∞ and that, for all $x, h \in \mathbb{R}^n$,

$$\nabla \mathcal{L}(x) = 4 \sum_{k=1}^m (\langle x, v_k\rangle^2 - y_k^2) \langle x, v_k\rangle v_k,$$

$$\nabla^2 \mathcal{L}(x) \cdot (h, h) = 4 \sum_{k=1}^m (3 \langle x, v_k\rangle^2 - y_k^2) \langle h, v_k\rangle^2 .$$

The goal of the exercise is to give an intuition as to why the second-order critical points of $\mathcal{L}$ are its global minimizers, provided that m is large enough and $v_1, \dots, v_m$ are chosen at random (Theorem 2.5.1). A precise study of $\mathcal{L}$ would be too long, and require tedious computations. Therefore, we will focus on the much simpler object $\mathbb{E}\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$, where "$\mathbb{E}$" is the expectation with respect to $v_1, \dots, v_m$.

3. From now on, we assume that $v_1, \dots, v_m$ are chosen at random according to independent normal distributions (that is, each coordinate of each v_k is independently chosen according to the law $\mathcal{N}(0, 1)$).

   Show that, for any $x, h \in \mathbb{R}^n$,

$$\begin{aligned}\mathbb{E}(\nabla\mathcal{L}(x)) &= 4m\left(\left(3||x||^2 - ||x_*||^2\right)x - 2\langle x_*, x\rangle x_*\right),\\ \mathbb{E}(\nabla^2\mathcal{L}(x)\cdot(h,h)) &= 4m\left(6\langle x, h\rangle^2 - 2\langle x_*, h\rangle^2\right.\\ &\qquad \left. +(3||x||^2 - ||x_*||^2)||h||^2\right).\end{aligned}$$

   [Hint: you can admit that, for arbitrary $a, b \in \mathbb{R}^n$ and any k,

$$\begin{aligned}\mathbb{E}(\langle a, v_k\rangle^2 \langle b, v_k\rangle v_k) &= 2\langle a, b\rangle a + ||a||^2 b,\\ \mathbb{E}(\langle a, v_k\rangle^2 \langle b, v_k\rangle^2) &= 2\langle a, b\rangle^2 + ||a||^2||b||^2.\end{aligned}$$

   Do not treat $y_1, \dots, y_m$ as constants: they depend on $v_1, \dots, v_m$.]

4. Assuming, for simplicity, $x_* \neq 0$, compute the first and second-order critical points of $\mathbb{E}\mathcal{L}$.

   [Remark : for any $x, h \in \mathbb{R}^n$, it holds $\nabla(\mathbb{E}\mathcal{L})(x) = \mathbb{E}(\nabla\mathcal{L}(x))$ and $\nabla^2(\mathbb{E}\mathcal{L})(x)\cdot(h,h) = \mathbb{E}(\nabla^2\mathcal{L}(x)\cdot(h,h))$.]

Solution

1. For any $x \in \mathbb{R}^n$, $\mathcal{L}(x) \geq 0$ (since it is a sum of squares). In addition,

$$\mathcal{L}(x_*) = \sum_{k=1}^{m}\left(\langle x_*, v_k\rangle^2 - |\langle x_*, v_k\rangle|^2\right)^2 = 0.$$

Consequently, $\min \mathcal{L} = 0$ and, for any $x \in \mathbb{R}^n$,

$$\begin{aligned}
& x \text{ is a global minimizer of } \mathcal{L} \\
\iff & \mathcal{L}(x) = 0 \\
\iff & \sum_{k=1}^{m} (\langle x, v_k \rangle^2 - y_k^2)^2 = 0 \\
\iff & \langle x, v_k \rangle^2 - y_k^2 = 0, \quad \forall k = 1, \dots, m \\
\iff & \langle x, v_k \rangle = \pm |\langle x_*, v_k \rangle|, \quad \forall k = 1, \dots, m \\
\iff & |\langle x, v_k \rangle| = |\langle x_*, v_k \rangle|, \quad \forall k = 1, \dots, m.
\end{aligned}$$

2. The map $\mathcal{L}$ is polynomial in the coordinates of x. It is thus C^∞.

Let us compute its derivatives. For any $x \in \mathbb{R}^n$, the gradient $\nabla \mathcal{L}(x)$ is the only vector such that

$$\mathcal{L}(x + w) = \mathcal{L}(x) + \langle \nabla \mathcal{L}(x), w \rangle + o(||w||).$$

And, for any x, w,

$$\begin{aligned}
\mathcal{L}(x + w) &= \sum_{k=1}^{m} \left(\langle x + w, v_k \rangle^2 - y_k^2 \right)^2 \\
&= \sum_{k=1}^{m} \left(\langle x, v_k \rangle^2 - y_k^2 + 2 \langle x, v_k \rangle \langle w, v_k \rangle + o(||w||) \right)^2 \\
&= \sum_{k=1}^{m} \Big[\left(\langle x, v_k \rangle^2 - y_k^2 \right)^2 \\
&\qquad\qquad + 4 \left(\langle x, v_k \rangle^2 - y_k^2 \right) \langle x, v_k \rangle \langle w, v_k \rangle + o(||w||) \Big] \\
&= \mathcal{L}(x) + 4 \sum_{k=1}^{m} \left(\langle x, v_k \rangle^2 - y_k^2 \right) \langle x, v_k \rangle \langle w, v_k \rangle + o(||w||) \\
&= \mathcal{L}(x) + \left\langle 4 \sum_{k=1}^{m} \left(\langle x, v_k \rangle^2 - y_k^2 \right) \langle x, v_k \rangle v_k, w \right\rangle + o(||w||).
\end{aligned}$$

We thus have

$$\nabla \mathcal{L}(x) = 4 \sum_{k=1}^{m} \left(\langle x, v_k \rangle^2 - y_k^2 \right) \langle x, v_k \rangle v_k.$$

As to the Hessian at a point $x \in \mathbb{R}^n$, it is the only quadratic function such that, for any $l \in \mathbb{R}^n$,

$$\langle \nabla \mathcal{L}(x+h), l \rangle = \langle \nabla \mathcal{L}(x), l \rangle + \nabla^2 \mathcal{L}(x) \cdot (h, l) + o(||h||).$$

For any x, h, l,

$$\begin{aligned}
\langle \nabla \mathcal{L}(x+h), l \rangle &= 4 \sum_{k=1}^{m} \left(\langle x+h, v_k \rangle^2 - y_k^2 \right) \langle x+h, v_k \rangle \langle v_k, l \rangle \\
&= 4 \sum_{k=1}^{m} \left(\langle x, v_k \rangle^2 - y_k^2 + 2 \langle x, v_k \rangle \langle h, v_k \rangle + o(||h||) \right) \\
&\qquad \times \left(\langle x, v_k \rangle + \langle h, v_k \rangle \right) \langle v_k, l \rangle \\
&= 4 \sum_{k=1}^{m} \big[(\langle x, v_k \rangle^2 - y_k^2) \langle x, v_k \rangle \langle v_k, l \rangle \\
&\qquad + (3 \langle x, v_k \rangle^2 - y_k^2) \langle h, v_k \rangle \langle v_k, l \rangle \big] + o(||h||) \\
&= \langle \nabla \mathcal{L}(x), l \rangle + 4 \sum_{k=1}^{m} (3 \langle x, v_k \rangle^2 - y_k^2) \langle v_k, h \rangle \langle v_k, l \rangle + o(||h||).
\end{aligned}$$

Consequently

$$\nabla^2 \mathcal{L}(x) \cdot (h, l) = 4 \sum_{k=1}^{m} (3 \langle x, v_k \rangle^2 - y_k^2) \langle h, v_k \rangle \langle l, v_k \rangle ,$$

which implies that, for any x, h,

$$\nabla^2 \mathcal{L}(x) \cdot (h, h) = 4 \sum_{k=1}^{m} (3 \langle x, v_k \rangle^2 - y_k^2) \langle h, v_k \rangle^2 .$$

Another possibility to solve the question would have been to compute the partial derivatives. Indeed, we know that, for any $x, h \in \mathbb{R}^n$,

$$\nabla \mathcal{L}(x) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial x_1}(x) \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial x_n}(x) \end{pmatrix} \quad \text{and} \quad \nabla^2 f(x) \cdot (h, h) = \sum_{i,j} \frac{\partial^2 \mathcal{L}}{\partial x_i \partial x_j}(x) h_i h_j.$$

3. For any $x \in \mathbb{R}^n$,

$$\begin{aligned}\mathbb{E}\left(\nabla\mathcal{L}(x)\right) &= 4\sum_{k=1}^{m}\mathbb{E}\left(\langle x, v_k\rangle^3 v_k\right) - 4\sum_{k=1}^{m}\mathbb{E}\left(\langle x_*, v_k\rangle^2 \langle x, v_k\rangle v_k\right)\\ &= 4\sum_{k=1}^{m}(3||x||^2 x) - 4\sum_{k=1}^{m}(2\langle x_*, x\rangle x_* + ||x_*||^2 x)\\ &= 4m\left(\left(3||x||^2 - ||x_*||^2\right)x - 2\langle x_*, x\rangle x_*\right).\end{aligned}$$

For any $x, h \in \mathbb{R}^n$,

$$\begin{aligned}\mathbb{E}\left(\nabla^2\mathcal{L}(x)\cdot(h,h)\right) &= 12\sum_{k=1}^{m}\mathbb{E}\left(\langle x, v_k\rangle^2\langle h, v_k\rangle^2\right) - 4\sum_{k=1}^{m}\mathbb{E}\left(\langle x_*, v_k\rangle^2\langle h, v_k\rangle^2\right)\\ &= 12\sum_{k=1}^{m}(2\langle x, h\rangle^2 + ||x||^2||h||^2)\\ &\qquad - 4\sum_{k=1}^{m}(2\langle x_*, h\rangle^2 + ||x_*||^2||h||^2)\\ &= 4m\left(6\langle x, h\rangle^2 - 2\langle x_*, h\rangle^2 + (3||x||^2 - ||x_*||^2)||h||^2\right).\end{aligned}$$

4. We start with the first-order critical points. For any $x \in \mathbb{R}^n$, $\nabla(\mathbb{E}\mathcal{L})(x) = 0$ if and only if

$$4m\left(\left(3||x||^2 - ||x_*||^2\right)x - 2\langle x_*, x\rangle x_*\right) = 0.$$

This happens if and only if

$$3||x||^2 - ||x_*||^2 = \langle x_*, x\rangle = 0 \tag{2.3}$$

or

$$3||x||^2 - ||x_*||^2 \neq 0 \quad \text{and} \quad x = \frac{2\langle x_*, x\rangle}{3||x||^2 - ||x_*||^2}x_*. \tag{2.4}$$

The set of vectors x satisfying Equation (2.3) is

$$\left\{\frac{||x_*||}{\sqrt{3}}u \,\middle|\, u \in \{x_*\}^\perp, ||u|| = 1\right\}.$$

Additionally, a vector x satisfies Equation (2.4) if and only if it is colinear to x_* (that is, $x = \lambda x_*$ for some $\lambda \in \mathbb{R}$) and the colinearity factor λ is such that

$$0 \neq 3||\lambda x_*||^2 - ||x_*||^2 = (3\lambda^2 - 1)||x_*||^2$$

and

$$\begin{aligned}\lambda x_* &= x \\ &= \frac{2\langle x_*, \lambda x_*\rangle}{3||\lambda x_*||^2 - ||x_*||^2} \\ &= \frac{2\lambda}{3\lambda^2 - 1} x_*.\end{aligned}$$

These two equations are equivalent to the following conditions:

1. $3\lambda^2 - 1 \neq 0$;
2. $\lambda = \frac{2\lambda}{3\lambda^2-1}$, that is $\lambda = 0$ or $1 = \frac{2}{3\lambda^2-1}$, that is $\lambda \in \{-1, 0, 1\}$.

Consequently, the set of vectors x which satisfy Equation (2.4) is

$$\{-x_*, 0, x_*\}.$$

We have therefore shown that the set of first-order critical points of $\mathbb{E}\mathcal{L}$ is

$$\left\{ \frac{||x_*||}{\sqrt{3}} u \,\middle|\, u \in \{x_*\}^\perp, ||u|| = 1 \right\} \cup \{-x_*, 0, x_*\}.$$

A second-order critical point of $\mathbb{E}\mathcal{L}$ is a point x such that

1. x is first-order critical;
2. $\nabla^2 \mathbb{E}\mathcal{L}(x) \succeq 0$.

Let us consider a first-order critical point x, and determine whether $\nabla^2 \mathbb{E}\mathcal{L}(x) \succeq 0$.

- First case: $x = \frac{||x_*||}{\sqrt{3}} u$ for some unit-normed vector u orthogonal to x_*. For any h,

$$\begin{aligned}&\nabla^2 \mathbb{E}\mathcal{L}(x) \cdot (h, h) \\ =&4m \left(6 \left\langle \frac{||x_*||}{\sqrt{3}} u, h \right\rangle^2 - 2\langle x_*, h\rangle^2 + \left(3 \left|\left| \frac{||x_*||}{\sqrt{3}} u \right|\right|^2 - ||x_*||^2 \right) ||h||^2 \right) \\ =&4m \left(2||x_*||^2 \langle u, h\rangle^2 - 2\langle x_*, h\rangle^2 \right).\end{aligned}$$

  In particular,

$$\nabla^2 \mathbb{E}\mathcal{L}(x) \cdot (x_*, x_*) = -8m||x_*||^2 < 0.$$

  Therefore, $\nabla^2 \mathbb{E}\mathcal{L}(x) \not\succeq 0$.

- Second case: $x = 0$.

  For any h,

  $$\nabla^2 \mathbb{E}\mathcal{L}(x) \cdot (h, h) = -4m(2\langle x_*, h\rangle + ||x_*||^2||h||^2).$$

  In particular,

  $$\nabla^2 \mathbb{E}\mathcal{L}(x) \cdot (x_*, x_*) = -12||x_*||^2 < 0.$$

  Therefore, $\nabla^2 \mathbb{E}\mathcal{L}(x) \not\succeq 0$.

- Third case: $x = \pm x_*$.

  For any h,

  $$\nabla^2 \mathbb{E}\mathcal{L}(x) \cdot (h, h) = 8m\left(2\langle x_*, h\rangle^2 + ||x_*||^2||h||^2\right).$$

  This is a sum of squares, hence always nonnegative: $\nabla^2 \mathbb{E}\mathcal{L}(x) \succeq 0$.

The only second-order critical points are $-x_*$ and x_*.

2.6 References

- *Gradient descent only converges to minimizers*, by J. D. Lee, M. Simchowitz, M. Jordan and B. Recht, in the Conference on Learning Theory (COLT) (2016).

- *Second-order optimization algorithms I*, lecture notes by Y. Ye, available at http://web.stanford.edu/class/msande311/2017lecture13.pdf.

- *Computing a trust region step*, by J. J. Moré and D. C. Sorensen, in the SIAM journal on scientific and statistical computing, volume 4(3) (1983).

- *A geometric analysis of phase retrieval*, by J. Sun, Q. Qu and J. Wright, Foundations of Computational Mathematics, volume 18(5) (2018).