Optimization for large scale machine learning
ENS, M2 Info

Mathurin Massias
mathurin.massias@gmail.com

Goals of the class

In this class we will study the resolution of optimization problems arising in Machine Learning. These problems are usually characterized by their large scale, which call for dedicated classes of algorithms. They also usually present some specific structure, that can be leveraged to obtain moderate precision solutions.

The main objectives are to:

1. develop a taxonomy of optimization problems and which algorithms can be used in which settings
2. understand convergence properties and functions properties that govern them
3. master theoretical tools and techniques to derive convergence proofs
4. get hands on practice to challenge theoretical results, in Python

I thank Cesare Molinari, Saverio Salzo and Silvia Villa for their graduate course on convex analysis.

General notational conventions

We almost always work in finite dimension and the optimization is performed over the set $\mathbb{R}^d$. In a Machine Learning context, the integer d will thus denote the dimension of the parameter space, while n will denote the number of samples (observed data points). We adopt the optimization convention, i.e. we write a linear system as $Ax = b$ (while statistics has $y = X\beta$, the inverse problems field writes $f = Au$, signal processing writes $y = \Phi x$...): the unknown is x; the iterates are x_k or x_t. As much as possible we write x^* for the minimizer of given problems, but due to the various meaning of star (Fenchel conjugation, adjoint), we also write $\hat{x}$ as in the statistical literature.

These conventions are meant to ease reading, but there can always be exceptions. It is anyways a good exercise to adapt to different notation.

1 Motivation: gradient descent on ordinary least squares

1.1 Least squares and other optimization problems in ML

Most optimization problems studied in this class are of the following form:

$$\inf_{x \in \mathcal{C}} f(x) \ , \tag{1}$$

where $\mathcal{C}$ is a subset of $\mathbb{R}^d$ and f is a real-valued function on a subset of $\mathbb{R}^d$.

Formally, solving Problem (1) means to find the largest lower bound on all values $f(x)$ when x describes $\mathcal{C}$. This infimum may not exist, and if it exists it may not necessarily be attained, even for nicely-behaved f. The reader should mind that most of the time, we write $\min_{\mathcal{C}} f$ instead of $\inf_{\mathcal{C}} f$; this is an abuse of notation and does not mean that the infimum is attained (not even that it is finite).

In Machine Learning, rather that the smallest value taken by f, we are more interested in finding a minimizer, that is

$$x^* \in \operatorname*{argmin}_{x \in \mathcal{C}} f(x) \ , \tag{2}$$

meaning in finding $x^* \in \mathcal{C}$ such that for all $x \in \mathcal{C}$, $f(x^*) \leq f(x)$. In Machine Learning, x^* usually represent the parameters of a model, that once computed are used to achieve some statistical task.

Example 1.1 (Ordinary least squares)**.** *Let $n \in \mathbb{N}$, let $a_1, \dots, a_n \in \mathbb{R}^d$, let $b_1, \dots, b_n \in \mathbb{R}$. In a statistical setting, suppose that there is an approximate*[1] *linear relationship between the a_i's and the b_i's:*

$$b_i \approx a_i^\top x \ . \tag{3}$$

One may try to infer x from the observation of the pairs (a_i, b_i). From the optimization point-of-view, this task can be formulated as follows: amongst all linear functions $a \mapsto a^\top x$, find the one that "best fits" (in the squared loss sense; other senses are possible) the n pairs (a_i, b_i). The latter is given by

$$x^* \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \sum_1^n (b_i - a_i^\top x)^2 = \operatorname{argmin} \frac{1}{2} \|Ax - b\|^2 \ , \tag{4}$$

*where the **design matrix** $A \in \mathbb{R}^{n \times d}$ has i-th row a_i, and the **target vector** $b \in \mathbb{R}^n$ has i-th entry b_i. Problem (4) is called Ordinary Least Squares.*

We said we are looking for "the" linear function that best fits, but it may not be unique[2]*, hence we have written $x^* \in \operatorname{argmin}$ and not $x^* = \operatorname{argmin}$.*

This is also an illustration that the value of the infimum is not always very interesting (if $d \geq n$ and A has rank n it is 0), contrary to the minimizer.

[1] in a voluntarily vague sense, one possible meaning being (5)

[2] When is it not unique? And does there always exist one?

Least Squares form a statistical perspective Suppose we have two random variables A and B linked by

$$B \sim \mathcal{N}(A^\top x, \sigma^2) \ . \tag{5}$$

Consider a dataset of n independent observations, meaning realizations $(a_i, b_i)_{i\in[n]}$ of (A, B). The conditional log-likelihood writes:

$$\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|a_i^\top x - b_i\|^2}{2\sigma^2}\right) \ . \tag{6}$$

Minimization of the negative log likelihood in x gives Problem (4): the least squares solution is the maximum likelihood estimator for the linear model with Gaussian noise. Solving Problem (4) gives x and allows predicting a value b_{n+1} if one observes a new entry point a_{n+1}.

It is perfectly possible to postulate a different generative model than (5), or simply not impose a generative model, and find x by minimizing another cost/loss function. This is the setting of empirical risk minimization (ERM): one finds the parameters of a model by minimizing a cost function, often –but not always– arising from likelihood maximization.

Example 1.2 (ℓ_1 and Huber regression)**.** *Ordinary least squares are strongly influenced by outliers: the squared loss gives a lot of importance to large differences: in the objective function (4) a mismatch of* 2 *between the true observation* b_i *and the model fit* $a_i^\top x$ *costs* $2^2/2 = 2$*, but a mismatch of* 10 *costs* $10^2/2 = 50$*. A more* robust *solution, that is less dominated by large mismatches (or "outliers"), can be found by minimizing the sum of absolute errors:*

$$\operatorname*{argmin}_{x\in\mathbb{R}^d} \sum_{i=1}^{n} |a_i^\top x - b_i| = \operatorname*{argmin}_{x\in\mathbb{R}^d} \|Ax - b\|_1 \ . \tag{7}$$

Problem (7) is unfortunately more complicated to solve, as we shall see. It can be reformulated as a Linear Program, but the cost to solve it scales very badly with d*. A hybrid of Problems (4) and (7) involves the Huber loss (Figure 1):*

$$\operatorname*{argmin}_{x\in\mathbb{R}^d} \sum_{1}^{n} \mathrm{huber}_\rho(a_i^\top x - b_i) \ , \tag{8}$$

with the Huber loss defined as:

$$\mathrm{huber}_\rho(x) = \begin{cases} \frac{x^2}{2\rho} + \frac{\rho}{2} \ , & \text{if } |x| \le \rho \ , \\ |x| \ , & \text{otherwise.} \end{cases} \tag{9}$$

The Huber loss is less sensitive to outliers than the squared ℓ_2 *loss, and more optimization friendly than the* ℓ_1 *loss, but there is no free lunch: it relies on a parameter* $\rho \in \mathbb{R}_+$ *that needs to be tuned, and there is no obvious and cheap way to chose the best* ρ*.*

Finally, instead of a linear model, one may model the dependency between observations and target by a function belonging to some parametric class $\{f_x : \mathbb{R}^d \mapsto \mathbb{R} : x \in \mathbb{R}^p\}$, described by p parameters stored in x:

$$b_i \approx f_x(a_i) \ , \tag{10}$$

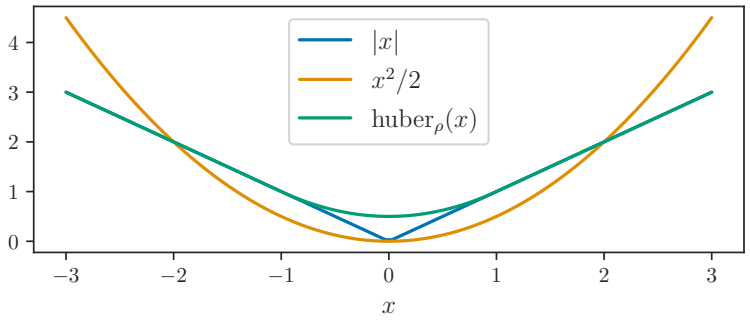

Figure 1: ℓ_1, squared ℓ_2 and Huber ($\rho = 1$) losses

and, depending on what is meant by $\approx$, minimize a loss $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, yielding the following Empirical Risk Minimization (ERM) problem:

$$x^* \in \operatorname*{argmin}_{x \in \mathbb{R}^p} \sum_{i=1}^{n} L(f_x(a_i), b_i) \ . \tag{11}$$

As we have seen, the typical example is $f_x = x^\top \cdot$; but this framework also fits the Deep Learning framework (with f_x a composition of matrix vector multiplications followed by pointwise application of a non-linear function.).

1.2 Solving least squares with gradient descent

Problem (4) can be solved in closed-form with dedicated algorithms such as Cholesky, LU or QR decompositions (see Exercise 1.7); but ℓ_1, Huber and generic ERM models cannot. We need general purpose algorithms to solve classes of similar problems.

One of the arguably simplest minimization algorithm is gradient descent (GD). If it exists, the gradient of a function points in the direction of maximal increase, since the directional derivative $f'(x; v) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon v) - f(x)}{\epsilon}$ is given by

$$f'(x; v) = \langle \nabla f(x), v \rangle \ . \tag{12}$$

To decrease the value of f as fast as possible, it makes sense to move in the opposite direction of the gradient. Taking a stepsize $\eta > 0$, this gives the gradient descent iterations:

$$x_{t+1} = x_t - \eta \nabla f(x_t) \ . \tag{13}$$

Proposition 1.3 (Convergence rate of GD on OLS)**.** *Consider the OLS problem, and assume that there exist strictly positive scalars μ and L such that $\mu \operatorname{Id} \preceq A^\top A \preceq L \operatorname{Id}$. Let $\kappa = L/\mu$ denote the* condition number*. Their exists a unique minimizer of least squares, x^*, and the iterates of gradient descent on ordinary least squares with stepsize $\eta = 1/L$ satisfy:*

$$\|x_t - x^*\| \leq \exp(-t/\kappa) \|x_0 - x^*\| \ . \tag{14}$$

Proof. The gradient of $x \mapsto \frac{1}{2}\|Ax - b\|^2$ is $A^\top(Ax - b)$, hence gradient descent on ordinary least squares read

$$x_{t+1} = x_t - \eta A^\top (A x_t - b) \ . \tag{15}$$

There exists a unique minimizer x^*, and it satisfies $A^\top A x^* = A^\top b$ (Exercises 1.6 and 1.7), so:

$$x_{t+1} - x^* = (\mathrm{Id} - \eta A^\top A)(x_t - x^*) \ . \tag{16}$$

Observe that

$$1 - \eta L \preccurlyeq \mathrm{Id} - \eta A^\top A \preccurlyeq 1 - \eta\mu \ , \tag{17}$$

hence

$$\| \mathrm{Id} - \eta A^\top A\|_2 \leq q(\eta) \triangleq \max(|1 - \eta L|, |1 - \eta\mu|) \ . \tag{18}$$

Setting $\eta = 1/L$ yields $q(\eta) = 1 - \mu/L$, and using $(1-u)^t \leq \exp(-tu)$ concludes. □

Remark 1.4 (Other choices of η.)**.** *We have actually proven the stronger result:*

$$\|x_t - x^*\| \leq q(\eta)^t \|x_0 - x^*\| \ . \tag{19}$$

This shows that any choice of η in $]0, 2/L[$ leads to convergence, because $q(\eta) < 1$.

We can also tune η to minimize $q(\eta)$: picking $\eta = \frac{2}{L+\mu}$ yields $q(\eta) = \frac{L-\mu}{L+\mu} = \frac{\kappa-1}{\kappa+1}$. Suppose that we want to get ε-close to x^*, then we need at most $\kappa \log(\|x_0 - x^*\|/\varepsilon)$ iterations of GD. That is a very good rate, so-called linear*: the number of iterations grows very slowly as the required precision goes to 0.*

Remark 1.5. *We can also obtain a rate in objective value:*

$$f(x_t) - f(x^*) = \langle \nabla f(x^*), x_t - x^* \rangle + \frac{1}{2}(x_t - x^*)A^\top A(x_t - x^*) \tag{20}$$

$$= \frac{1}{2}(x_t - x^*)A^\top A(x_t - x^*) \tag{21}$$

$$\leq \frac{L}{2}\|x_t - x^*\|^2 \ , \tag{22}$$

but as we've seen, such rates are usually less desirable than rates on x_t.

Questions: Does there always exist an L as in the assumptions of Proposition 1.3? And a μ? To what do they correspond geometrically?

1.3 Numerical puzzles

On Figure 2 is the numerically observed convergence rate of GD on one instance of OLS. The matrix A is 500 by 500 with i.i.d. normal entries so it is full rank, $\mu > 0$ and Proposition 1.3 applies. Yet the rate we see is clearly not linear. Why?

On Figure 3, A is still random but its shape is 200×300, so $A^\top A$ is not full rank, $\mu = 0$. Yet we see a clear linear convergence rate (numerical floating point errors kick in at iteration 1000). Why?

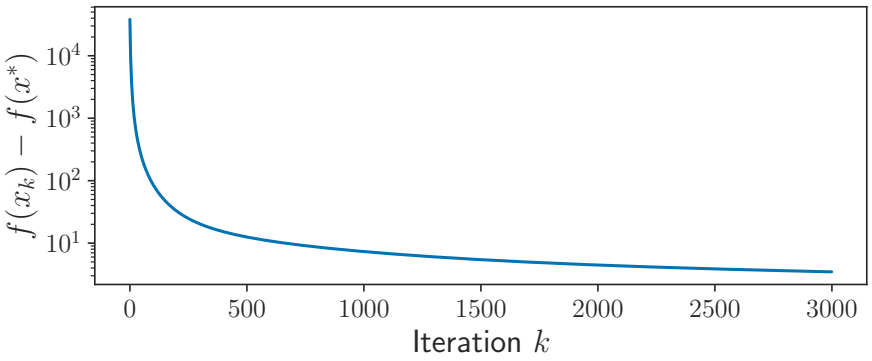

Figure 2: Objective convergence rate of GD on OLS on a random i.i.d. $A \in \mathbb{R}^{500\times500}$.

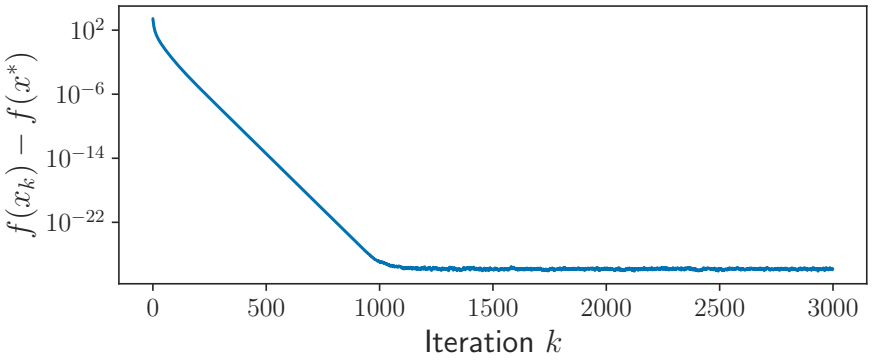

Figure 3: Objective convergence rate of GD on OLS on a random i.i.d. $A \in \mathbb{R}^{200\times300}$.

1.4 Exercises

Exercise 1.6. ☕ *Let $A \in \mathbb{R}^{n\times d}$. Show that* $\operatorname{Ker} A = \operatorname{Ker} A^*A$.
Show that for any $b \in \mathbb{R}^n$ there exist a solution to $A^*Ax = A^*b$.
☕☕ *Show that there does not always exist a solution to $A^*Ax = A^*b$ in the infinite dimensional case (when A is a bounded linear operator between infinite dimensional Hilbert spaces.)*

Exercise 1.7. ☕ *Let $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$. Show that solving Ordinary Least Squares:*

$$\min \frac{1}{2}\|Ax - b\|^2 \ , \tag{23}$$

amounts to solving $A^*Ax = A^*b$ (aka the normal equations*).*
Show that the set of solutions is:

$$A^\dagger b + \operatorname{Ker} A \ .$$

Exercise 1.8. ☕ *When is $x \mapsto \|Ax - b\|^2$ strictly convex? Strongly convex?*

Exercise 1.9 (Least squares with intercept)**.** ☕☕ *An* intercept *x_0 is a constant scalar term in the linear prediction function, that becomes $a \mapsto a^\top x + x_0$. Fitting an intercept can be done by adding a column of 1s to A. Alternatively, show that the solution of least squares with intercept,*

$$(\hat{x}, \hat{x}_0) \in \underset{x\in\mathbb{R}^d, x_0\in\mathbb{R}}{\operatorname{argmin}} \frac{1}{2}\|Ax - b - x_0\mathbf{1}\|^2 \tag{24}$$

is given by:

$$\hat{x} = \hat{x}_c \ , \tag{25}$$

$$\hat{x}_0 = \frac{1}{n}\sum_1^n (a_i^\top \hat{x} - b_i) \ , \tag{26}$$

where $\hat{x}_c$ is the solution of least squares without intercept on centered data A_c and b_c (versions of A and b where the rowwise mean has been subtracted).

Exercise 1.10 (Gradient descent on isotropic parabola)**.** ☕ *Let $A \in \mathbb{R}^{n\times d}$ be such that the condition number of $A^\top A$[3] is equal to 1. Show that gradient descent with stepsize $1/L$ converges in a single iteration for the problem* $\min \frac{1}{2}\|Ax - b\|^2$.

[3] i.e. the ratio between the largest and the smallest eigenvalues of $A^\top A$.

2 Reminder on convex analysis

Section 1 has shown us basic notions in optimization. We have studied gradient descent, an iterative algorithm producing a sequence of iterates x_k that aim at solving an optimization problem. We have proved two types of convergence results[4]:

- in iterates: $\|x_k - x^*\| \to 0$
- in function values: $f(x_k) - \inf f \to 0$.

Both result were *non-asymptotic*: we had information about the speed at which convergence happened.

We also saw that the minimizer could be non unique. In the sequel we want to minimize functions beyond least squares. Under which conditions does there exist a minimizer? Under which is it unique?

2.1 Characterization, uniqueness and existence of minimizers

Definition 2.1 (Global and local minimizers)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$. A global minimizer of f is a point x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^d$. A local minimizer of f is a point x^* such that there exists a neighborhood V of x^* such that $f(x^*) \leq f(x)$ for all $x \in V$.*

In optimization, it is often convenient to work with functions that take values in $\overline{\mathbb{R}} \triangleq \mathbb{R} \cup \{+\infty\}$. The prototypical example is the *indicator* function of a set.

Definition 2.2 (Indicator function)**.** *In convex analysis, the indicator function $\iota_{\mathcal{C}}$ of a subset $\mathcal{C}$ of $\mathbb{R}^d$ is:*

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0 \ , & \text{if } x \in \mathcal{C} \ , \\ +\infty \ , & \text{otherwise} \ . \end{cases} \tag{27}$$

The indicator function conveniently allows transforming all constrained minimization problems into unconstrained ones with extended value functions:

$$\operatorname*{argmin}_{x \in \mathcal{C}} f(x) = \operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) + \iota_{\mathcal{C}}(x) \ . \tag{28}$$

Definition 2.3. *The* domain *of a function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ is:*

$$\operatorname{dom} f \triangleq \{x \in \mathbb{R}^d, f(x) < +\infty\} \ . \tag{29}$$

Definition 2.4. *A function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ is convex iff its domain is convex and it lies below its cords:*

$$\forall x, y \in \operatorname{dom} f, \forall \lambda \in [0,1], \quad f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1 - \lambda f(y)) \ . \tag{30}$$

It is strictly convex if the inequality (30) holds strictly.

[4] Does one imply the other? Under which condition?

Convex functions are amenable to optimization because of the nice "local to global" properties they enjoy.

Proposition 2.5 (Local min are global min)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function. Then all local minimizers of f are also global.*

Proposition 2.6 (Global lower bounds)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex differentiable function. Then*

$$\forall x, y \in \mathbb{R}^d, \quad f(x) - f(y) \leq \langle \nabla f(x), x - y \rangle \ .$$

Written as $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$, it tells that f lies above all of its tangents. From local information $\nabla f(x)$, we get an information about the whole behavior of f.

Proposition 2.7 (Characterization on constrained minimizers)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function, let $\mathcal{C}$ be a non-empty closed and convex subset of $\mathbb{R}^d$. Then x^* is a minimizer of f restricted to $\mathcal{C}$ if and only if:*

$$\forall x \in \mathcal{C}, \quad \langle \nabla f(x^*), x^* - x \rangle \leq 0 \ . \tag{31}$$

aka the gradient of f points inwards $\mathcal{C}$ at x^*. If the problem is unconstrained ($\mathcal{C} = \mathbb{R}^d$), this implies $\nabla f(x^*) = 0$.

Finally, the following proposition provides conditions under which a function admits a minimizer.

Proposition 2.8 (Direct method of calculus of variations)**.** *Let $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ be a lower semicontinuous function that has one nonempty and bounded sublevel set. Then* $\operatorname{argmin} f$ *is not empty.*

Equipped with this theoretical framework, we can move to our first convergence proofs on generic functions.

2.2 Exercises

2.2.1 Convexity

Exercise 2.9. ☕ *Show that local minimizers of convex functions are global minimizers.*

Exercise 2.10 (Pointwise supremum preserves convexity)**.** ☕ *Let $(f_i)_I$ be a family of convex functions (not necessarily countable). Show that $x \mapsto \sup_{i \in I} f_i(x)$ is convex.*

Exercise 2.11 (Precomposition by linear operator preserves convexity)**.** ☕ *Let $f : \mathcal{Y} \to \mathbb{R}$ be a convex function and $A : \mathcal{X} \to \mathcal{Y}$ a linear operator. Show that $f(A\cdot)$ is convex (on $\mathcal{X}$).*

Exercise 2.12 (Misconceptions on existence of minimizers)**.** ☕ *Provide an example of convex function which does not admit a minimizer.*
What if the function is continuous and lower bounded?

Exercise 2.13. ☕ *Show that a strictly convex function has at most one minimizer.*

Exercise 2.14 (Jensen's inequality)**.** ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex. Let $n \in \mathbb{N}$, $x_1, \dots, x_n \in \mathbb{R}^d$, and let $\lambda_1, \dots, \lambda_n$ be positive scalars summing to 1.*
Show that $f(\sum_{i=1}^n \lambda_i x_i) \leq \sum_{i=1}^n \lambda_i f(x_i)$.

Exercise 2.15. ☕ *Show that a function is lower semicontinuous if its sublevel sets are closed.*

Exercise 2.16. ☕ *Show that the sublevel sets of a convex function are convex. Find a function with convex sublevel sets which is not convex.*

Exercise 2.17 (First order characterization of convex functions)**.** ☕☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a Gateaux differentiable function. Show that the following are equivalent:*

1. *f is convex*
2. *f lies above its tangents: $\forall x, y \in \mathbb{R}^d, f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle$*
3. *∇f is monotone: $\forall x, y \in \mathbb{R}^d, \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0$*

Exercise 2.18 (Continuity)**.** ☕☕☕ *Show that a convex function is locally Lipschitz (hence continuous) on the interior of its domain.*

Exercise 2.19 (Characterization of strongly convex functions in the Euclidean case)**.** ☕ *Show that f is μ-strongly convex with respect to the Euclidean norm if and only if $f - \frac{\mu}{2}\|\cdot\|^2$ is convex.*

Exercise 2.20. ☕ *Show that a strongly convex function admits exactly one minimizer.*

2.2.2 Gradient

Exercise 2.21. ☕ *Provide an example of setting where the gradient is not equal to the vector of partial derivatives.*

Exercise 2.22. ☕ *Show that the gradient of a function is orthogonal to the level lines of that function.*

Exercise 2.23. ☕ *Compute the gradients and Hessians of $x \mapsto \|x\|^2$, $x \mapsto \|x\|$, $x \mapsto a^\top x$. Is it true that the gradient of $x \mapsto \frac{1}{2} x^\top A x$ is equal to Ax?*

Exercise 2.24. ☕☕ *Compute the gradient of the logdet function, $M \mapsto \log\det(M)$.*

Exercise 2.25. ☕ *Let $A \in \mathbb{R}^{n \times d}$, $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$ defined as $g(x) = f(Ax)$ for all $x \in \mathbb{R}^d$. Show that*

$$\nabla g(x) = A^* \nabla f(Ax) \ ,$$
$$\nabla^2 g(x) = A^* \nabla^2 f(Ax) A \ .$$

3 Gradient descent on convex functions

In Section 1 we proved that for gradient descent on ordinary least squares we can get very fast (so-called "linear") convergence rates in terms of function values and iterates. Two numbers governed this convergence: global upper and lower bounds L and μ on the Hessian.

Our proof was very ad hoc, relying on the explicit expression of minimizer, the gradient and the Hessian. The questions we will try to answer in the sequel are:

- can we generalize the results of Proposition 1.3 to other objective functions?
- under which conditions?
- what are the objective's property that influence the convergence rate we obtain?

The first and weakest result we'll prove in this class concerns convex Lipschitz functions.

Proposition 3.1 (Gradient descent on Lipschitz convex functions)**.** *Let f be a convex, L-Lipschitz differentiable function admitting at least one minimizer x^*. For $t \in \mathbb{N}$, the iterates of gradient descent $x_{k+1} = x_k - \eta \nabla f(x_k)$ satisfy*

$$\min_{1 \leq k \leq t} f(x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|}{\sqrt{t}} \quad , \tag{32}$$

and

$$f(\tfrac{1}{t}\sum_{k=1}^{t} x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|}{\sqrt{t}} \quad , \tag{33}$$

if the stepsize η is taken as $\eta = \frac{\|x_0 - x^*\|}{L\sqrt{t}}$.

Let us make a few observations before the proof:

- This algorithm is a bit weird: the total number of iterations t must be known in advance to select the stepsize (we can get rid of this up to a slight worsening in the rate, with a decaying stepsize $\eta_t \propto 1/\sqrt{t}$).
- The objective values are not necessarily decreasing, it is not a *descent* algorithm.
- The more iterations we do, the smaller we need to take the stepsize.
- The stepsize depends on the unknown quantity $\|x_0 - x^*\|$. We can get rid of this dependency by bounding $\|x_0 - x^*\|$ with e.g. the diameter of the domain.
- The kind of rate is not as strong as in the OLS case: we know nothing about the last iterate x_t and only have results on the ergodic (averaged) iterates or on the best iterate.

Proof. Let x^* be a minimizer of f. For $k \in \mathbb{N}$, using convexity of f, definition of x_{k+1} and the parallelogram identity,

$$\begin{aligned} f(x_k) - f(x^*) &\leq \langle \nabla f(x_k), x_k - x^* \rangle & (34)\\ &\leq \frac{1}{\eta}\langle x_k - x_{k+1}, x_k - x^* \rangle & (35)\\ &\leq \frac{1}{2\eta}\left(\|x_k - x_{k+1}\|^2 + \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right) \quad . & (36) \end{aligned}$$

Summing, the telescopic series cancels out, and since f is L-Lipschitz we can bound $\|x_{k+1} - x_k\| = \eta\|\nabla f(x_k)\|$ by ηL, so:

$$\frac{1}{t}\sum_{k=1}^{t} f(x_k) - f(x^*) \leq \frac{1}{2t\eta}(t\eta^2 L^2 + \|x_0 - x^*\|^2 - \|x_{t+1} - x^*\|^2) \tag{37}$$

$$\leq \frac{\eta L^2}{2} + \frac{\|x_0 - x^*\|^2}{2t\eta} \tag{38}$$

Minimizing the RHS in η gives $\eta = \frac{R}{L\sqrt{t}}$ and the upper bound has value $\frac{RL}{\sqrt{t}}$. Jensen's inequality concludes. □

Proposition 3.2 (The projected gradient descent miracle)**.** *Let $\mathcal{C}$ be a non-empty closed convex subset of $\mathbb{R}^d$. Consider the following constrained minimization problem:*

$$\min_{x\in\mathcal{C}} f(x) \ , \tag{39}$$

and the projected gradient descent *algorithm:*

$$x_{k+1} = \Pi_{\mathcal{C}}(x_k - \eta\nabla f(x_k)) \ . \tag{40}$$

Then we have exactly the same rate as in Proposition 3.1: adding the constraint does not hurt the rate. Of course there is a caveat: we need to be able to compute $\Pi_{\mathcal{C}}$, which is an optimization problem in itself that can be quite hard. But for many $\mathcal{C}$'s it is OK (ℓ_1, ℓ_2, ℓ_∞ unit-balls).

Proof. Let us introduce the intermediate step $y_k = x_k - \eta\nabla f(x_k)$. Equation (36) becomes:

$$f(x_k) - f(x^*) \leq \frac{1}{2\eta}\left(\|x_k - y_k\|^2 + \|x_k - x^*\|^2 - \|y_k - x^*\|^2\right) \ . \tag{41}$$

Using $\|x_{k+1} - x^*\| \leq \|y_k - x^*\|$ (Exercise 3.18), we end up with exactly the same bound! □

The $1/\sqrt{k}$ rate is quite poor: to approach the optimal value within ε, one needs $\mathcal{O}(1/\varepsilon^2)$ iterations. With more assumptions on f, it can be improved to $\mathcal{O}(1/t)$, as we shall see in Proposition 3.6.

Definition 3.3 (L-smoothness)**.** *A differentiable function f is L-smooth if its gradient is L-Lipschitz: for all $x, y \in \operatorname{dom} f$,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \ . \tag{42}$$

A widely used property of L-smooth functions is that they satisfy the so-called Descent lemma.

Lemma 3.4 (Descent lemma)**.** *Let f be a L-smooth function. Then for all $x, y \in \operatorname{dom} f$,*

$$f(x) \leq f(y) + \langle\nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2 \ . \tag{43}$$

Proof. Notice that:

$$f(x) - f(y) = \int_0^1 \frac{d}{dt} f(y + t(x-y))dt = \int_0^1 \langle \nabla f(y + t(x-y)), x - y \rangle dt \ . \tag{44}$$

Subtract $\langle \nabla f(y), x - y \rangle$, use Cauchy-Schwarz and the hypothesis on f, conclude. □

Lemma 3.5 (Cocoercivity of the gradient)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a L-smooth function.*

$$\forall x, y \in \mathbb{R}^d, \quad \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\| \ . \tag{45}$$

Proposition 3.6 (Convergence rate of gradient descent on L-smooth functions)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex L-smooth function with non empty set of minimizers. Then the iterates of gradient descent with stepsize $< 2/L$ converge at the rate:*

$$f(x_k) - f(x^*) \leq \frac{2L\|x_k - x^*\|}{k} \ . \tag{46}$$

Proof. Let x^* be a minimizer of f. First, we show that the distance of x_k to x^* decreases:

$$\|x_{k+1} - x^*\| = \|x_k - x^*\|^2 + 2\langle x_k - x^*, x_{k+1} - x^* \rangle + \|x_{k+1} - x_k\|^2 \tag{47}$$

$$= \|x_k - x^*\|^2 - \frac{2}{L} \langle \nabla f(x_k), x_k - x^* \rangle + \eta^2 \|\nabla f(x_k)\|^2 \ . \tag{48}$$

Lemma 3.5, combined with $\nabla f(x^*) = 0$, yields:

$$-\frac{2}{L} \langle \nabla f(x_k), x_k - x^* \rangle + \frac{1}{L^2} \|\nabla f(x_k)\|^2 \leq -\frac{1}{2L^2} \|\nabla f(x_k)\|^2 \leq 0 \ , \tag{49}$$

which concludes: $\|x_{k+1} - x^*\| \leq \|x_k - x^*\|$.

We then use convexity of f:

$$f(x_k) - f(x^*) \leq \langle \nabla f(x_k), x_k - x^* \rangle \leq \|\nabla f(x_k)\| \|x_k - x^*\| \leq \|\nabla f(x_k)\| \|x_0 - x^*\| \ . \tag{50}$$

Finally the descent Lemma (Lemma 3.4) quantifies the decrease in objective at each iteration:

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L} \|\nabla f(x_k)\|^2 \ . \tag{51}$$

Introducing $\delta_k = f(x_k) - f(x^*)$, we have:

$$\delta_{k+1} - \delta_k \leq -\frac{1}{2L} \|\nabla f(x_k)\|^2 \tag{52}$$

$$\leq -\frac{1}{2L} \frac{\delta_k^2}{\|x_0 - x^*\|^2} \ . \tag{53}$$

The above manipulations are quite standard in optimization, but the following trick is a bit surprising the first time you see it: dividing by $\delta_k \delta_{k+1}$,

$$\frac{1}{\delta_k} - \frac{1}{\delta_{k+1}} \leq -\frac{1}{2L\|x_0 - x^*\|^2} \frac{\delta_k}{\delta_{k+1}} \tag{54}$$

$$\leq -\frac{1}{2L\|x_0 - x^*\|^2} \ , \tag{55}$$

since (δ_k) decreases (by (52)). Summing and telescoping concludes. □

If the function is even more regular, the rate of gradient descent can further be improved. By even more regular, we mean a stronger assumption on f, strong convexity.

Definition 3.7 (Strong convexity)**.** *Let $\mu > 0$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if for all $x, y \in \operatorname{dom} f$,*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - \frac{\mu}{2}\lambda(1-\lambda)\|x-y\|^2 \ . \tag{56}$$

Note: this is the only true definition of strong convexity. Other definitions are rather characterizations in special cases. In particular, many other definitions assume that the norm is the Euclidean one, a requirement that this definition does not have. This plays a role in Section 18.

Proposition 3.8. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and μ-strongly convex. Then it admits a unique minimizer (Exercise 2.20) and the iterates of gradient descent with stepsize $1/L$ converge linearly:*

$$f(x_k) - f(x^*) \leq (1 - \frac{\mu}{L})^k (f(x_0) - f(x^*)) \ . \tag{57}$$

Proof. Let x^* be the minimizer of f. Being μ-strongly convex, f satisfies the Polyak-Łojasiewicz inequality (Exercise 3.11):

$$\forall x \in \mathbb{R}^d, f(x) - f(x^*) \leq \frac{1}{2\mu}\|\nabla f(x)\|^2 \ . \tag{58}$$

Combined with the descent lemma,

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2 \tag{59}$$

$$\leq -\frac{\mu}{L}(f(x_k) - f(x^*)) \ . \tag{60}$$

hence $f(x_{k+1}) - f(x^*) \leq (1 - \frac{\mu}{L})(f(x_k) - f(x^*))$. □

3.1 Exercises

3.1.1 Convexity inequalities

Exercise 3.9. ☕☕ *Let f be a convex and Gateaux-differentiable function. Let $L > 0$. Show that the following properties are equivalent:*

1. $\forall x, y,\ \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$
2. $\forall x, y,\ f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2$
3. $\forall x, y,\ \frac{1}{L}\|\nabla f(x) - \nabla f(y)\| \leq \langle x - y, \nabla f(x) - \nabla f(y)\rangle$

Exercise 3.10. ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$, let $\mu > 0$. Show that f is μ-strongly convex function* with respect to the ℓ_2-norm *if and only if $f - \frac{\mu}{2}\|\cdot\|^2$ is convex.*
Provide a counter example when the underlying norm is not the Euclidean one.

Exercise 3.11 (Polyak-Łojasiewicz inequality)**.** ☕☕ *Let f be a μ-strongly-convex and differentiable function. Let $x^* = \operatorname{argmin} f(x)$. Show that f satisfies the Polyak-Łojasiewicz inequality:*

$$\mu(f(x) - f(x^*)) \leq \frac{1}{2}\|\nabla f(x)\|^2 \ . \tag{61}$$

Provide an example of function which is not strongly convex, but satisfies the inequality.

Exercise 3.12. ☕☕ *Let f be a L-smooth μ-strongly convex function. Show that for any x, y,*

$$\frac{\mu L}{\mu + L}\|x - y\|^2 + \frac{1}{L + \mu}\|\nabla f(x) - \nabla f(y)\|^2 \leq \langle x - y, \nabla f(x) - \nabla f(y)\rangle \ .$$

Exercise 3.13 (3 point descent lemma)**.** ☕ *Let f be convex and L-smooth. Show that for any triplet (x, y, z),*

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

3.1.2 Gradient descent

Exercise 3.14 (**??** in discrete time)**.** ☕☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex, twice differentiable L-smooth function.*
Show that the iterates of gradient descent on f with step size $0 < \alpha < 2/L$ have decreasing gradient norm.
Can you show it if f is not twice differentiable?
Is it still true when f is not convex?

Exercise 3.15. ☕ *Provide a finite dimensional example of convex L-smooth function f such that gradient descent with stepsize $< 2/L$ diverges.*

3.1.3 Constrained optimization

Exercise 3.16. ☕ *Show that the indicator function ι_C is convex (resp. lower semicontinuous, resp. proper) if C is convex (resp. closer, resp. nonempty).*

Exercise 3.17 (Global optimality condition for constrained convex optimisation)**.** ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex differentiable function, let C be a convex subset of $\mathbb{R}^d$. Show that $x^* \in \operatorname{argmin}_{x \in C} f(x)$ if and only if*

$$\forall x \in C, \langle \nabla f(x^*), x - x^* \rangle \geq 0 \ .$$

Exercise 3.18. ☕☕ *Let C be a non-empty closed convex set. Show that for all $x \in C, y \in \mathbb{R}^d$,*

$$\|x - y\|^2 \geq \|x - \Pi_C(y)\|^2 + \|y - \Pi_C(y)\|^2 \ .$$

In particular this shows that $\|y - x\| \geq \|x - \Pi_C(y)\|$, which says that projection can only get you closer to optimum (taking $x = x^*$, if your current iterate is y).