Optimization for large scale machine learning

Mathurin Massias
mathurin.massias@gmail.com

Contents

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 calls for dedicated classes of algorithms. They also usually present some specific structure, that can be leveraged.

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 meanings of the star symbol (Fenchel conjugation, adjoint), it happens that 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 anyway a good exercise for your mathematical dexterity to adapt to different notation.

0 Reminders

0.1 Calculus

Definition 0.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable. Then for any $x \in \mathbb{R}^d$, there exists a vector of $\mathbb{R}^d$, denoted $\nabla f(x)$ and called the gradient of f at x, such that for all $h \in \mathbb{R}^d$,*

$$f(x+h) = f(x) + \langle \nabla f(x), h \rangle + o(\|h\|) \tag{0.1}$$

This vector is unique.

Definition 0.2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable. Then for any $x \in \mathbb{R}^d$, there exists a matrix in $\mathbb{R}^{d \times d}$, denoted $\nabla^2 f(x)$ and called the Hessian of f at x, such that for all $h \in \mathbb{R}^d$,*

$$f(x+h) = f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2} h^\top \nabla^2 f(x) h + o(\|h\|^2) \tag{0.2}$$

As the gradient, if the Hessian exists it is uniquely defined.

If f is vector-valued, there is a generalization of the gradient, called the Jacobian.

Definition 0.3 (Jacobian)**.** *Jacobian of $f : x \in \mathbb{R}^n \mapsto \begin{pmatrix} f_1(x) \\ \dots \\ f_m(x) \end{pmatrix}$ is the matrix of partial derivatives:*

$$\left(\frac{\partial f_i}{\partial x_j}(x) \right)_{ij} \in \mathbb{R}^{m \times n} \ . \tag{0.3}$$

For real-valued functions, the Jacobian is the *transposed* gradient. Else, the Jacobian consists of stacked transposed gradients:

$$\mathcal{J}_f(x) = \begin{pmatrix} — \nabla f_1(x)^\top — \\ \dots \\ — \nabla f_n(x)^\top — \end{pmatrix} \tag{0.4}$$

Jacobian of gradient is Hessian (transposed, equal if f twice differentiable by Clairaut/Schwarz theorem; holds if second partial derivatives are continuous or more weakly if partial derivatives are continuous).

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}$$

$$\mathcal{J}_{\nabla f}(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{pmatrix}$$

Proposition 0.4 (Chain rule)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ be differentiable functions. Then their composition $g \circ f$ is differentiable and;*

$$\mathcal{J}_{g \circ f}(x) = \mathcal{J}_g(f(x)) \, \mathcal{J}_f(x) \ . \tag{0.5}$$

0.2 Linear algebra

Theorem 0.5 (Spectral theorem)**.** *Let $M \in \mathbb{R}^{d\times d}$ be a symmetric real matrix. Then it can be diagonalized in an orthonormal basis: there exists $U \in \mathbb{R}^{d\times d}$ and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ such that $UU^\top = U^\top U = \mathrm{Id}_d$ and*

$$M = U^\top \Lambda U \tag{0.6}$$

This is the most important theorem in this section. It is very convenient to diagonalize a matrix in order to raise it to some power $M^k = U^\top \lambda^k U$, and to simplify computations up to a change of basis.

Definition 0.6. *A **symmetric** matrix M is said to be positive semidefinite (PSD, written $M \succeq 0$) if for all $x \in \mathbb{R}^d$, $x^\top M x \geq 0$. If the inequality holds strictly for all nonzero x, M is said to be positive definite (PD, written $M \succ 0$).*

Note that by definition, a PSD matrix must be symmetric; also notice that for a generic A, the antisymmetric part of A has no influence of $x^\top A x$, so it's not really a restriction.

Example 0.7. *The typical way to create a PSD matrix is to pick $M = A^\top A$ for A of any shape, because then $x^\top M x = x^\top A^\top A x = \|Ax\|^2 \geq 0$. By this formulation, we deduce that $A^\top A$ is PD if and only if A is injective.*

Proposition 0.8 (Characterization of PSD matrices)**.** *A matrix $M \in \mathbb{R}^{d\times d}$ is PSD if and only all its eigenvalues are positive.*

Definition 0.9 (Loewner order)**.** *The cone of PSD matrices induces a partial order on symmetric matrices: for $A, B \in \mathbb{R}^{d\times d}$ symmetric, we write $A \succeq B$ if and only if $A - B \succeq 0$. This order is not total (find an example of two matrices which cannot be compared).*

Apart from SPD matrices, another class of very useful matrices are rank one matrices.

Proposition 0.10. *The matrix $M \in \mathbb{R}^{d\times d}$ is rank one if and only if it writes $M = xy^\top$ for $x, y \in \mathbb{R}^d$, both nonzero. If M is both symmetric and rank one, then it writes $xx^\top$ for $x \in \mathbb{R}^d$.*

Exercise 0.1. *What are the eigenvalues of a rank one matrix? Of a symmetric rank one matrix?*

Remark 0.11 (Writing a matrix as sum of rank one matrices)**.** *Let $M \in \mathbb{R}^{n\times p}$, $N \in \mathbb{R}^{p\times q}$. Then $MN = \sum_{i=1}^p M_{:i} N_{i:}$.*

0.3 Exercises

Exercise 0.2. ☕ *Provide an example of a setting where the gradient is not equal to the vector of partial derivatives.*

Exercise 0.3. ☕ *Show that the gradient of a function is orthogonal to the level lines of that function.*

Exercise 0.4. ☕ *For a differentiable function f, compute the gradient and Hessian of f^2.*

Exercise 0.5. ☕ *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 0.6. ☕ *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *and* $g : \mathbb{R} \to \mathbb{R}$ *be differentiable.*
Show that $\nabla g \circ f(x) = g'(f(x)) \nabla f(g(x))$.

Exercise 0.7. ☕ *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*

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

Exercise 0.8. ☕☕ *Compute the gradient of the logdet function,* $M \mapsto \log\det(M)$.

1 Motivation: gradient descent on ordinary least squares

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

$$\inf_{x\in\mathcal{C}} f(x) \ , \tag{1.1}$$

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 largest lower bound, called the *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 than the smallest value taken by f, we are more interested in finding a minimizer, that is

$$x^* \in \operatorname*{argmin}_{x\in\mathcal{C}} f(x) \ , \tag{1.2}$$

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 represents the parameters of a model, that once computed are used to achieve some statistical task.

1.1 Why does statistical learning need optimization?

Let $n \in \mathbb{N}$. Suppose that we have collected a dataset $(a_i, b_i)_{i\in[n]}$ of observation-target pairs, and we postulate that they are roughly linearly related, i.e. there exists $x \in \mathbb{R}^d$ such that $b_i \approx a_i^\top x$. This "$\approx$" is very vague. To make it more precise and rigorous, it is frequent to model the link as

$$\forall i \in [n],\ b_i = x_{\text{true}}^\top a_i + \varepsilon_i \tag{1.3}$$

with $(\varepsilon_i)_{i\in[n]}$ n independent realizations of a centered Gaussian variable of variance σ^2.

Given the collected observations, how can we infer x_{true}? From a statistical point of view, an answer is to maximize the conditional likelihood. Under model (1.3), the conditional likelihood is:

$$\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|a_i^\top x - b_i\|^2}{2\sigma^2}\right) \ . \tag{1.4}$$

Minimization of the negative log-likelihood in x gives

$$x^* \in \operatorname*{argmin}_{x\in\mathbb{R}^d} \sum_{1}^{n} (b_i - a_i^\top x)^2 = \operatorname{argmin} \frac{1}{2}\|Ax - b\|^2 \ , \tag{1.5}$$

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 (5) is called Ordinary Least Squares (OLS). Solving it gives an estimator for x_{true} and allows, for example, predicting a value b_{n+1} if one observes a new input point a_{n+1}. 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.

How to compute the solution of Problem (5)? This is a classical problem in linear algebra and there exist direct ways to do it (Exercise 1.2). But in other settings there may not exist

a closed-form solution: it is perfectly possible to postulate a different generative model than (1.3), 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 1.1 (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 (5) 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:*

$$\operatorname*{argmin}_{x\in\mathbb{R}^d} \sum_{i=1}^{n} |a_i^\top x - b_i| = \operatorname*{argmin}_{x\in\mathbb{R}^d} \|Ax - b\|_1 \ . \tag{1.6}$$

Problem (6) 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 (5) and (6) involves the Huber loss (Figure 1):

$$\operatorname*{argmin}_{x\in\mathbb{R}^d} \sum_{1}^{n} \mathrm{huber}_\rho(a_i^\top x - b_i) \ , \tag{1.7}$$

with the Huber loss defined as:

$$\mathrm{huber}_\rho(x) = \begin{cases} \frac{x^2}{2\rho} + \frac{\rho}{2} \ , & \text{if } |x| \le \rho \ , \\ |x| \ , & \text{otherwise.} \end{cases} \tag{1.8}$$

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.

For both problems, there is no closed-form solution and we need optimization techniques to approximate the minimizer.

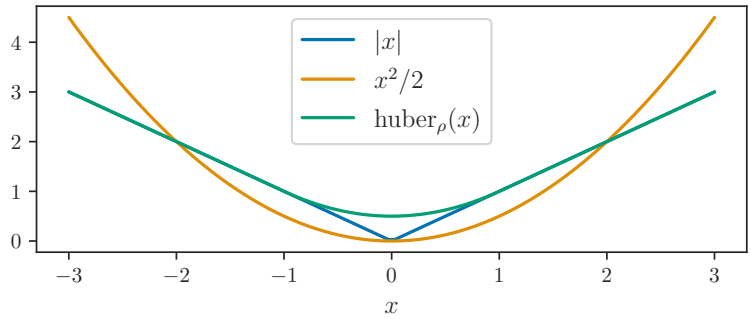

Figure 1: ℓ_1, squared ℓ_2 and Huber ($\rho = 1$) losses

Finally, instead of a linear model, one may model the dependency between observation 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:

$$b_i \approx f_x(a_i) \ , \tag{1.9}$$

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:

$$x^* \in \operatorname*{argmin}_{x \in \mathbb{R}^p} \sum_{i=1}^{n} L(f_x(a_i), b_i) \ . \tag{1.10}$$

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).

Example 1.2 (Logistic regression)**.** *When working with classification data ($b_i \in \{0, 1\}$), a very popular model is logistic regression*[1]*, which has the following statistical interpretation. Suppose that the law of each b_i is a Bernoulli of parameter $\sigma(x^\top a_i)$, with σ the sigmoid function: $\sigma(x) = \frac{1}{1+\exp(-x)}$. This model thus assumes that $\mathbb{P}(b_i = 1) = \sigma(x^\top a_i)$. Using $\sigma(-u) = 1-\sigma(u)$, one immediately has $\mathbb{P}(b_i = 0) = 1-\mathbb{P}(b_i = 1) = 1-\sigma(x^\top a_i) = \sigma(-x^\top a_i)$.*

The likelihood of the observations is:

$$\prod_{i:b_i=1} \mathbb{P}(b_i = 1) \prod_{i:b_i=0} \mathbb{P}(b_i = 0) = \prod_{i=1}^{n} \mathbb{P}(b_i = 1)^{b_i} \mathbb{P}(b_i = 0)^{1-b_i} \, , \tag{1.11}$$

hence the negative log-likelihood, also known as the binary cross-entropy*, is*

$$- \sum_{i=1}^{n} b_i \log\left(\frac{1}{1+\exp(-a_i^\top x)}\right) + (1-b_i) \log\left(\frac{1}{1+\exp(a_i^\top x)}\right) \tag{1.12}$$

$$= - \sum_{i=1}^{n} b_i \log\left(\frac{1+\exp(a_i^\top x)}{1+\exp(-a_i^\top x)}\right) + \log\left(\frac{1}{1+\exp(a_i^\top x)}\right) \tag{1.13}$$

$$= \sum_{i=1}^{n} -b_i a_i^\top x + \log\left(1+\exp(a_i^\top x)\right) \ . \tag{1.14}$$

Minimizing the above function defines the logistic regression estimator.

Remark 1.3. *In the case where the labels are in $\{-1, 1\}$, the negative log likelihood is given by:*

$$\sum_{i:b_i=1} \log(1+\exp(-x^\top a_i)) + \sum_{i:b_i=1} \log(1+\exp(x^\top a_i)) \tag{1.15}$$

$$= \sum_{i=1}^{n} \log(1+\exp(-b_i a_i^\top x)) \, . \tag{1.16}$$

[1] despite the name, this is a model for classification and not for regression

1.2 Solving least squares with gradient descent

Problem (5) can be solved in closed-form with dedicated algorithms such as Cholesky, LU or QR decompositions (see Exercise 1.2); 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 algorithms is gradient descent (GD). If it exists, the gradient of a function points towards 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

$$f'(x;v) = \langle \nabla f(x), v \rangle \ . \tag{1.17}$$

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:

$$\begin{aligned} &x_0 \in \mathbb{R}^d \ , \\ &x_{t+1} = x_t - \eta \nabla f(x_t) \ . \end{aligned} \tag{1.18}$$

Proposition 1.4 (Convergence rate of GD on OLS)**.** *Consider the OLS problem, and assume that there exist strictly positive scalars μ and L such that $\mu \mathrm{Id} \preceq A^\top A \preceq L\mathrm{Id}$. Let $\kappa = L/\mu$ denote the* condition number*. Then there exists a unique minimizer of least squares, x^*, and the iterates of gradient descent on ordinary least squares with stepsize $\eta = 1/L$ satisfy:*

$$\|x_t - x^*\| \leq \exp(-t/\kappa) \|x_0 - x^*\| \ . \tag{1.19}$$

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

$$x_{t+1} = x_t - \eta A^\top (A x_t - b) \ . \tag{1.20}$$

There exists a unique minimizer x^*, and it satisfies $A^\top A x^* = A^\top b$ (Exercises 1.1 and 1.2), so:

$$x_{t+1} - x^* = (\mathrm{Id} - \eta A^\top A)(x_t - x^*) \ . \tag{1.21}$$

Observe that

$$(1 - \eta L)\mathrm{Id} \preccurlyeq \mathrm{Id} - \eta A^\top A \preccurlyeq (1 - \eta\mu)\mathrm{Id} \ , \tag{1.22}$$

hence

$$\|\mathrm{Id} - \eta A^\top A\|_2 \leq q(\eta) \triangleq \max(|1 - \eta L|, |1 - \eta\mu|) \ . \tag{1.23}$$

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

Remark 1.5. *The convergence rate (1.19), is a very good rate in the sense that it decays fast towards 0. It is called a* linear *rate*[2]*. Suppose that we want to get ε-close to x^*, then we need at most $\kappa \log(\|x_0 - x^*\|/\varepsilon)$ iterations of GD, which grows quite slowly as ε goes to 0.*

[2]this is a terrible name: it is linear in log scale

We can also obtain a rate in objective value:

$$f(x_t) - f(x^*) = \langle \nabla f(x^*), x_t - x^* \rangle + \frac{1}{2}(x_t - x^*)A^\top A(x_t - x^*) \tag{1.24}$$

$$= \frac{1}{2}(x_t - x^*)A^\top A(x_t - x^*) \tag{1.25}$$

$$\leq \frac{L}{2}\|x_t - x^*\|^2 \ , \tag{1.26}$$

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

Remark 1.6 (Other choices of η.)**.** *We have actually proven the stronger result:*

$$\|x_t - x^*\| \leq q(\eta)^t \|x_0 - x^*\| \ . \tag{1.27}$$

This shows that any choice of η in $]0, 2/L[$ leads to convergence, because it results in $q(\eta) < 1$. We can tune η to minimize $q(\eta)$: doing so yields $\eta = \frac{2}{L+\mu}$ and $q(\eta) = \frac{L-\mu}{L+\mu} = \frac{\kappa-1}{\kappa+1}$.

Questions: Does there always exist an L as in the assumptions of Proposition 1.4? 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 1.4 applies. Yet the rate we see is clearly not linear. Why?

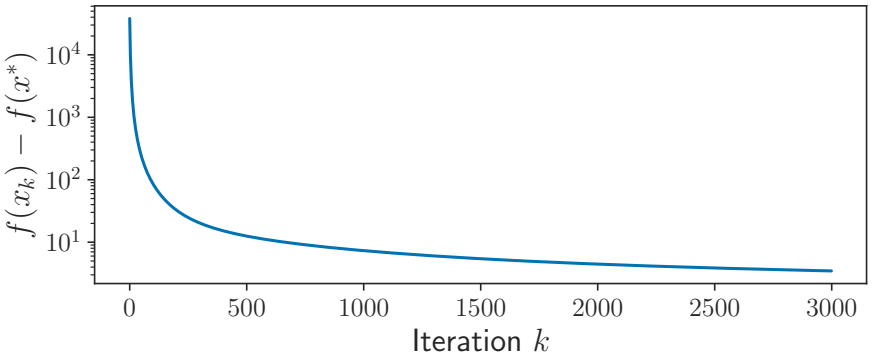

Figure 2: Objective convergence rate of GD on OLS on a random i.i.d. $A \in \mathbb{R}^{500\times500}$.

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?

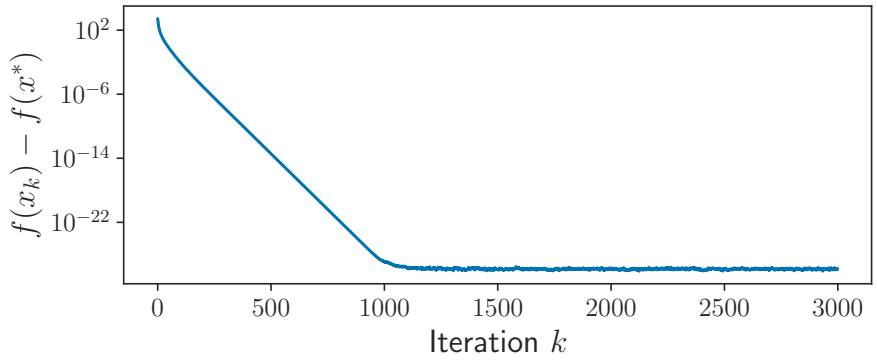

Figure 3: Objective convergence rate of GD on OLS on a random i.i.d. $A \in \mathbb{R}^{200\times 300}$.

1.4 Exercises

Exercise 1.1. ☕ *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 1.2. ☕ *Let $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$. Show that solving Ordinary Least Squares:*

$$\min \frac{1}{2}\|Ax - b\|^2 \ , \tag{1.28}$$

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

Exercise 1.4 (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,*

$$(\hat{x}, \hat{x}_0) \in \operatorname*{argmin}_{x\in\mathbb{R}^d, x_0\in\mathbb{R}} \frac{1}{2}\|Ax - b - x_0\mathbf{1}\|^2 \tag{1.29}$$

is given by:

$$\hat{x} = \hat{x}_c \ , \tag{1.30}$$

$$\hat{x}_0 = \frac{1}{n}\sum_1^n (a_i^\top \hat{x} - b_i) \ , \tag{1.31}$$

where $\hat{x}_c$ is the solution of least squares without intercept on centered data $A_c = A - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top A$ and $b_c = b - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top b$ (versions of A and b where the mean per column has been subtracted).

Exercise 1.5 (Gradient descent on isotropic parabola). ☕ *Let $A \in \mathbb{R}^{n\times d}$ be such that the condition number*[3] *of $A^\top A$ 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 aims at solving an optimization problem. For the Ordinary Least Squares objective function, 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 results were *non-asymptotic*: we had information about the speed at which convergence happened.

In the sequel, we want to minimize functions beyond least squares. For starters, we'll work with convex functions, because they're roughly the only ones we can hope to minimize globally.

2.1 Convexity and 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 $x \in V$.*

Definition 2.2 (Convexity)**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if it lies below its cords:*

$$\forall x, y \in \mathbb{R}^d, \forall \lambda \in]0,1[\,, \quad f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \ . \tag{2.1}$$

It is strictly convex if the inequality (2.1) holds strictly.

Remark 2.3. *Equation (2.1) trivially holds for $\lambda = 0$ and $\lambda = 1$, and so we could equivalently define it for $\lambda \in [0,1]$. However, we will later work with functions that can take the value $+\infty$, and in that case, we could end up with $\lambda f(x) = 0 \times (+\infty)$ which is not defined. So it is more rigorous to require the convexity inequality to hold only for $\lambda \in]0,1[$.*

Definition 2.4 (Strong convexity)**.** *Let $\mu > 0$ and $\|\cdot\|$ be a norm. A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex with respect to $\|\cdot\|$ if for all $x, y \in \mathbb{R}^d$ and $\lambda \in]0,1[$,*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \frac{\mu}{2}\lambda(1-\lambda)\|x-y\|^2 \ . \tag{2.2}$$

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 will play an important role in Mirror Descent Section 11.*

When $\|\cdot\|$ is the Euclidean norm, then f is μ-strongly convex if and only if $f - \frac{\mu}{2}\|\cdot\|^2$ is convex (Exercise 2.11). In this sense, a strongly convex function is so convex *that when you subtract a parabola with positive curvature, it remains convex.*

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

Proposition 2.6 (Above its tangents)**.** *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. This property is sometimes called "local to global": from local information $\nabla f(x)$, we get information about the global behavior of f.

Convex differentiable functions are nice because it is easy to characterize their minimizers.

Proposition 2.7 (Optimality condition for convex differentiable functions)**.** *Let f be a convex differentiable function. Then*

$$x^* \in \operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \Leftrightarrow \nabla f(x^*) = 0 \ . \tag{2.3}$$

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

2.2 Exercises

2.2.1 Convexity

Exercise 2.1. ☕ *Show that local minimizers of convex functions are global minimizers.*

Exercise 2.2 (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 2.3 (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 2.4 (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 2.5. ☕ *Show that a strictly convex function has at most one minimizer.*

Exercise 2.6 (Jensen's inequality)**.** ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex. Let $n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathbb{R}^d$, and let $\lambda_1, \ldots, \lambda_n$ 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 2.7. ☕ *Show that the sublevel sets of a convex function are convex. Find a function with convex sublevel sets which is not convex.*

Exercise 2.8. ☕☕ *A function is said to be* midpoint convex *if for all x, y,*

$$f\left(\frac{x+y}{2}\right) \leq \frac{1}{2}f(x) + \frac{1}{2}f(y)\,. \tag{2.4}$$

Show that a continuous midpoint convex function is convex.
What happens if you remove the continuity assumption? (this is ☕☕☕)

Exercise 2.9 (First order characterization of convex functions)**.** ☕☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a 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 2.10 (Continuity)**.** ☕☕☕ *Show that a convex function is locally Lipschitz (hence continuous) on the interior of its domain.*

Exercise 2.11 (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.*
☕☕ *Show that this does not always hold if the norm is not Euclidean.*

Exercise 2.12. ☕ *Show that a strongly convex function admits exactly one minimizer.*

Exercise 2.13 (A convenient equality)**.** *Show that, for any $x, y \in \mathbb{R}^d$ and any λ (not necessarily in $[0, 1]$),*

$$\|\lambda x + (1-\lambda)y\|^2 = \lambda\|x\|^2 + (1-\lambda)\|y\|^2 - \lambda(1-\lambda)\|x-y\|^2\,. \tag{2.5}$$

In which sense is it a generalization of the parallelogram identity?

2.2.2 Gradient

Exercise 2.14. ☕ *Provide an example of a setting where the gradient is not equal to the vector of partial derivatives.*

Exercise 2.15. ☕ *Show that the gradient of a function is orthogonal to the level lines of that function.*

Exercise 2.16. ☕ *For a differentiable function f, compute the gradient and Hessian of f^2.*

Exercise 2.17. ☕ *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 $x \mapsto \frac{1}{2}x^\top Ax$ is equal to Ax?*

Exercise 2.18. ☕ *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be differentiable.*
Show that $\nabla g \circ f(x) = g'(f(x))\nabla f(g(x))$.

Exercise 2.19. ☕ *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*

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

Exercise 2.20. ☕☕ *Compute the gradient of the logdet function, $M \mapsto \log\det(M)$.*

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 the minimizer, the gradient and the Hessian. The questions we will try to answer in the sequel are:

- Can we generalize the results of Proposition 1.4 to other objective functions?
- Under which conditions?
- What are the properties that influence the convergence rate we obtain?

3.1 Basic properties and convergence rates for Lipschitz, smooth and smooth+strongly convex functions

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

Remark 3.1. *The following proposition in fact applies to* subgradient *descent, a more generic algorithm which does not require the function to be differentiable. This notion will be introduced in Section 6.3.*

Proposition 3.2 (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*

$$\min_{1 \leq k \leq t} f(x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|}{\sqrt{t}} \quad , \tag{3.1}$$

and

$$f(\tfrac{1}{t}\sum_{k=1}^{t} x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|}{\sqrt{t}} \quad , \tag{3.2}$$

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 step size (in Exercise 3.10 we get rid of this up to a slight worsening in the rate, using a decaying stepsize $\eta_k \propto 1/\sqrt{k}$).
- 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 step size.
- 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,

$$f(x_k) - f(x^*) \leq \langle \nabla f(x_k), x_k - x^* \rangle \tag{3.3}$$

$$\leq \frac{1}{\eta} \langle x_k - x_{k+1}, x_k - x^* \rangle \tag{3.4}$$

$$\leq \frac{1}{2\eta} \left(\|x_k - x_{k+1}\|^2 + \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right) . \tag{3.5}$$

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:

$$\frac{1}{t} \sum_{k=1}^{t} f(x_k) - f(x^*) \leq \frac{1}{2t\eta} (t\eta^2 L^2 + \|x_0 - x^*\|^2 - \|x_{t+1} - x^*\|^2) \tag{3.6}$$

$$\leq \frac{\eta L^2}{2} + \frac{\|x_0 - x^*\|^2}{2t\eta} \tag{3.7}$$

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

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 $x, y \in \operatorname{dom} f$,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \ . \tag{3.8}$$

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 $x, y \in \operatorname{dom} f$,*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2 \ . \tag{3.9}$$

Think of this inequality for a y fixed, and involving two functions of x, f and $\phi_y = f(y) + \langle \nabla f(y), \cdot - y \rangle + \frac{L}{2} \| \cdot - y\|^2$. The function ϕ_y is a convex, isotropic parabola. Equation (3.9) says that ϕ_y globally upper bounds f; in addition, the two functions are equal and tangent at y.

Proof. Notice that:

$$f(x) - f(y) = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} f(y + t(x - y)) \, \mathrm{d}t = \int_0^1 \langle \nabla f(y + t(x - y)), x - y \rangle \, \mathrm{d}t \ . \tag{3.10}$$

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

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

$$\forall x, y \in \mathbb{R}^d, \quad \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\| \ . \tag{3.11}$$

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 nonempty set of minimizers. Then the iterates of gradient descent with stepsize $1/L$ converge at the rate:*

$$f(x_k) - f(x^*) \leq \frac{2L\|x_0 - x^*\|}{k} \ . \tag{3.12}$$

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

$$\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 + 2\langle x_k - x^*, x_{k+1} - x_k \rangle + \|x_{k+1} - x_k\|^2 \tag{3.13}$$

$$= \|x_k - x^*\|^2 - \frac{2}{L}\langle \nabla f(x_k), x_k - x^* \rangle + \frac{1}{L^2}\|\nabla f(x_k)\|^2 \ . \tag{3.14}$$

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

$$-\frac{2}{L}\langle \nabla f(x_k), x_k - x^* \rangle + \frac{1}{L^2}\|\nabla f(x_k)\|^2 \leq -\frac{1}{L^2}\|\nabla f(x_k)\|^2 \leq 0 \ , \tag{3.15}$$

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

We then use the convexity of f:

$$f(x_k) - f(x^*) \leq \langle \nabla f(x_k), x_k - x^* \rangle \leq \|\nabla f(x_k)\| \|x_k - x^*\| \leq \|\nabla f(x_k)\| \|x_0 - x^*\| \ . \tag{3.16}$$

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

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2 \ . \tag{3.17}$$

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

$$\delta_{k+1} - \delta_k \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2 \tag{3.18}$$

$$\leq -\frac{1}{2L}\frac{\delta_k^2}{\|x_0 - x^*\|^2} \ . \tag{3.19}$$

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

$$\frac{1}{\delta_k} - \frac{1}{\delta_{k+1}} \leq -\frac{1}{2L\|x_0 - x^*\|^2}\frac{\delta_k}{\delta_{k+1}} \tag{3.20}$$

$$\leq -\frac{1}{2L\|x_0 - x^*\|^2} \ , \tag{3.21}$$

since (δ_k) decreases (by (3.18)). Summing and telescoping concludes. □

If the function is even more regular, the rate of gradient descent can further be improved.

Proposition 3.7. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and μ-strongly convex. Then it admits a unique minimizer x^* (Exercise 2.12) and the iterates of gradient descent with stepsize $1/L$ converge linearly:*

$$f(x_k) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)^k (f(x_0) - f(x^*)) \ . \tag{3.22}$$

Note that, since $e^x \geq 1 + x$ (by convexity!), so $(1 - \frac{\mu}{L})^k \leq \exp(-\frac{\mu}{L}k)$, and this quantity is often used instead in (3.22). Since $\mu/L \leq 1$ and is usually very close to 0, one does not lose much in this majorization.

Proof. Let x^* be the minimizer of f. Being μ-strongly convex, f satisfies the Polyak-Łojasiewicz inequality (Exercise 3.5):

$$\forall x \in \mathbb{R}^d, f(x) - f(x^*) \leq \frac{1}{2\mu}\|\nabla f(x)\|^2 \ . \tag{3.23}$$

Combined with the descent lemma,

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2 \tag{3.24}$$

$$f(x_{k+1}) - f(x^*) + f(x^*) - f(x_k) \leq -\frac{\mu}{L}(f(x_k) - f(x^*)) \ . \tag{3.25}$$

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

Remark 3.8. *Using $f(x_k) - f(x^*) \leq \langle \nabla f(x_k), x_k - x^* \rangle = \langle \nabla f(x_k) - \nabla f(x^*), x_k - x^* \rangle \leq LR^2$ where $R \triangleq \|x_0 - x^*\|$, one obtains:*

$$f(x_k) - f(x^*) \leq LR^2 \left(1 - \frac{\mu}{L}\right)^k \tag{3.26}$$

3.2 Exercises

3.2.1 Convexity inequalities

Exercise 3.1. ☕☕ *Let f be a convex and 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. $\forall x, y,\ \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2 \leq \langle x - y, \nabla f(x) - \nabla f(y)\rangle$

Exercise 3.2. ☕ *Let f be a twice differentiable L-smooth function. Show that for all $x \in \mathbb{R}^d$, $\nabla^2 f(x) \preccurlyeq L\mathrm{Id}$.*

Exercise 3.3. ☕ *Let f be a differentiable function. Show that the following properties are equivalent:*

1. *f is μ-strongly convex (with respect to the Euclidean norm)*
2. $\forall x, y,\ f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2$
3. $\forall x, y,\ \mu\|x - y\|^2 \leq \langle x - y, \nabla f(x) - \nabla f(y)\rangle$

Exercise 3.4. ☕ *Let f be a twice differentiable μ-strongly convex function. Show that for all $x \in \mathbb{R}^d$, $\mu\mathrm{Id} \preccurlyeq \nabla^2 f(x)$.*

Exercise 3.5 (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:*

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

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

Exercise 3.6. ☕☕ *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 3.7 (3 point descent lemma)**.** ☕ *Let f be convex and L-smooth. Show that for any triplet (x, y, z),*

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

3.2.2 Gradient descent

Exercise 3.8 (Exercise 12.2 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 3.9. ☕ *Provide a finite-dimensional example of convex L-smooth function f such that gradient descent with stepsize $< 2/L$ diverges.*

Exercise 3.10 (Gradient descent on Lipschitz function without knowing the horizon)**.** ☕☕ *For gradient descent on a Lipschitz differentiable objective, the classical proof assumes that the total number of iterations t is known in advance, to set the fixed stepsize $\eta \propto 1/\sqrt{t}$. Show that using a decreasing stepsize $\eta_k \propto 1/\sqrt{k}$ leads to a rate of order $\log k/\sqrt{k}$ on the ergodic or best iterate.*

Exercise 3.11 (Gradient descent is equivariant to rotation)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and Lipschitz differentiable (not required for the exercise, just to be in a setup where gradient descent is relevant). Let U be a rotation matrix ($U^\top U = \mathrm{Id}$) and $\phi : \mathbb{R}^d \to \mathbb{R}$ a reparametrization of f defined as $\phi = f(U\cdot)$.*
Show that gradient descent is equivariant to rotation: the iterates of GD of f and ϕ respectively, (x_k) and (y_k), started at $x_0 = Uy_0$, satisfy $x_k = Uy_k$ for all $k \in \mathbb{N}$. How do the respective losses compare?
Show that studying the convergence of gradient descent on a quadratic can be reduced to studying gradient descent on a separable quadratic (meaning, with diagonal Hessian).

4 Inertial acceleration: Nesterov and Heavy-ball

Very cool resource with animations: https://distill.pub/2017/momentum/

The rate of convergence of gradient descent for L-smooth convex functions is $\mathcal{O}(1/k)$); moreover if the function is μ strongly convex, the rate improves – without modifying the algorithm – to $\mathcal{O}(\exp(-\frac{\mu}{L}k))$. But this can still be slow: consider a function for which $\mu/L = 10^{-3}$. Then, to divide the suboptimality by 10, the number k of iterations to perform must be such that $\exp(-\frac{\mu}{L}k) \leq 0.1$, i.e. $k \geq \frac{L}{\mu}\log(10) \approx 2300$.

In this section, we introduce algorithms that are faster. Namely, their convergence rates are in $\mathcal{O}(1/k^2)$ and $\exp(-\sqrt{\frac{\mu}{L}})$. This means that in the same example, only $k \geq \sqrt{\frac{L}{\mu}}\log(10) \approx 72$ would be needed.

4.1 Polyak's heavy ball method

The heavy ball iterations are defined as:

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \underbrace{\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}(x_k - x_{k-1})}_{\text{momentum}} \tag{4.1}$$

In this algorithm, the iterate moves in the direction of the gradient through $\alpha\nabla f(x_k)$, but it also continues in the direction of its previous update step $x_k - x_{k-1}$. Actually, this is where heavy ball name get its name from: the momentum term can be seen as the inertia of a ball rolling down a valley.

In this class we will only apply it on quadratics, as it can diverge otherwise (see the negative result by Goujaud et al. (2023)). For results beyond quadratics, see Ghadimi et al. (2015) and the recent works of Apidopoulos et al. (2022); Aujol et al. (2023).

It may be enlightening to introduce $\gamma = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}} = \frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}$ and rewrite heavy ball as:

$$\begin{cases} m_{k+1} = \gamma m_k + (1-\gamma)\nabla f(x_k) & \left(= (1-\gamma)\sum_{i=0}^{k}\gamma^k\nabla f(x_{k-i})\right) \\ x_{k+1} = x_k - \frac{\alpha}{1-\gamma}m_{k+1} \end{cases} \tag{4.2}$$

which makes it visible that Heavy Ball replaces the gradient by a geometrically weighted average of past gradients. These two forms are equivalent: indeed unrolling the second equation we get

$$x_{k+1} = x_k - \frac{\alpha}{1-\gamma}\gamma m_k - \alpha\nabla f(x_k) \tag{4.3}$$

$$= x_k + \gamma(x_k - x_{k+1}) - \alpha\nabla f(x_k) \tag{4.4}$$

which is Equation (4.1).

Proposition 4.1. *On strongly convex quadratics with conditioning κ, and a proper choice of α, Heavy Ball has the convergence rate:*

$$f(x_k) - f(x^*) \leq Ck\left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^k \|x_0 - x^*\|^2 \tag{4.5}$$

Proof. Proof sketch: consider that the Hessian is diagonal wlog, write down update of m and x for an arbitrary coordinate. Show that a 2×2 matrix appears, put it in triangular form, find value of all powers of said matrix, bound its spectral norm by its absolute norm, show that the two values on its diagonal are complex conjugates, conclude. □

4.2 Nesterov acceleration

The heavy ball suffers from two flaws: it does not apply to functions beyond quadratics, and it does not apply to functions that are only convex (although, for convex quadratics, there is a straightforward adaptation with the lowest non zero eigenvalue, everything remaining constant in the kernel of the Hessian).

Nesterov's accelerated gradient, also called the Fast Gradient Method, solves these two issues. Schematically, the difference bet¡een heavy ball and Nesterov is the order in which gradient steps and momentum are applied: Nesterov does momentum then gradient, while heavy ball does gradient then momentum.

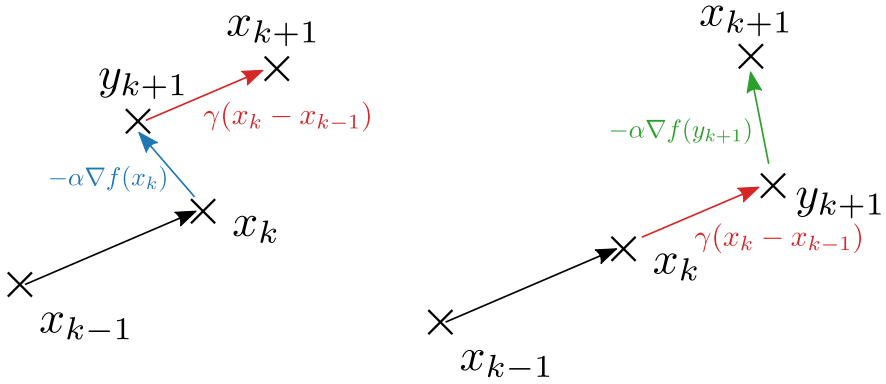

Figure 4: Left: heavy ball (gradient, then momentum). Right: Nesterov: momentum, then gradient.

Proposition 4.2 (Nesterov with strong convexity)**.** *Let f be μ-strongly convex and L-smooth. Consider the following inertial algorithm: $y_0 = x_0$, and for $k \in \mathbb{N}^*$,*

$$\begin{cases} x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k) \\ y_{k+1} = x_{k+1} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}(x_{k+1} - x_k) \end{cases} \tag{4.6}$$

It gives the rate:

$$f(x_k) - f(x^*) \leq 2(1 - \sqrt{\mu}L)^k (f(x_0) - f(x_k)) \tag{4.7}$$

Remark 4.3.

- *note that as κ approaches 1 (the function becomes well-conditioned), we need less and less extrapolation/acceleration and the algorithm gets closer to gradient descent.*
- *beware: the convergence rates holds for (x_k), the sequence after the gradient step, but not for (y_k). For recent work on this question, see Attouch and Fadili (2022)*
- *using the algorithm requires knowledge of μ; when μ is unknown, it can be estimated TODO add literature estimating μ*

- *leads to oscillating behavior in practice. TODO literature on restart.*

Proposition 4.4 (Nesterov without strong convexity)**.** *Let f be convex and L-smooth. Let the sequence (t_k) be defined by $t_0 = 1$, and $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$ (mysterious choice... let us only observe that it is equivalent to $t_k^2 = t_{k+1}^2 - t_{k+1}$, which makes the proof work[5]).*

Consider the following algorithm: $y_0 = x_0$, and for $k \in \mathbb{N}^*$,

$$\begin{cases} x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k) \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k) \end{cases} \tag{4.8}$$

Its iterates satisfy:

$$f(x_k) - f(x^*) \leq \frac{2L}{(k+1)^2}\|x_0 - x^*\|^2 . \tag{4.9}$$

Proof. Use the 3 point descent lemma (Exercise 3.7:

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(y_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - y\|^2 \tag{4.10}$$

$$f(x_{k+1}) - f(x_k) \leq L\langle y_k - x_{k+1}, x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - y_k\|^2 \tag{4.11}$$

$$f(x_{k+1}) - f(x_k) \leq L\langle y_k - x_{k+1}, y_k - x_k \rangle - \frac{L}{2}\|x_{k+1} - y_k\|^2 \tag{4.12}$$

Use it a second time, with x^* instead of x_k:

$$f(x_{k+1}) - f(x^*) \leq L\langle y_k - x_{k+1}, y_k - x^* \rangle - \frac{L}{2}\|x_{k+1} - y_k\|^2 \tag{4.13}$$

Let $\delta_k = f(x_k) - f(x^*)$. Multiply (4.12) by $(t_k - 1)$ and add to (4.13):

$$t_k\delta_{k+1} - (t_k - 1)\delta_k \leq L\langle y_k - x_{k+1}, t_k y_k - (t_k - 1)x_k - x^* \rangle - \frac{L}{2}t_k\|y_k - x_{k+1}\|^2 \tag{4.14}$$

Multiplying by t_k, using the equality satisfied by t_k, and using the parallelogram identity:

$$t_k^2\delta_{k+1} - t_{k-1}^2\delta_k \leq \frac{L}{2}\left(2\langle t_k(y_k - x_{k+1}), t_k y_k - (t_k - 1)x_k - x^* \rangle - \|t_k(y_k - x_{k+1})\|^2\right) \tag{4.15}$$

$$= \frac{L}{2}\left(\|t_k y_k - (t_k - 1)x_k - x^*\|^2 - \|t_k x_{k+1} - (t_k - 1)x_k - x^*\|^2\right) \tag{4.16}$$

Letting $u_k = t_k y_k - (t_k - 1)x_k - x^*$, we obtain, using $t_k x_{k+1} - (t_k - 1)x_k = t_{k+1}y_{k+1} - (t_{k+1} - 1)x_k$:

$$t_k^2\delta_{k+1} - t_{k-1}^2\delta_k \leq \frac{L}{2}(\|u_k\|^2 - \|u_{k+1}\|^2) \tag{4.17}$$

Sum up and use that, by recursion, $t_k \geq \frac{k-1}{2}$. □

[5] inequality would be enough

5 Lower bounds for first-order methods

Gradient descent has rate $\mathcal{O}(1/k)$, Nesterov has rate $\mathcal{O}(1/k^2)$... Could we reach lower? No.

TODO: clarify, what if an alternative oracle was used? Dependency on init?

Definition 5.1 (First-order method)**.** *A first-order method to minimize the function f is an algorithm that, given access to an oracle returning $g^{(i)} \in \partial f(x^{(i)})$ for each query point $x^{(i)}$, produces a sequence of iterates such that $x^{(k+1)} \in \mathrm{Span}(g^{(0)}, \dots, g^{(k)})$. For simplicity, we also assume that $x^{(0)} = 0$.*

Proposition 5.2 (Lower bound for first-order methods, Lipschitz case)**.** *Let $k \leq d-1$. For any $L > 0, R > 0$, there exists a function f, L-Lipschitz on the ball of radius R, such that for any first-order method in the sense of Definition 5.1,*

$$\min_{t \leq k} f(x^{(t)}) - \min_{\|x\| \leq R} f(x) \geq \frac{RL}{2(1+\sqrt{k})}\,. \tag{5.1}$$

Crucial note: this is valid only for a number of iterations less than the dimension!

Proof. Let $\gamma \geq 0$, $\alpha \geq 0$. Consider the function

$$f(x) = \gamma \max_{1 \leq i \leq k} x_i + \frac{\alpha}{2}\|x\|^2\,. \tag{5.2}$$

Its subdifferential is given by:

$$\partial f(x) = \alpha x + \gamma\, \mathrm{conv}\{e_j : x_j = \max_{1 \leq i \leq k} x_i\}\,. \tag{5.3}$$

Hence, for $\|x\| \leq R$, subgradients have norm less than $\alpha R + \gamma$, and f is $(\alpha R + \gamma)$-Lipschitz on the ball of radius R.

Suppose that the first-order oracle returns as subgradient $\alpha x + \gamma e_i$, where i is the smallest coordinate such that $x_i = \max_{1 \leq j \leq k} x_j$. TODO clarify. Then, it is easy to see that $x^{(i)} \in \mathrm{Span}(e_1, \dots, e_i)$, and thus for $i < d$, $x^{(i)}_d = 0$ and $f(x^{(i)}) \geq 0$.

Now let us compute the minimal value of f. It is equal to $\min \gamma v + \frac{\alpha}{2}\|x\|^2$ s.t. $x_i - v \leq 0\, \forall i \in [k]$. Hence by introducing the Lagrangian $\mathcal{L}(t, x, \lambda) = \gamma v + \frac{\alpha}{2}\|x\|^2 + \sum_{i=1}^k \lambda_i (x_i - v)$, and using the first-order optimality conditions, one gets that the minimum is reached for x^* such that $x^*_i = -\frac{\gamma}{\alpha k}$ for $i \leq k$, and $x^*_i = 0$ otherwise, yielding $f(x^*) = -\frac{\gamma^2}{2\alpha k}$. Therefore for any first-order algorithm and $i \leq k$,

$$f(x^{(i)}) - f(x^*) \geq \frac{\gamma^2}{2\alpha t}\,. \tag{5.4}$$

Pick $\alpha = \frac{L}{R(1+\sqrt{k})}$ and $\gamma = \frac{L\sqrt{k}}{1+\sqrt{k}}$ to prove the result (one needs to check that $\|x^*\| \leq R$, which is the case). □

Note: the same function with $\gamma = L/2$ and $R = L/(2\alpha)$ can also be used to prove lower bounds for L-Lipschitz, α-strongly convex functions (see Bubeck et al. (2015, Thm 3.13)).

Proposition 5.3 (Lower bounds for first-order methods, smooth case)**.** *Let k be an integer such that $k \leq (d-1)/2$. For any $L > 0$, there exists a L-smooth function such that for any first-order method in the sense of Definition 5.1,*

$$\min_{t \leq k} f(x^{(t)}) - \min f \geq \frac{3L}{32} \frac{\|x_0 - x^*\|^2}{(k+1)^2} \,. \tag{5.5}$$

Again, be careful: this result is valid only for a number of iterations that does not exceed half the dimension.

Proof. For $i \leq d$, let H_i be the tridiagonal matrix with 2 on the diagonal and -1 on the over- and underdiagonal:

$$H_i = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix} \in \mathbb{R}^{i \times i} \tag{5.6}$$

Let $A_i \in \mathbb{R}^{d \times d}$ be the d by d matrix such that its upper left i by i block is equal to H_i, and the rest is 0. One has $0 \preceq A_i \preceq 4\mathrm{Id}_d$ since for all $x \in \mathbb{R}^d$:

$$x^\top A_i x = 2 \sum_{j=1}^{i} x_j^2 - 2 \sum_{j=1}^{i-1} x_j x_{j+1} \tag{5.7}$$

$$= \sum_{j=1}^{i-1} (x_j - x_{j+1})^2 + x_1^2 + x_i^2 \geq 0 \tag{5.8}$$

$$\leq 2 \sum_{j=1}^{i-1} (x_j^2 + x_{j+1}^2) + x_1^2 + x_i^2 \tag{5.9}$$

$$\leq 4 \sum_{j=1}^{i} x_j^2 \,. \tag{5.10}$$

Let the function f to optimize be defined by:

$$f(x) = \frac{L}{8} x^\top A_{2k+1} x - \frac{L}{4} x_1 \,. \tag{5.11}$$

Because $\nabla f(x) = \frac{L}{4}(A_{2k+1}x - e_1)$, we have that iterates of any first-order method will satisfy $x^{(t)} \in \mathrm{Span}(e_1, e_t)$. In particular, ${x^{(t)}}^\top A_{2k+1} x^{(t)} = {x^{(t)}}^\top A_t x^{(t)}$. Let $f_t(x) = \frac{L}{8} x^\top A_t x - \frac{L}{4} x_1$, with minimal value f_t^*. It is clear that f_t^* decreases with t. We have proved that

$$f(x^{(t)}) - f^* = f_t(x^{(t)}) - f^* \geq f_t^* - f_{2k+1}^* \geq f_k^* - f_{2k+1}^* \tag{5.12}$$

Now we compute the minimizer $x^{*,t}$ of f_t. It must satisfy $A_t x^{*t} = e_1$. To solve this linear system, show by recursion on i that $x^{*,t}_{t-i} = (i+1)x^{*,t}_t$, then use the condition $2x^{*,t}_1 - x^{*,t}_2 = 1$ to conclude that:

$$x^{*,t}_i = \begin{cases} 1 - \frac{i}{t+1}\,, & \text{if } 1 \le i \le t\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.13}$$

Thus $f^t_* = -\frac{L}{8} x^{*,t\top} e_1 + \frac{L}{4} x^{*,t\top} e_1 = -\frac{L}{8}\frac{k}{k+1}$.

Now

$$\|x^{*,t}\|^2 = \sum_1^t \left(1 - \frac{i}{t+1}\right)^2 = \sum_1^t \left(\frac{i}{t+1}\right)^2 = \frac{t(t+1)(2t+1)}{6(t+1)^2} \le \frac{t+1}{3} \tag{5.14}$$

Thus finally:

$$f^* k - f^*_{2k+1} = \frac{L}{8}\left(\frac{k}{k+1} - \frac{2k-1}{2k+2}\right) \tag{5.15}$$

$$= \frac{L}{8}\frac{1}{2(k+1)} \tag{5.16}$$

$$\ge \frac{3L}{32}\frac{x^{*,2k+1}}{(k+1)^2} \tag{5.17}$$

□

TODO last result, for L-smooth, μ-strongly convex, in infinite dimension. Point to Nesterov et al. (2018, Thm. 2.1.13).

6 Non-smooth convex optimization

For a differentiable objective function $f : \mathbb{R}^d \to \mathbb{R}$, we have seen:

1. convexity: f is above its tangents
2. Lipschitzness
3. L-smoothness, implying the descent lemma (Lemma 3.4), meaning that at any point $y \in \mathbb{R}^d$ there exists an isotropic parabola of curvature L that globally **upper** bounds f and is tangent to f at y.
4. strong convexity: at any point $y \in \mathbb{R}^d$ there exists an isotropic parabola of curvature μ that globally **lower** bounds f and is tangent to f at y.

Depending on the properties satisfied by f, we have:

- 1 gives the necessary and sufficient condition for x to be a minimizer: $\nabla f(x^*) = 0$.
- $1 + 2$ gives a convergence rate of $1/\sqrt{k}$ under stepsize depending on the total number of iterations. The rate is in objective value, for the best iterate or the averaged iterates.
- $1 + 3$ improves it to $1/k$ with stepsize $1/L$ not depending on the number of iterations. The rate is in objective value, on the last iterate.
- $3 + 4$ sandwiches the functions between two isotropic parabolas of curvature μ and L and gives the excellent linear rate. The rate is on the distance from the last iterate to the unique minimizer.

Now we want to consider non-differentiable functions f. Why do we need to handle these?

6.1 Constrained optimization

Optimizers sometimes want the minimizer of their cost functions to belong to some set $\mathcal{C} \subset \mathbb{R}^d$.

For example, suppose that the variable x corresponds to an image, stored as a 2D array of pixels – an element of $\mathbb{R}^{d \times d}$. Then the values of the pixels are not just any real numbers: they should be in $[0, 1]$ if they correspond to gray levels for example.

If the optimization variable $x \in \mathbb{R}^d$ represents proportions (say, of various types of cells in a sane/cancerous tissue), it should have non-negative entries that sum to 1: $x \in \Delta_d$, where Δ_d is the d-dimensional *simplex*

$$\Delta_d = \{x \in \mathbb{R}^d : \forall i \in [d],\, x_i \geq 0, \sum_{i=1}^{d} x_i = 1\} \ . \tag{6.1}$$

Finally, suppose the practitioner is looking for the approximate solution to a linear system (using least squares) but wants a solution x^* that uses as few variables as possible. There are many ways to do this, but one popular (because convex) way to do so is to constrain the ℓ_1 norm of x:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2}\|Ax - b\|^2 \quad \text{subject to} \quad \|x\|_1 \leq \tau \ .$$

All these examples lead us to consider constrained problems:

$$\min_{x \in \mathcal{C}} f(x) \ . \tag{6.2}$$

Because of the constraint, even if f is still smooth, the tools we have used so far no longer apply. Consider the basic one-dimensional problem:

$$\min_{x \in [0,1]} x \ . \tag{6.3}$$

This problem is friendly: the objective is convex, and the constraint set $[0, 1]$ is convex and compact. The minimizer is 0, yet $\nabla f(0) = 1 \neq 0$ and so for constrained convex optimization, the global characterization of minimizers (Proposition 2.7) does not hold.

Fortunately, we can replace some concepts seen so far with generalizations that are very well adapted. First, let us introduce a tool that allows removing the constraints.

6.2 Extended value functions

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

$$\begin{aligned} \iota_{\mathcal{C}} : \mathbb{R}^d &\to \overline{\mathbb{R}} \\ x &\mapsto \begin{cases} 0 \ , & \text{if } x \in \mathcal{C} \ , \\ +\infty \ , & \text{otherwise} \ . \end{cases} \end{aligned} \tag{6.4}$$

Note that this is different from the other indicator function that takes the value 1 on the set and 0 outside[6]. The indicator function conveniently allows transforming all constrained minimization problems into unconstrained ones of course at the price of working with extended value functions:

$$\operatorname*{argmin}_{x \in \mathcal{C}} f(x) = \operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) + \iota_{\mathcal{C}}(x) \ . \tag{6.5}$$

Definition 6.2. *The* domain *of a function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ is:*

$$\operatorname{dom} f \triangleq \{x \in \mathbb{R}^d, f(x) < +\infty\} \ . \tag{6.6}$$

The second tool to overcome non-differentiability is a substitute for the gradient.

6.3 Subdifferential of convex functions

Definition 6.3. *The subdifferential of f at x is the set of slopes of all affine minorants of f that are exact at x:*

$$\partial f(x) = \{u \in \mathbb{R}^d : \forall y \in \mathbb{R}^d, \ f(y) \geq f(x) + \langle u, y - x \rangle\} \ . \tag{6.7}$$

An element of the subdifferential is called a subgradient.

[6] this function is rarely convex

Note that contrary to the gradient, the subdifferential is a set-valued mapping: the subdifferential at one point may contain more than one subgradient. It may also be empty at some point (Exercise 6.1); the set of points where the subdifferential is not empty is called its *domain* (not to be confounded with for extended-value functions). As Exercise 6.7 shows, if f is convex the domain of ∂f contains the interior of the domain of f.

The subdifferential is a generalization of the gradient in the following sense.

Proposition 6.4. *Let f be a convex differentiable function. Then the subdifferential only contains the gradient:*

$$\partial f(x) = \{\nabla f(x)\} \ . \tag{6.8}$$

Note: the converse is true (if the subdifferential reduces to a point, f is differentiable at this point).

The subdifferential allows an elegant characterization of the minimizers of *all* convex functions, even the nondifferentiable ones.

Proposition 6.5 (Fermat's rule)**.** *Let f be convex. Then for all x,*

$$x \in \operatorname{argmin} f \Leftrightarrow 0 \in \partial f(x) \ . \tag{6.9}$$

The proof is 1 line and left to the reader.

Example 6.6. *The subdifferential of the absolute value is:*

$$\partial |\cdot|(x) = \begin{cases} \{x/|x|\}, & \text{if } x \neq 0 \ , \\ [-1,1], & \text{otherwise} \ . \end{cases} \tag{6.10}$$

What is the subdifferential of $\iota_{[0,1]}$?

Proposition 6.7 (Subdifferential of sum)**.** *The subdifferential of the sum contains the sum of subdifferentials:*

$$\partial(f+g) \supset \partial f + \partial g \ , \tag{6.11}$$

(to be understood as the Minkowski sum). Equality does not hold in general, although the counterexamples are pathological (Exercise 6.8).

Remark 6.8 (Subgradient descent)**.** *If you look again with nonsmooth eyes at Proposition 3.2 regarding gradient descent on Lipschitz convex functions, we actually never used the fact that f was differentiable: we only started with $f(x_k) - f(x^*) \leq \langle \nabla f(x_k), x_k - x^* \rangle$. Since a subgradient of f at x_k, by definition, also satisfies this inequality, it means we can replace $\nabla f(x_k)$ by any $g_k \in \partial f(x_k)$ and the proof still holds.*

This algorithm is called subgradient descent.

The last tool we need for nonsmooth optimization is the proximal operator.

6.4 The proximal operator

So far we have brushed away the existence of minimizers, and only assumed they exist. In classical analysis, the celebrated Weierstrass theorem says that a continuous function is bounded and reaches its minimum on any compact. We will provide a slight relaxation of this theorem that handles a slightly weaker property than continuity.

Definition 6.9 (Lower semi-continuity)**.** *The function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ is lower semicontinuous at $x \in \operatorname{dom} f$ if for any sequence (x_k) converging to x,*

$$f(x) \leq \liminf_{k\to\infty} f(x_k) \ . \tag{6.12}$$

Any continuous function is clearly lower semicontinuous.

Remark 6.10. *Lower semicontinuous functions are also referred to as* closed *functions. This is because a function is l.s.c. if and only if its* epigraph

$$\operatorname{epi} f = \{(x,t) : \ f(x) \leq t\} \subset \mathbb{R}^d \times \mathbb{R} \tag{6.13}$$

is closed.

Proposition 6.11 (Existence of minimizers, the direct method of calculus of variations)**.** *Let $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ be a coercive lower semicontinuous function, with nonempty domain. Then f admits at least one minimizer.*

Note: this result is only valid in finite dimension.

Proof. Let x_0 such that $f(x_0) < +\infty$. Since f is coercive, $f(x) \to_{\|x\|\to\infty} +\infty$. Let M such that $\|x\| \geq M \implies f(x) \geq f(x_0)$. Let $\mathcal{B}$ be the ball of center 0, radius M. Since we are in finite dimension, $\mathcal{B}$ is compact.

Let $f^* = \inf_{x\in\mathcal{B}} f(x)$. In general, we could have $f^* = -\infty$ (think $d = 1$, $M = 1$, $f(x) = 1/x$ except $f(0) = 0$). But we'll show that since f is l.s.c., it can't be the case.

Assume $f^* = -\infty$. We can construct a sequence (x_k) in $\mathcal{B}$ such that $\forall k \in \mathbb{N}, f(x_k) \leq -k$. But $\mathcal{B}$ is compact: we can extract a converging subsequence, that we call y_k, converging to $\tilde{x} \in \mathcal{B}$.

By lower semicontinuity of f, $f(\tilde{x}) \leq \liminf_{k\to\infty} f(y_k) = -\infty$ which is not possible since f does not take value $-\infty$.

Hence f^* is finite. Now we do the exact same reasoning to construct (x_k) such that $f(x_k) \leq f^* + 1/k$. Take a converging subsequence, then use lower semicontinuity to show that the limit point $\tilde{x}$ satisfies $f(\tilde{x}) \leq f^*$ and thus $f(\tilde{x}) = f^*$. □

Definition 6.12. *The function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ is said to be (or more rigorously belong to) $\Gamma_0(\mathbb{R}^d)$ if it is:*

- *convex*
- *proper (its domain is not empty)*
- *lower semicontinuous.*

To get familiar with these three notions, determine under which condition of the set $\mathcal{C} \subset \mathbb{R}^d$ does $\iota_{\mathcal{C}}$ belong to $\Gamma_0(\mathbb{R}^d)$?

Definition 6.13 (Proximal operator)**.** *Let $f \in \Gamma_0(\mathbb{R}^d)$. The proximal operator of f evaluated at x is*

$$\operatorname{prox}_f(x) = \operatorname*{argmin}_{y \in \mathbb{R}^d} f(y) + \frac{1}{2}\|x - y\|^2 \ . \tag{6.14}$$

Why is it well-defined (why does the infimum exist? Why is it attained? Why does the argmin contain exactly one point)?

The following is a useful characterization of proximal operators.

Proposition 6.14 (Characterization of proximal operators)**.** *Let $f \in \Gamma_0(\mathbb{R}^d)$. Then $p = \operatorname{prox}_f(x)$ if and only if $x - p \in \partial f(p)$.*

This justifies the frequent formulation $\operatorname{prox}_f = (\mathrm{Id} + \partial f)^{-1}$ (a.k.a. the prox is the resolvent of the subdifferential).

Proof. Apply Fermat's rule. □

Proximal operators may seem new, but without realizing it you know and have used some particular instances: if $\mathcal{C}$ is non-empty, closed and convex, $\operatorname{prox}_{\iota_\mathcal{C}}$ is the (well-defined) projection onto $\mathcal{C}$. For this and other reasons, proximal operators can be thought of as a generalization of projections.

TODO prox of tikhonov. prox of absolute value. Prox of L1.

Example 6.15. *The proximal operator of the absolute value is the soft-thresholding:*

$$\operatorname{prox}_{\gamma|\cdot|}(x) = \begin{cases} x - \lambda, & \text{if} x > \lambda \,, \\ 0, & \text{if} x \in [-\lambda, \lambda] \,, \\ x + \lambda, & \text{if} x < -\lambda \,. \end{cases} \tag{6.15}$$

Proposition 6.16 (Firm non expansivity of proximal operators)**.** *Proximal operators are firmly nonexpansive:*

$$\|x - y\|^2 \leq \langle \operatorname{prox}_f(x) - \operatorname{prox}_f(y), x - y \rangle \ . \tag{6.16}$$

By Cauchy-Schwarz inequality, this immediately implies that proximal operators are non-expansive:

$$\|\operatorname{prox}_f(x) - \operatorname{prox}_f(y)\| \leq \|x - y\| \ . \tag{6.17}$$

Proof. Let $p_x = \operatorname{prox}_f(x)$, $p_y = \operatorname{prox}_f(y)$. The prox characterization for p_x is $x - p_x \in \partial f(p_x)$, so by definition of the subdifferential:

$$f(p_y) \geq f(p_x) + \langle x - p_x, p_y - p_x \rangle \ . \tag{6.18}$$

Do the same for p_y, sum the two inequations to cancel out the f's, and reorder. □

6.5 The proximal point algorithm

TODO: write the algorithm and explanation

By Fermat's rule and the characterization of proxs (Proposition 6.14), x^* is a minimizer of f if and only if it is a fixed point of prox_f: $x^* = \text{prox}_f(x^*)$.

TODO: connect with the fixed-point section. TODO: say that it's a weird algorithm because to minimize f it requires minimizing f + parabola. But it's a powerful analysis tool, as algorithms rewrite as the proximal point algorithm in different spaces (eg Chambolle-Pock, ADMM (?))

6.6 The proximal gradient descent algorithm

TODO write proposition

$$x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k)) \Leftrightarrow \frac{x_k - x_{k+1}}{\gamma} - \nabla f(x_k) \in \partial g(x_{k+1}) \tag{6.19}$$

Hence for all x,

$$g(x_{k+1}) - g(x) \leq \langle \frac{x_k - x_{k+1}}{\gamma} - \nabla f(x_k), x_{k+1} - x \rangle \tag{6.20}$$

and by the descent lemma:

$$f(x_{k+1}) - f(x_k) \leq \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \ . \tag{6.21}$$

By using $x = x_k$ and summing, we get a first nice result: the proximal gradient descent method is a descent method for $0 < \gamma < 2/L$:

$$F(x_{k+1}) - F(x_k) \leq \left(\frac{L}{2} - \frac{1}{\gamma}\right) \|x_{k+1} - x_k\|^2 \ . \tag{6.22}$$

Back to our general proof, we sum Equations (6.20) and (6.21):

$$F(x) - F(x_{k+1}) - f(x) + f(x_k) \geq \frac{1}{\gamma}\langle x_{k+1} - x_k, x_{k+1} - x \rangle - \frac{L}{2}\|x_{k+1} - x_k\|^2 - \langle \nabla f(x_k), x_k - x \rangle$$

$$F(x) - F(x_{k+1}) \geq \frac{1}{\gamma}\langle x_{k+1} - x_k, x_{k+1} - x \rangle - \frac{L}{2}\|x_{k+1} - x_k\|^2 + D_f(x, x_k) \tag{6.23}$$

$$\geq \frac{L}{2}\|x - x_{k+1}\|^2 - \frac{L}{2}\|x_k - x\|^2 \tag{6.24}$$

TODO: general γ? Apply in $x = x^*$, sum and use the fact that suboptimality decreases at each iteration to obtain:

$$k(F(x_k) - F(x^*)) \leq \sum_{t=1}^{k} F(x_t) - F(x^*) \leq \frac{L}{2}\|x_0 - x^*\|^2 \ . \tag{6.25}$$

6.7 Exercises

6.7.1 Subdifferential

Exercise 6.1. ☕ *Provide an example of convex function which has an empty subdifferential at some point of its domain.*

Exercise 6.2. ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be separable: $f(x) = \sum_1^d f_i(x_i)$ where the f_i's are functions of the real variable.*
Show that $\partial f(x) = \partial f_1(x_1) \times \ldots \times \partial f_d(x_d)$.
Compute the subdifferential of the ℓ_1-norm.

Exercise 6.3. ☕ *Show that the subdifferential of a function at a point writes as an intersection of half-spaces. Show that it is convex and closed.*

Exercise 6.4. ☕ *Show that the subdifferential of ι_C at $x \in C$ is equal to the* normal cone *of C at x, that is:*

$$\mathcal{N}_C(x) = \{u : \forall z \in C, \langle u, z - x\rangle \leq 0\}$$

Exercise 6.5. ☕☕ *Compute the subdifferential of the Euclidean norm at any point in $\mathbb{R}^d$.*

Exercise 6.6 (Exercise 6.10 revisited)**.** ☕☕ *Show that the first order optimality condition for differentiable convex constrained optimization rewrites:*

$$x^* \in \operatorname*{argmin}_C f(x) \Leftrightarrow -\nabla f(x^*) \in \mathcal{N}_C(x^*) \ .$$

Exercise 6.7 (Non emptiness of subdifferential)**.** ☕☕☕ *Let $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ be convex. Show that if $x \in \operatorname{int}(\operatorname{dom} f)$, $\partial f(x)$ is non empty and compact.*
Show that if x lies on the boundary of $\operatorname{dom} f$*, $\partial f(x)$ is either empty or unbounded.*

Exercise 6.8. ☕ *This exercise is in dimension 2. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two closed balls of strictly positive radius, that share a single point. Show that $\partial(\iota_{\mathcal{C}_1} + \iota_{\mathcal{C}_2})$ is a strict superset of $\partial\iota_{\mathcal{C}_1} + \iota_{\mathcal{C}_2}$ at this point.*

6.7.2 Constrained optimization

Exercise 6.9. ☕ *Show that the indicator function ι_C is convex (resp. lower semicontinuous, resp. proper) if C is convex (resp. closer, resp. nonempty).*

Exercise 6.10 (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 6.11. *Let C be a nonempty closed convex set. Show that for all $x \in C, y \in \mathbb{R}^d$,*

$$\langle y - \Pi_C(y), x - \Pi_C(y) \rangle \leq 0 \ . \tag{6.26}$$

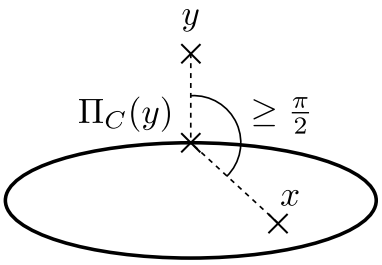

Figure 5: Illustration of (6.26)

Exercise 6.12. ☕ *Let C be a nonempty 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).

Exercise 6.13. ☕ *Let C be a nonempty closed convex set. Show that the* overprojection, *$2\Pi_C - \mathrm{Id}$, is nonexpansive (i.e. it has Lipschitz constant 1).*

7 Duality

TODO: this is not a good motivation. Duality is everywhere! One motivation (but not the single reason for this chapter to exist!) is the following question: when do we stop iterative algorithms? Usually we'd settle for $f(x_k) - f(x^*)$ less than some tolerance, but $f(x^*)$ is unknown.

7.1 The Fenchel transform

A fundamental tool in convex analysis is the Fenchel transform (sometimes called the Fenchel-Legendre or Legendre transform), which is, in many ways, akin to the Fourier transform in signal processing.

Definition 7.1 (Fenchel conjugate)**.** *Let $f : \mathbb{R}^d \to \overline{\mathbb{R}}$. The Fenchel conjugate (or transform) of f is:*

$$f^*(u) = \sup_{x \in \mathbb{R}^d} \langle u, x \rangle - f(x) \ . \tag{7.1}$$

It is always convex, even if f is not (Exercise 2.2). To get familiar with it, compute the Fenchel transforms of $\| \cdot \|^2/2a$ ($a > 0$), $\| \cdot \|$ and $x \mapsto \sup_{y \in [-1,1]} x$.

Geometrical interpretation of the Fenchel transform $f^*(u) \leq \alpha$ is equivalent to

$$\forall x \in \mathbb{R}^d, \quad \langle x, u \rangle - \alpha \leq f(x) \ , \tag{7.2}$$

meaning that $-\alpha$ is the intercept of a global affine minorant of f of slope u. $-f^*(u)$ is thus the biggest possible intercept of such minorants – beware, there may not exist such minorants ($f^*(u) = +\infty$).

Proposition 7.2 (Involution on Γ_0)**.** *If $f \in \Gamma_0(\mathbb{R}^d)$, $f^{**} = f$.*

The proof technique is (to my knowledge) not really used in other optimization proofs, but it uses a beautiful argument.

Proof. By definition, $f^{**}(x) = \sup_u \langle u, x \rangle - f^*(u)$. Let us write $\phi_u(x) = \langle u, x \rangle - f^*(u)$. What are these functions? They are affine, and they globally lower bound f by Fenchel-Young inequality: $\forall x \in \mathbb{R}^d, \phi_u(x) \leq f(x)$. So

$$f^{**}(x) = \sup_u \phi_u(x) \leq \sup_{\substack{a \text{ affine} \\ a \leq f}} a(x) \ . \tag{7.3}$$

In fact, these two supremums are equal: take an affine function globally lower bounding f, with slope u. It must therefore be equal to $\langle x, u \rangle - c$ for some $c \in \mathbb{R}$. For all $x \in \mathbb{R}^d$, $-c + \langle x, u \rangle \leq f(x)$, so $c \geq \sup_x \langle x, u \rangle - f(x) = f^*(u)$. Therefore $\langle \cdot, u \rangle - c \leq \phi_u$.

Hence, the supremum over all affine functions lower bounding f is equal to the supremum over the "extremal" functions ϕ_u only:

$$f^{**}(x) = \sup_{\substack{a \text{ affine} \\ a \leq f}} a(x) \ . \tag{7.4}$$

This shows that $f^{**} \leq f$ all the time (no requirement on convexity).

Now we show equality when f is convex. Take $y \in \mathbb{R}, x \in \mathbb{R}^d$ such that $y < f(x)$ i.e. (y, x) not in the epigraph of f. The latter is convex, so by the celebrated Hahn-Banach theorem, we can find a hyperplane separating the two. This hyperplane corresponds to an affine function (TODO what if it is vertical?) that is above y at x, at globally lower bounds f. So $f^{**}(x) > y$. This concludes the proof.

TODO this also means that f^{**} is the best pointwise: if $f^{**}(x) < g(x)$ with g convex and $g \leq f$, then we can apply Hahn-Banach too and construct an affine a that is above $(x, f^{**}(x))$ and globally below f, which is impossible.

□

Proposition 7.3 (Fenchel-Young inequality)**.** *Let $f : \mathbb{R}^d \to \overline{\mathbb{R}}$. Then for all $x, u \in \mathbb{R}^d$,*

$$f(x) + f^*(u) \geq \langle x, u \rangle \ . \tag{7.5}$$

Equality holds if and only if $u \in \partial f(x)$.

Proof. The first part is trivial by definition of f^*. Then,

$$\begin{aligned} f^*(u) = \langle x, u \rangle - f(x) &\Leftrightarrow x \in \operatorname{argmin} f - \langle \cdot, u \rangle \\ &\Leftrightarrow 0 \in \partial(f - \langle \cdot, u \rangle)(x) \\ &\Leftrightarrow u \in \partial f(x) \ . \end{aligned}$$

□

Proposition 7.4. *If $f \in \Gamma_0(\mathbb{R}^d)$, $\partial f^* = (\partial f)^{-1}$, in the sense that:*

$$u \in \partial f(x) \Leftrightarrow x \in \partial f^*(u) \ . \tag{7.6}$$

Proof. Use then Fenchel-Young equality for f and then use $f^{**} = f$. □

Proposition 7.5 (Moreau decomposition formula)**.** *Let $f \in \Gamma_0(\mathbb{R}^d)$. Then*

$$\operatorname{prox}_{\gamma f} + \gamma \operatorname{prox}_{\gamma^{-1} f^*}(\cdot/\gamma) = \mathrm{Id} \tag{7.7}$$

Proof.

$$p = \operatorname{prox}_{\gamma f}(x) \Leftrightarrow \frac{x - p}{\gamma} \in \partial f(p) \tag{7.8}$$

$$\Leftrightarrow p \in \partial f^* \left(\frac{x - p}{\gamma} \right) \tag{7.9}$$

$$\Leftrightarrow \frac{x}{\gamma} - \frac{x - p}{\gamma} \in \partial \frac{1}{\gamma} f^* \left(\frac{x - p}{\gamma} \right) \tag{7.10}$$

$$\Leftrightarrow \frac{x - p}{\gamma} = \operatorname{prox}_{f^*/\gamma} \left(\frac{x}{\gamma} \right) \ . \tag{7.11}$$

□

Proposition 7.6. *f is μ-strongly convex iff f^* is $\frac{1}{\mu}$-smooth.*

Definition 7.7. *Moreau envelope*

$$M_f^\lambda(x) = \min_x \frac{1}{2\lambda}\|y - x\|^2 + f(y) \ . \tag{7.12}$$

TODO

Proposition 7.8.
- *for a γ_0 f, its Moreau envelope is convex*
- *the Moreau envelope is differentiable and its gradient is a prox*

7.2 Fenchel-Rockafellar duality

Many problems in Machine Learning have the following form:

$$\min_{x\in\mathbb{R}^d} f(Ax) + g(x) = P(x) \ , \tag{7.13}$$

with $A \in \mathbb{R}^{n\times d}$, $f \in \Gamma_0(\mathbb{R}^n)$ and $g \in \Gamma_0(\mathbb{R}^d)$. In this section, we will study the link between such a problem, called the primal, and a closely related one called its *dual problem*.

Definition 7.9. *The Fenchel-Rockafellar dual problem of Problem (13) is:*

$$\min_{y\in\mathbb{R}^n} f^*(y) + g^*(-A^\top y) = D(y) \ . \tag{7.14}$$

It is often alternatively written as a concave maximization problem

$$\max_{y\in\mathbb{R}^n} -f^*(y) - g^*(-A^\top y) \ . \tag{7.15}$$

Where does this problem come from? It can be obtained by Lagrangian duality (Section 7.4):

$$\begin{aligned}
\inf_{x\in\mathbb{R}^d} f(Ax) + g(x) &= \inf_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} f(z) + g(x) \quad \text{s.t. } Ax = z && (7.16)\\
&= \inf_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} f(z) + g(x) + \iota_{\{z\}}(Ax) && (7.17)\\
&= \inf_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} f(z) + g(x) + \sup_{y\in\mathbb{R}^n} \langle y, Ax - z\rangle && (7.18)\\
&= \inf_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} \sup_{y\in\mathbb{R}^n} f(z) + g(x) + \langle y, Ax - z\rangle && (7.19)
\end{aligned}$$

Let's swap inf and sup. By doing so, we lose the equivalence (sup inf $\leq$ inf sup, without equality in general). We have:

$$\begin{aligned}
&\sup_{y\in\mathbb{R}^n} \inf_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} f(z) + g(x) + \langle y, Ax - z\rangle && (7.20)\\
&= \sup_{y\in\mathbb{R}^n} \inf_{x\in\mathbb{R}^d} g(x) + \langle y, Ax\rangle + \inf_{z\in\mathbb{R}^n} f(z) - \langle y, z\rangle && (7.21)\\
&= \sup_{y\in\mathbb{R}^n} - \sup_{x\in\mathbb{R}^d} -g(x) + \langle -A^\top y, x\rangle - \sup_{z\in\mathbb{R}^n} -f(z) + \langle y, z\rangle && (7.22)\\
&= \sup_{y\in\mathbb{R}^n} -g^*(-A^\top y) - f^*(y) \ , && (7.23)
\end{aligned}$$

that is our dual problem.

Proposition 7.10 (Weak duality)**.** *For all $x, y \in \mathbb{R}^d \times \mathbb{R}^n$, $P(x) + D(y) \geq 0$.*

Proof. Write the sum, group f with f^* and g with g^*, then use the Fenchel-Young inequality. □

Proposition 7.11. *Let $x^*, y^* \in \mathbb{R}^d \times \mathbb{R}^n$. The following conditions are equivalent:*

1. $x^* \in \operatorname{argmin}_x P(x)$, $y^* \in \operatorname{argmin}_y D(y)$, $\inf P + \inf D = 0$.
2. $P(x^*) + D(y^*) = 0$
3. $f(Ax^*) + f^*(y^*) = \langle Ax^*, y^* \rangle$ *and* $g(x^*) + g^*(-A^\top y^*) = -\langle x^*, A^\top y^* \rangle$,
4. $-A^\top y^* \in \partial g(x^*)$ *and* $y^* \in \partial f(Ax^*)$
5. $x^* \in \partial g^*(-A^\top y^*)$ *and* $Ax^* \in \partial f^*(y^*)$

Note that 4 rewrites $0 \in A^* \partial f(Ax^*) + \partial g(x^*)$, and similarly 5 rewrites $0 \in \partial f^*(y^*) - A \partial g^*(-A^\top y^*)$.

Proof. 1 same as 2 is OK.
2 same as 3: Do the same as in the proof of Proposition 7.10, and use that the sum of two positive terms is 0, if and only if both terms are 0.
3 same as 4: equality in Fenchel-Young case
4 same as 5: Proposition 7.4 □

Definition 7.12 (Saddle point of the Lagrangian)**.** *The pair $x^*, y^* \in \mathbb{R}^d \times \mathbb{R}^n$ is said to be a saddle-point of the Lagrangian if:*

$$\forall x, y \in \mathbb{R}^d \times \mathbb{R}^n,\ \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x, y^*) \ . \tag{7.24}$$

Proposition 7.13. *If $x^*, y^* \in \mathbb{R}^d \times \mathbb{R}^n$ is a saddle-point of the Lagrangian, then x^* solves the primal and y^* solves the dual.*

Proof. Write the two inequalities stating that the pair is a saddle point, make subgradients appear, and use Proposition 7.11. □

Definition 7.14 (Qualification condition)**.** *The following inclusion is called a* qualification condition*:*

$$0 \in \operatorname{int}(\operatorname{dom} f - A \operatorname{dom} g) \ . \tag{7.25}$$

Where does it come from? It can be seen as a stronger version of "the primal objective P is proper":

$$\begin{aligned}
\operatorname{dom} P \neq \emptyset &\Leftrightarrow \exists x \in \mathbb{R}^d, f(Ax) + g(x) < +\infty \\
&\Leftrightarrow \exists x \in \mathbb{R}^d, f(Ax) < +\infty,\ g(x) < +\infty \\
&\Leftrightarrow \exists x \in \mathbb{R}^d, y \in \mathbb{R}^n,\ y = Ax,\ f(y) < +\infty,\ g(x) < +\infty \\
&\Leftrightarrow \exists x \in \mathbb{R}^d, y \in \mathbb{R}^n,\ y = Ax,\ y \in \operatorname{dom} f,\ x \in \operatorname{dom} g \\
&\Leftrightarrow \exists x \in \operatorname{dom} g, y \in \operatorname{dom} f,\ 0 = y - Ax \\
&\Leftrightarrow 0 \in \operatorname{dom} f - A \operatorname{dom} g \ .
\end{aligned} \tag{7.26}$$

Remark 7.15. *When does QC hold? One often used property is that it holds when f is continuous at Ax for some $x \in \operatorname{dom} g$. Indeed, by continuity, this means that there exists some δ such that with B_δ the ball of center 0 and radius δ, $Ax + B_\delta \subset \operatorname{dom} f$. Then $B_\delta \subset \operatorname{dom} -f - Ax \subset \operatorname{dom} f - A \operatorname{dom} g$ and so QC holds.*

Proposition 7.16 (Strong duality)**.** *Suppose that QC holds. Then the dual has a solution and* $\inf P + \inf D = 0$.

Proof. The case where $\inf P = -\infty$ is easy to address. Let us now consider the case $\inf P > -\infty$. Define the function:

$$h : y \mapsto \inf_x f(Ax + y) + g(x) \ . \tag{7.27}$$

A few observations about this function: it does not take value $-\infty$; it is convex, its value at 0 is $h(0) = \inf P(x)$ and its domain is $\operatorname{dom} f - A \operatorname{dom} g$, so QC translates into $0 \in \operatorname{int} \operatorname{dom} h$.

Proof: h does not take value $-\infty$: observe that by QC f is finite in a neighborhood of 0; hence $h(-\epsilon y)$ is finite for ϵ small enough. Then by convexity, since $0 = -\frac{1}{1+\epsilon}\epsilon y + (1 - \frac{1}{1+\epsilon})y$, $h(0) \leq -\frac{1}{1+\epsilon}(\epsilon y) + (1 - \frac{1}{1+\epsilon})h(y)$ and so $h(y)$ cannot be $-\infty$.

Since $0 \in \operatorname{int} \operatorname{dom} h$ and h is convex, the subgradient of h at 0 is non-empty. Let $v \in \partial h(0)$. For all $y \in \mathbb{R}^d$, $h(y) \geq h(0) + \langle v, y \rangle$. Reordering, using $h(0) = \inf P$ and the definition of h as an infimum, we get that for all x, y:

$$\begin{aligned} \inf P &\leq g(x) + f(Ax + y) - \langle v, y \rangle & (7.28) \\ &\leq g(x) + f(Ax + y) - \langle x, -A^* v \rangle - \langle v, Ax + y \rangle \ . & (7.29) \end{aligned}$$

Taking the infimum of the right-hand side with respect to y, then with respect to x yields:

$$\inf P \leq -f^*(v) - g^*(-A^* v) \ , \tag{7.30}$$

so the duality gap is 0 and v is a solution of the dual. □

7.3 Primal-dual algorithms

In this section, we introduce more algorithms to deal with composite problems, based on duality/splitting.

Chambolle-Pock algorithm The Chambolle-Pock algorithm solves problems of the form:

$$\min_{x \in \mathbb{R}^d} f(Ax) + g(x) \,, \tag{7.31}$$

where $f \in \Gamma_0(\mathbb{R}^n)$, $g \in \Gamma_0(\mathbb{R}^d)$, but both need not be smooth. It requires access to their proximal operators. Let σ and τ be positive stepsizes such that $\sigma\tau\|A\|_2^2 < 1$. Let $\theta \in]0, 1]$. The Chambolle-Pock iterations write:

$$\begin{cases} x_{k+1} = \operatorname{prox}_{\tau g}(x_k - \tau A^\top \bar{y}_k) \\ y_{k+1} = \operatorname{prox}_{\sigma f^*}(y_k + \sigma A[(1+\theta)x_{k+1} - \theta x_k]) \end{cases} \tag{7.32}$$

Note: The interpolation parameter θ is most often taken equal to 1. The interpolation step can be performed on y instead of x, which yields a different version of the algorithm.

Proposition 7.17. *For $\theta = 1$, the Chambolle-Pock algorithm is an instance of the preconditioned proximal point algorithm, in the space $\mathbb{R}^d \times \mathbb{R}^n$, to find a 0 of a suitable operator.*

Proof. The optimality conditions from the definitions of x_{k+1} and y_{k+1} rewrite:

$$\frac{1}{\tau}x_k - \frac{1}{\tau}x_{k+1} - A^\top y_k \in \partial g(x_{k+1}) \tag{7.33}$$

$$\frac{1}{\sigma}y_k - \frac{1}{\sigma}y_{k+1} + 2Ax_{k+1} - Ax_k \in \partial f^*(y_{k+1}) \tag{7.34}$$

In turn, this writes:

$$\begin{pmatrix} \frac{1}{\tau}\mathrm{Id} & -A^\top \\ -A & \frac{1}{\sigma}\mathrm{Id} \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \in \left[\begin{pmatrix} \frac{1}{\tau}\mathrm{Id} & -A^\top \\ -A & \frac{1}{\sigma}\mathrm{Id} \end{pmatrix} + \begin{pmatrix} \partial g & A^\top \\ A^\top & \partial f^* \end{pmatrix} \right] \begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} \tag{7.35}$$

Thus for $v_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$, $M = \begin{pmatrix} \frac{1}{\tau}\mathrm{Id} & -A^\top \\ -A & \frac{1}{\sigma}\mathrm{Id} \end{pmatrix}$ and $T = \begin{pmatrix} \partial g & A^\top \\ A^\top & \partial f^* \end{pmatrix}$ this writes: $v_{k+1} = (\mathrm{Id} + M^{-1}T)^{-1}v_k$, which is the proximal point with different metric. □

Vu-Condat algorithm TODO refs Vu and Condat independently proposed an algorithm that handles an additional smooth term h in the objective:

$$\min_{x \in \mathbb{R}^d} f(Ax) + g(x) + h(x)\,, \tag{7.36}$$

where $f \in \Gamma_0(\mathbb{R}^n)$, $g \in \Gamma_0(\mathbb{R}^d)$, $h \in \Gamma_0(\mathbb{R}^d)$, f and g's proximal operators are known, and h is L_h-smooth[7].

For stepsizes τ and σ satisfying[8]:

$$\frac{\tau\sigma}{\|A\|^2} + \frac{\tau L_h}{2} \leq 1\,, \tag{7.37}$$

the Condat-Vu algorithm reads:

$$\begin{cases} \bar{y}_k = y_k + \theta(y_k - y_{k-1}) \\ x_{k+1} = \mathrm{prox}_{\tau g}(x_k - \tau\nabla h(x_k) - \tau A^\top \bar{y}_k)\,, \\ y_{k+1} = \mathrm{prox}_{\sigma f^*}(y_k + \sigma A x_k)\,. \end{cases} \tag{7.38}$$

Note: it can even handle a fourth smooth term, so that the saddle point formulation becomes TODO

[7] for completeness let us mention that it can even handle the slightly more complex case where f is replaced by $f \,\square\, F$ with F strongly convex, leading to a second smooth term, F^*, in the saddle point formulation.

[8] this condition is in the case $\beta > 0$; otherwise inequality must be strict

ADMM The Alternating Direction Method of Multipliers (ADMM method) is designed to solve problems of the form:

$$\min_{x\in\mathbb{R}^d, z\in\mathbb{R}^n} f(z)+g(x) \quad \text{s.t.} \quad Ax+Bz=c \ . \tag{7.39}$$

For simplicity and to match the setup of Chambolle-Pock we take $B=-\mathrm{Id}$ and $c=0$.

The augmented Lagrangian of parameter ρ is:

$$\mathcal{L}^\rho(x,y,z)=f(z)+g(x)+\langle y, Ax-z\rangle+\frac{\rho}{2}\|Ax-z\|^2 . \tag{7.40}$$

Note that some formulations use a *scaled* dual variable $y'=\rho y$, and rewrite the last two terms as $\frac{\rho}{2}\|Ax-z+y'\|-\frac{\rho}{2}\|y'\|^2$.

The ADMM algorithm reads:

$$\begin{cases} x_{k+1}\in \operatorname{argmin}_x \mathcal{L}^\rho(x,y_k,z_k) \\ z_{k+1}\in \operatorname{argmin}_y \mathcal{L}^\rho(x_{k+1},y_k,z) \\ y_{k+1}=y_k+r(Ax_{k+1}-z_{k+1}) \end{cases} \tag{7.41}$$

Douglas-Rachford algorithm

$$x_{k+1}=x_k+\operatorname{prox}_g(2\operatorname{prox}_f(x_k)-x_k)-\operatorname{prox}_f(x_k) \tag{7.42}$$

Application of the operator $\frac{1}{2}\mathrm{Id}+\frac{1}{2}(2\operatorname{prox}_g-\mathrm{Id})(2\operatorname{prox}_f-\mathrm{Id})$.

TODO ADMM is the Douglas Rachford algorithm.

7.4 Lagrangian duality

Fenchel-Rockafellar duality (Section 7.2) is a particular case of the more versatile *Lagrangian* duality. The latter is concerned with convex problems with affine constraints, that is problems of the form

$$\min_{x\in\mathcal{D}} f_0(x) \quad \text{s.t.} \quad f_i(x)\le 0 \quad \forall i\in[m], \quad h_i(x)=0 \quad \forall i\in[p] \tag{7.43}$$

where $\mathcal{D}\subset\mathbb{R}^d$ is convex, the functions $f_0,\dots,f_m$ are convex and $h_1,\dots,h_p$ are affine.

Definition 7.18 (Lagrangian, dual function, Lagrangian dual)**.** *The Lagrangian associated with Problem (43) is the function*

$$\begin{aligned} \mathcal{L}:\mathcal{D}\times\mathbb{R}^n_+\times\mathbb{R}^p &\to \mathbb{R} \\ (x,\lambda,\mu) &\mapsto f_0(x)+\sum_{i=1}^n \lambda_i f_i(x)+\sum_{i=1}^p \mu_i h_i(x) \ ; \end{aligned} \tag{7.44}$$

Note that the variables λ_i associated with the inequalities are positive; the rationale is that, as soon as x does not satisfy one constraint, maximizing in λ_i or μ_i will make the Lagrangian go to $+\infty$. Also note that for any x feasible for Problem (43), the Lagrangian value for any λ,μ upper bounds $f_0(x)$.

Finally, the dual function is:

$$\begin{aligned} g : \mathbb{R}_+^m \times \mathbb{R}^p &\to \mathbb{R} \\ (\lambda, \mu) &\mapsto \min_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \mu) \ . \end{aligned} \tag{7.45}$$

and the Lagrange dual problem of Problem (43) is:

$$\max_{\lambda \in \mathbb{R}_+^m, \mu \in \mathbb{R}^p} g(\lambda, \mu) \ . \tag{7.46}$$

Proposition 7.19 (Weak duality)**.** *Under the convexity and linearity assumptions on the f_i's and h_i's respectively, weak duality holds: the dual optimal value is smaller than the primal optimal value.*

Proof. Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, let $a_0, b \in \mathbb{R}^n \times \mathbb{R}^m$. Then:

$$\inf_{a \in \mathbb{R}^n} f(a, b) \leq f(a_0, b) \tag{7.47}$$

$$\sup_{b \in \mathbb{R}^m} \inf_{a \in \mathbb{R}^n} f(a, b) \leq \sup_{b \in \mathbb{R}^m} f(a_0, b) \tag{7.48}$$

The left-hand side is a constant, so the infimum of the right-hand side with respect to a_0 must be greater:

$$\sup_{b \in \mathbb{R}^m} \inf_{a \in \mathbb{R}^n} f(a, b) \leq \inf_{a \in \mathbb{R}^n} \sup_{b \in \mathbb{R}^m} f(a, b) \ . \tag{7.49}$$

So the supremum of the infimums is smaller than the inf of the sups, which concludes. □

Under some conditions, however, we have strong duality: the primal and dual values are equal. There are many such conditions; one of the most popular ones is Slater's condition.

Proposition 7.20 (Slater's condition)**.** *If Problem (43) satisfies Slater's condition, meaning that is is* strictly *feasible:*

$$\exists x \in \mathcal{D}, \ f_i(x) < 0 \, \forall i \in [m], \quad h_i(x) = 0 \, \forall i \in [p] \ , \tag{7.50}$$

then strongly duality holds: the primal and dual optimal values are equal.

7.5 Exercises

Exercise 7.1. ☕ *Show that the Fenchel transform is order-reversing: if $f \leq g$, $g^* \leq f^*$.*

Exercise 7.2. ☕☕ *Compute the Fenchel transform of the ℓ_p-norm for $p \in [1, +\infty]$.*

Exercise 7.3. ☕☕ *Show that $x \mapsto \frac{1}{2}\|x\|^2$ is a fixed point of the Fenchel transform. Show that it is the only fixed point of the Fenchel transform.*

Exercise 7.4. ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be separable: $f(x) = \sum_1^d f_i(x_i)$ where the f_i's are functions of the real variable.*
Show that $f^*(u) = (f_i^*(u_i))_{i\in[d]}$.

Exercise 7.5 ("Lipschitzing trick")**.** ☕☕ *Let $f \in \Gamma_0(\mathbb{R}^d)$.*
Show that f if M-Lipschitz if and only if the domain of f^* is included in the Euclidean ball of center 0 and radius M.

Exercise 7.6 (Convolution smoothing)**.** ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex.*
Show that $f \square \frac{L}{2}\|\cdot\|^2$ is L-smooth.

Exercise 7.7 (Convolution Lipschitzing)**.** ☕ *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex.*
Show that $f \square L\|\cdot\|$ is L-Lipschitz.

Exercise 7.8 (A case of strict inclusion in subdifferential of sums)**.** ☕ *or* ☕☕ *depending on whether or not you know the result of Exercise 6.5*
In $\mathbb{R}^2$, let D_1 be the closed disk of center $(-1, 0)$ and radius 1, and D_2 be the closed disk of center $(0, 1)$ and radius 1. Compute $\partial(\iota_{D_1} + \iota_{D_2})(0, 0)$ and compare to $\partial\iota_{D_1}(0, 0) + \partial\iota_{D_2}(0, 0)$. Does the qualification condition hold for the two functions under consideration?

Exercise 7.9 (A case without strong duality)**.** ☕ *Let*

$$f : \mathbb{R}^d \to \mathbb{R}$$
$$x \mapsto \begin{cases} x\log x - x & \text{if } x > 0 \ , \\ 0 & \text{if } x > 0 \ , \\ +\infty & \text{otherwise} \ . \end{cases} \tag{7.51}$$

Show that $\min_{x\in\mathbb{R}^d} f(x) + f(-x)$ has solutions. What about the dual problem?

Exercise 7.10. *Let $\mathcal{D} = \mathbb{R} \times \mathbb{R}_+^*$, and $f : \mathcal{D} \to \mathbb{R}$ defined by $f(x, y) = e^{-x}$.*
Show that the problem

$$\min_{x,y\in\mathcal{D}} f(x, y) \quad s.t. \quad \frac{x^2}{y} \leq 0 \tag{7.52}$$

is convex. Compute its optimal value. Compute the Lagrangian dual and its optimal value. Does Slater's condition hold for this problem?

8 Convergence through fixed-point iterations

References: 40 pages "gentle and self-contained introduction and tutorial" by Ryu and Boyd (2016), book by Ryu and Yin (2022). See also the very extensive catalog of applications in Combettes and Pesquet (2021).

Splitting methods can be analyzed in an elegant and unified way through the lens of (monotone) operators and fixed point iterations.

8.1 Picard iterations

Let us start by citing the best-known result in fixed point theory, called the Banach (or Banach-Cacciopoli or *contraction mapping*) theorem. It applies to *contractions*.

Definition 8.1 (Contraction)**.** *For a normed space*[9] X*, a* contraction *or* contractive mapping *is an operator* $T : X \to X$*, which has Lipschitz constant strictly less than 1, i.e.*

$$\exists q \in [0, 1[, \forall (x, y) \in X^2, \|T(x) - T(y)\| \leq q\|x - y\| . \tag{8.1}$$

Theorem 8.2 (Banach fixed-point)**.** *Let* X *be a complete space and* T *a contractive mapping. Then* T *admits exactly one fixed point* x^**, i.e. such that* $T(x^*) = x^*$*. For any* $x_0 \in X$*, the sequence defined by* $x_{k+1} = T(x_k)$ *converges to* x^**, and the convergence is linear.*

Proof. Show that $T(x_k)$ is Cauchy, thus converges (X is complete), and the limit must be a fixed point. This shows the fixed point is unique, which one can also see using two fixed points a and b: $\|a - b\| = \|T(a) - T(b)\| \leq q\|a - b\|$. □

Remark 8.3. *The iterations* $x_{k+1} = T(x_k)$ *are called* Picard iterations.

If the Lipschitz constant of T is 1 instead of being strictly less than 1 (such T is called *nonexpansive*) we cannot say anything (neither about eventual fixed points, nor about convergence of Picard iterations). Two good examples to keep in mind for possible behaviors are $-\mathrm{Id}$ (a fixed point exists, Picard iterations do not converge) and translations $T(x) = x + b$ (no fixed point, no convergence).

It turns out that it is possible to refine this division: there is a subclass of nonexpansive operators, larger than contractions, for which Picard iterations converge[10], as shown in Theorem 8.13.

Definition 8.4 (α-averaged operators)**.** *For* $\alpha \in]0, 1]$*,* T *is* α*-averaged if and only if there exists a nonexpansive operator* R *such that* $T = (1 - \alpha)\mathrm{Id} + \alpha R$*.*
It is easy to see that such operators are nonexpansive.

Remark 8.5. *Definition 8.4 requires to be in a vector space to have addition and scalar multiplication, while Theorem 8.2 works in a metric space. There seems to be recent work extending the definition of* α*-averaged operators to more generic spaces (Berdellima, 2020).*

[9] the definition can straightforwardly be applied to the more general setting of metric spaces
[10] provided a fixed point exists

Proposition 8.6 (Characterization of α-averaged operators)**.** *Let $\alpha \in]0,1]$. The following are equivalent:*

(i) T *is α-averaged.*
(ii) $(1-\alpha^{-1})\mathrm{Id} + \alpha^{-1}T$ *is nonexpansive.*
(iii) $\|T(x)-T(y)\|^2 \leq \|x-y\|^2 - (\alpha^{-1}-1)\|(\mathrm{Id}-T)(x)-(\mathrm{Id}-T)(y)\|^2$.
(iv) $\|T(x)-T(y)\|^2 + (1-2\alpha)\|x-y\|^2 \leq 2(1-\alpha)\langle x-y, T(x)-T(y)\rangle$.

Proof. For $(i) \Leftrightarrow (ii)$: just use the definition.

For $(ii) \Leftrightarrow (iii)$, use the famous equality from Exercise 2.13 ($\alpha^{-1} > 1$ here, but it still holds):

$$\|\lambda a + (1-\lambda b)\|^2 = \lambda\|a\|^2 + (1-\lambda)\|b\|^2 - \frac{1}{2}\lambda(1-\lambda)\|a-b\|^2 \tag{8.2}$$

to get

$$\begin{aligned}\|x-y\|^2 &\geq \|(1-\alpha^{-1})(x-y) + \alpha^{-1}(T(x)-T(y))\|^2 &(8.3)\\ &= (1-\alpha^{-1})\|x-y\|^2 \\ &\quad + \alpha^{-1}\|T(x)-T(y)\|^2 - \tfrac{1}{2}\alpha^{-1}(1-\alpha^{-1})\|(\mathrm{Id}-T)(x)-(\mathrm{Id}-T)(y)\|^2 &(8.4)\end{aligned}$$

Simplify the $\|x-y\|^2$ on both sides and multiply by α.

For $(iii) \Leftrightarrow (iv)$ just develop the last term in (iii). □

An interesting property of the α-averaged operators is that they are stable by composition – though to be precise, one ends up with a different α for the composition.

Proposition 8.7. *The composition of an α_1-averaged and an α_2-averaged operator is $\frac{\alpha_1+\alpha_2-2\alpha_1\alpha_2}{1-\alpha_1\alpha_2}$ averaged.*

The proof is computational and can be found in Ogura and Yamada (2002, Thm 3(b)).

A particularly interesting case is $\frac{1}{2}$-averaged operators, that have their own name.

Definition 8.8. *A $\frac{1}{2}$-averaged operator is called* firmly nonexpansive.

Proposition 8.9 (Characterization of firmly nonexpansive operators)**.** *The following are equivalent:*

1. T *is firmly nonexpansive*
2. $\|T(x)-T(y)\|^2 \leq \|x-y\|^2 - \|(\mathrm{Id}-T)x-(\mathrm{Id}-T)y\|^2$
3. $\|T(x)-T(y)\|^2 \leq \langle T(x)-T(y), x-y\rangle$

Property T is called "cocoercivity" of T, and β-cocoercivity of T means that the equality is satisfied with a positive scalar β multiplying the norm on the left-hand side.

Example 8.10.
1. *Projections and proximal operators (of convex functions) are firmly nonexpansive.*
2. *If f is L-smooth, $\frac{1}{L}\nabla f$ is firmly nonexpansive by the Baillon-Haddad theorem.*
3. *If T is firmly nonexpansive, so is* $\mathrm{Id} - T$.

TODO cite it for weak convergence to be precise?

Lemma 8.11 (Opial lemma)**.** *Let F a non empty subset and (x_k) a sequence such that:*

(i) $\|x_k - y\|$ *converges for all* $y \in F$.
(ii) Every cluster point of (x_k) *belongs to* F.

Then there exists $\bar{x} \in F$ *such that* $x_k \to \bar{x}$.

Proof. First, the sequence (x_k) is bounded since $F \neq \emptyset$. So the sequence has at least one cluster point. Let y_1 and y_2 be two cluster points of the sequence. Extract $x_k^1 \to y_1$, $x_k^2 \to y_2$.

Let $y \in F$. $\lim_k \|x_k^1 - y\|^2 = \lim_k \|x_k^2 - y\|^2$ by assumption, so developing and reordering

$$\lim_k \|x_k^1\|^2 - \|x_k^2\|^2 = 2 \lim_k \langle \bar{x}, x_k^1 - x_k^2 \rangle = 2\langle \bar{x}, y_1 - y_2 \rangle \,. \tag{8.5}$$

Applying this equality to $\bar{x}$ equal to y_1 and y_2 successively, one gets $\langle y_1, y_2 - y_1 \rangle = \langle y_2, y_2 - y_1 \rangle$ so $y_1 = y_2$ and there is a single cluster point, the bounded sequence must converge. □

The Opial lemma is frequently applied to Fejér-monotone sequences (see Combettes (2001) for an excellent overview of the properties of such sequences).

Definition 8.12. *The sequence (x_k) is Fejér-monotone with respect to the nonempty, closed and convex set S if and only if, for all $\bar{x} \in S$:*

$$\|x_{k+1} - \bar{x}\| \leq \|x_k - \bar{x}\| \,. \tag{8.6}$$

It follows easily that every Fejér monotone sequence is bounded, and thus has weak cluster points. One also has that $\|x_k - \bar{x}\|$ converges (being decreasing and lower bounded) for all $\bar{x} \in S$.

An application of the Opial lemma is the following theorem on the convergence of α-averaged iterations.

Theorem 8.13. *Consider the Picard iterations $x_{k+1} = T(x_k)$ with $X_0 \in X$, where T is α-averaged and has* ***non-empty set of fixed points****.*

Then

1. *$(d(x_k, \operatorname{Fix} T))$ is decreasing*
2. *the sequence of squared iterate distance is summable*
3. *$x_k \to \bar{x} \in \operatorname{Fix} T$.*

Theorem 8.13 can be used to find fixed points of any nonexpansive operator, using the following fact.

Proposition 8.14. *For any $\alpha \neq 0$, R and $(1-\alpha)\mathrm{Id} + \alpha R$ have the same fixed points.*

Proof.

$$(1-\alpha)x + \alpha R(x) = x \Leftrightarrow \alpha x = \alpha R(x) \,. \tag{8.7}$$

□

Therefore, to find fixed points of a nonexpansive operator, one can pick $\alpha \in]0,1[$ and perform Picard iterations on the (by definition) α-averaged operator $T = (1-\alpha)\mathrm{Id} + \alpha_R$. Such iterations are called *Krasnoselskij-Mann iterations* and read:

$$x_{k+1} = x_k + \alpha(R(x_k) - x_k)\,. \tag{8.8}$$

It is possible to take a varying λ when applying Krasnoselskij-Mann iterations. The results are preserved as long as the sequence $(\lambda_n)_n \in]0,1|[$ is such that $\sum_{\mathbb{N}} \lambda_n(1-\lambda_n) = +\infty$.

8.2 Monotone operators

This section considers set-valued operators, denoted with the double arrow symbol: $T : X \rightrightarrows X$.

Definition 8.15. *An operator $T : X \rightrightarrows X$ is* monotone *if:*

$$\forall (x,y),\, \forall (u,v) \in (Tx) \times T(y),\, \langle u - v, x - y\rangle \geq 0\,. \tag{8.9}$$

If T is a function from $\mathbb{R}^d$ to $\mathbb{R}$, it simply means that T is increasing.

Definition 8.16. *A monotone operator T is* maximal monotone *if there does not exist another monotone operator whose graph strictly contains its graph. Equivalently, this means that for all x, $u \in T(x)$, y and v,*

$$\langle u - v, x - y\rangle \geq 0 \implies v \in T(y)\,. \tag{8.10}$$

The best illustration of monotone and maximal monotone operators is the subdifferential.

Proposition 8.17. *The subdifferential of any function is a monotone operator. The subdifferential of a Γ_0 function is maximal monotone.*

Proof. The first part almost follows from the definition of the subdifferential, by writing the subgradient lower bound at $u \in \partial f(x)$ at y, and conversely at x for $v \in \partial f(y)$, then summing.
For the second part: let $f \in \Gamma_0(\mathbb{R}^d)$. Let $\tilde{x}, \tilde{g}$ such that $\tilde{x} \notin \partial f(\tilde{x})$. We will show that there exist $x \in \mathbb{R}^d$ and $g \in \partial f(x)$ such that $\langle x - \tilde{x}, g - \tilde{g}\rangle < 0$, meaning that any extension of the graph of f, the resulting operator is not monotone.

Indeed let $x = \mathrm{argmin}_y\, f(y) + \frac{1}{2}\|y - \tilde{x} - \tilde{g}\|^2$ – for the well-definedness of x we use the hypothesis $f \in \Gamma_0(\mathbb{R}^d)$. Thus for $g = \tilde{x} + \tilde{g} - x$, we have $g \in \partial f(x)$, and

$$\langle g - \tilde{g}, x - \tilde{x}\rangle = -\|x - \tilde{x}\|^2 = -\|g - \tilde{g}\|^2\,. \tag{8.11}$$

At least one of these quantities is strictly negative (and so both are), otherwise $(x,g) = (\tilde{x}, \tilde{g})$ and $\tilde{x} \in \partial f(\tilde{g})$. Thus any extension of ∂f cannot be monotone. □

Definition 8.18 (Resolvent)**.** *For any $T : X \rightrightarrows X$, the set-valued operator $(\mathrm{Id} + T)^{-1}$ is called the* resolvent *of T.*

Proposition 8.19. *If T is monotone, its resolvent is nonexpansive (and a fortiori, it is singe-valued). If T is maximal monotone, the resolvent is defined on the whole space: for any y, the inclusion $y \in x + T(x)$ has a unique solution (if T is only monotone, uniqueness is guaranteed, but not existence).*

Example 8.20. *The resolvent of the subdifferential of a Γ_0 function is simply its proximal operator.*

Theorem 8.21 (Minty's theorem)**.** *An operator is firmly nonexpansive if and only if it is the resolvent of a maximally monotone operator.*

Proposition 8.22. *T is maximmaly monotone in the following cases:*

- *T is monotone and continuous*
- *T is cocoercive*
- *$\mathrm{Id} - T$ is nonexpansive*
- *T is linear and positive ($\langle x, T(x)\rangle \geq 0$)*
- *T is linear and* skew*, meaning that $T^* = -T$.*

8.3 Applications

Example 8.23 (Proximal point algorithm)**.** *Let $f \in \Gamma_0(\mathbb{R})$. By Fermat's rule, to find a minimizer of f is to find a point such that $0 \in \gamma\partial f(x^*)$, meaning a fixed point of $(\mathrm{Id} - \gamma\partial f)^{-1}$. Picard iterations of this algorithm are precisely the proximal point algorithm.*

Example 8.24 (Gradient descent)**.** *Let f be differentiable. A minimizer of f is a fixed point of $\mathrm{Id} - \gamma\nabla f$, which is contractive (TODO only if f is strongly convex, take this setting?) for $\gamma < 2/L$*

Example 8.25 (Forward-backward/proximal gradient algorithm)**.**

Example 8.26 (Douglas-Rachford algorithm)**.**

Example 8.27 (Chambolle-Pock algorithm)**.**

8.4 Exercises

Exercise 8.1. ☕ *Show that if T is nonexpansive, its set of fixed points is closed and convex.*

Exercise 8.2. ☕☕ *Show that if T_1 and T_2 are two nonexpansive operators and $\alpha \in]0, 1[$, if $\operatorname{Fix} T_1$ and $\operatorname{Fix} T_2$ intersect, then $\operatorname{Fix}(1-\alpha)T_1 + \alpha T_2 = \operatorname{Fix} T_1 \cap \operatorname{Fix} T_2$.*
Generalize to n operators.

Exercise 8.3 (medium)**.** *Let T_1 and T_2 be α_1- and α_2-averaged, such that $\operatorname{Fix} T_1 \cap \operatorname{Fix} T_2 \neq \emptyset$. Show that $\operatorname{Fix} T_1 \circ T_2 = \operatorname{Fix} T_1 \cap \operatorname{Fix} T_2$.*

9 Second order optimization: Newton method

Let f be a twice differentiable function. We have seen that gradient descent for stepsize $1/L$ on a L smooth function can be interpreted as a Majorization-Minimization approach, using the descent lemma to upper bound f globally by an isotropic parabola (majorization step), then minimizing this upper bound (minimization step).

The issue with this majorization is that it is very coarse: in dimension 2, take $f(x) = x_1^2 + 0.001x^2$. For this function, $L = 1$ and $\mu = 0.001$: we get a linear convergence rate for gradient descent,but the constant is very poor (see also Figure 2), because the upper bounding parabolas that we construct have Hessian $L\mathrm{Id}$, and $\mu \ll L$. In practice, this leads to very small steps being performed after the first one.

9.1 Newton method

We can try to preserve more information about the curvature, by using a 2nd order Taylor expansion of f – hence the name of "second-order methods":

$$f(y) \approx f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}\langle y - x, \nabla^2 f(x)(y - x) \rangle \ . \tag{9.1}$$

As in the MM interpretation of GD, we can derive an iterative algorithm for this: set $x = x_k$, and define x_{k+1} as the minimizer of the approximation. This yields

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \ . \tag{9.2}$$

Compared to gradient descent, the scalar stepsize has been replaced with the application of a PSD matrix, that better captures the curvature of f.

Let us illustrate the possible behaviors for Newton's method on a simple 1-dimensional convex function, $x \mapsto \sqrt{1 + x^2}$. This function is $\mathcal{C}^2$, with $f'(x) = \frac{x}{\sqrt{1+x^2}}$ and $f''(x) = \frac{1}{(1+x^2)^{3/2}}$, so Newton's method iterates are given by

$$x_{k+1} = x_k - x_k(1 + x_k^2) = -x_k^3 \ . \tag{9.3}$$

There are therefore, depending on $|x_0|$, three possible behaviors:

- extremely fast convergence towards the minimizer if $|x_0| < 1$
- oscillations between 1 and -1 if $|x_0| = 1$
- extremely fast divergence if $|x_0| > 1$

This example illustrates a crucial fact about the vanilla Newton method: it only has *local* convergence properties – it converges if initialized close enough to the minimizer. But when it does, it enjoys a *superlinear* convergence rate. Let us formalize this and derive a proof.

Proposition 9.1. *If $x^* = \operatorname{argmin} f(x)$, f has M-Lipschitz Hessian, $\nabla^2 f(x^*) \succeq \mu \mathrm{Id}_n$ ($\mu > 0$), and $\|x_0 - x^*\| \leq \frac{\mu}{2M}$, then:*

$$\|x_{k+1} - x^*\| \leq \frac{M}{\mu}\|x_k - x^*\|^2 \ . \tag{9.4}$$

It is hard to imagine a better rate than this one: the sequence $u_k = \frac{M}{\mu}\|x_k - x^*\|$ satisfies $u_{k+1} \leq u_k^2 \leq u_0^{2^k}$. If $u_0 < 1$, the number of iterations to reach $u_k < \varepsilon$ is of the order $\log(\log \varepsilon)$, that is, in practice, basically a constant!

Proof. By the fundamental theorem of calculus, for all x, h in $\mathbb{R}^d$:

$$\nabla f(x+h) - \nabla f(x) = \int_0^1 \nabla^2 f(x+th)h \,\mathrm{d}t \tag{9.5}$$

Hence

$$\nabla f(x_k) = \int_0^1 \nabla^2 f(x^* + t(x_k - x^*))(x_k - x^*) \,\mathrm{d}t \ , \tag{9.6}$$

and thus

$$x_{k+1} - x^* = x_k - x^* - [\nabla^2 f(x_k)]^{-1} \int_0^1 \nabla^2 f(x^* + t(x_k - x^*))(x_k - x^*) \,\mathrm{d}t \tag{9.7}$$

$$= [\nabla^2 f(x_k)]^{-1} \int_0^1 \Big(\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))\Big)(x_k - x^*) \,\mathrm{d}t \ . \tag{9.8}$$

Let's control the operator norm of the matrix being integrated, using M-Lipschitzness of the Hessian

$$\int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))\| \,\mathrm{d}t \leq \int_0^1 (1-t)M\|x_k - x^*\| \,\mathrm{d}t = \frac{M}{2}\|x_k - x^*\| \tag{9.9}$$

Finally, let's bound the operator norm of $[\nabla^2 f(x_k)]^{-1}$:

$$\nabla^2 f(x_k) \succeq \nabla^2 f(x^*) - M\|x_k - x^*\|\mathrm{Id}_n \succeq (\mu - M \underbrace{\|x_k - x^*\|}_{\leq \frac{\mu}{2M} \text{ (induction)}})\mathrm{Id}_n \succeq \frac{\mu}{2}\mathrm{Id}_n \ . \tag{9.10}$$

□

Despite the excellent convergence rate, Newton's method is not the "one algorithm to rule them all":

- it may diverge if not started close enough to optimum
- the Hessian may not be invertible
- the Hessian is costly to compute
- the Hessian is even more costly to invert[11] ($\mathcal{O}(d^3)$)

The principle of Quasi-Newton methods is to replace the inverse Hessian in Equation (9.2) by less costly approximations, in order to retain the very fast convergence rate while alleviating the computational burden.

[11] To compute $A^{-1}b$ numerically, one should solve the linear system $Ax = b$ with a LU factorization (Gaussian elimination) rather than invert A and apply to b: this is more stable and less costly. But the cost of the approach is still high

9.2 Quasi Newton methods

Algorithms in this section will have the following form:

$$\begin{cases} d_k = -B_k g_k \\ x_{k+1} = x_k + \rho_k d_k \end{cases} \quad \text{or} \quad \begin{cases} d_k = -H_k^{-1} g_k \\ x_{k+1} = x_k + \rho_k d_k \end{cases} \tag{9.11}$$

where $g_k = \nabla f(x_k)$, d_k is to be thought of as a descent direction, H_k (resp. B_k) is an approximation of the Hessian (resp. the inverse Hessian) of f at x_k, and ρ_k is a step size.

How to build such approximations? It is reasonable to impose that an approximation should satisfy the *secant condition* or *Quasi-Newton relation*:

$$g_{k+1} - g_k = H_{k+1}(x_{k+1} - x_k) \quad \text{or} \quad x_{k+1} - x_k = B_{k+1}(g_{k+1} - g_k) \tag{9.12}$$

Where does this condition come from? Suppose we have finished iteration $k+1$ and, to compute x_{k+2}, we want to construct (then minimize) ϕ_{k+1}, a quadratic approximation of f having H_{k+1} as Hessian:

$$\phi_{k+1}(x) = f(x_{k+1}) + \langle g_{k+1}, x - x_{k+1} \rangle + \frac{1}{2}\langle x - x_{k+1}, H_{k+1}(x - x_{k+1}) \rangle \ . \tag{9.13}$$

This quadratic approximation of f is exact at x_{k+1} ($\phi(x_{k+1}) = f(x_{k+1})$), and tangent at x_{k+1} too ($\nabla\phi_{k+1}(x_{k+1}) = \nabla f(x_{k+1})$). To impose the secant condition is simply to impose that f and ϕ_{k+1} are also tangent at the previous iterate, x_k.

The approximations in this section are updated iteratively, using rank 1 or rank 2 "corrections" from B_k (resp. H_k) to B_{k+1} (resp. H_{k+1}).

Following the notation of Nocedal and Wright, we write

$$y_k = g_{k+1} - g_k = \nabla f(x_{k+1}) - \nabla f(x_k) \ , \tag{9.14}$$

$$s_k = x_{k+1} - x_k \ . \tag{9.15}$$

9.2.1 The Broyden/SR1 formula

First, we consider rank-one updates of the Hessian, i.e. updates of the form $H_{k+1} = H_k + \sigma v v^\top$. How should we set σ and v so that it satisfies the secant condition (9.12)? We want:

$$y_k = H_{k+1} s_k \tag{9.16}$$

$$= H_k s_k + \sigma v v^\top s_k \tag{9.17}$$

$$= H_k s_k + (\sigma v^\top s_k) v \quad \text{(rotation trick } ab^\top c = b^\top c a) \ , \tag{9.18}$$

This does not give us v, but it gives its direction: v is proportional to $y_k - H_k s_k$. So we know[12] that there exists a scalar δ such that $v = \delta(y_k - H_k s_k)$, and we can plug it back in Equation (9.17) to solve for δ and obtain that

$$H_{k+1} = H_k + \frac{1}{s_k^\top (y_k - H_k s_k)} (y_k - H_k s_k)(y_k - H_k s_k)^\top \ . \tag{9.19}$$

[12] what if $v^\top s_k = 0$?

satisfies the secant condition. This is called the Broyden or SR1 formula.

This approximation removes the cost of computing the Hessian, and also its cost of storing (we just store the scalars $s_k^\top(y_k - H_k s_k)$ and vectors $(y_k - H_k s_k)$). But at first sight, we still have the $\mathcal{O}(d^3)$ cost of computing the update (linear system solving). However, since the update of the Hessian is rank-one, by the Sherman-Morrison formula (itself a special case of the Woodbury formula, see Exercise 9.2), so is the update of the inverse Hessian $B_k = H_k^{-1}$:

$$B_{k+1} = B_k - \frac{1}{(B_k y_k - s_k)^\top y_k}(B_k y_k - s_k)(B_k y_k - s_k)^\top \quad . \tag{9.20}$$

Therefore if we start at a simple B_0, say Id, then we only need to do, at each iteration, rank 1 updates that are cheap to store and apply (Exercise 9.1)!

9.2.2 BFGS

The BFGS update is rank two:

$$H_{k+1} = H_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{H_k s_k s_k^\top H_k}{s_k^\top H_k s_k} \quad . \tag{9.21}$$

Proposition 9.2 (Inverse Hessian update for BFGS)**.** *In terms of inverse Hessian B, the BFGS update (9.21) translates into*

$$B_{k+1} = \left(\mathrm{Id} - \frac{s_k y_k^\top}{s_k^\top y_k}\right) B_k \left(\mathrm{Id} - \frac{y_k s_k^\top}{s_k^\top y_k}\right) + \frac{s_k s_k^\top}{s_k^\top y} \quad . \tag{9.22}$$

Proof. For lighter notation we write H, y and s for H_k, y_k, s_k. We apply the Sherman-Morrison formula twice in Equation (9.21). First

$$\left(H + \frac{yy^\top}{y^\top s}\right)^{-1} = H^{-1} - \frac{H^{-1}yy^\top H^{-1}}{s^\top y + y^\top H^{-1} y} \tag{9.23}$$

Then:

$$H_{k+1}^{-1} = H^{-1} - \underbrace{\frac{H^{-1}yy^\top H^{-1}}{s^\top y + y^\top H^{-1} y}}_{M_1} + \underbrace{\frac{\left(\mathrm{Id} - \frac{H^{-1}yy^\top}{s^\top y + y^\top H^{-1} y}\right) ss^\top \left(\mathrm{Id} - \frac{yy^\top H^{-1}}{s^\top y + y^\top H^{-1} y}\right)}{s^\top H s - s^\top H (H^{-1} - \frac{H^{-1}yy^\top H^{-1}}{s^\top y + y^\top H^{-1} y}) H s}}_{M_2} \tag{9.24}$$

The last term simplifies:

$$M_2 = \frac{s^\top y + y^\top H^{-1} y}{(s^\top y)^2}\left(\mathrm{Id} - \frac{H^{-1}yy^\top}{s^\top y + y^\top H^{-1} y}\right) ss^\top \left(\mathrm{Id} - \frac{yy^\top H^{-1}}{s^\top y + y^\top H^{-1} y}\right) \tag{9.25}$$

$$= \frac{1}{s^\top y} ss^\top + \frac{y^\top H^{-1} y}{(s^\top y)^2} ss^\top - \frac{1}{(s^\top y)^2}\left(H^{-1}yy^\top ss^\top + ss^\top yy^\top H^{-1}\right) + M_1 \tag{9.26}$$

$$= \frac{1}{s^\top y} ss^\top + \frac{y^\top H^{-1} y}{(s^\top y)^2} ss^\top - \frac{1}{s^\top y}\left(H^{-1}ys^\top + sy^\top H^{-1}\right) + M_1 \tag{9.27}$$

And so M_1 cancels out in Equation (9.24):

$$H_{k+1}^{-1} = H^{-1} + \frac{ss^\top}{s^\top y} + \frac{y^\top H^{-1} y s s^\top}{(s^\top y)^2} - \frac{s y^\top H^{-1} + H^{-1} y s^\top}{s^\top y} \tag{9.28}$$

$$= H^{-1} + \frac{ss^\top}{s^\top y} + \frac{s y^\top H^{-1} y s^\top}{(s^\top y)^2} - \frac{s y^\top H^{-1} + H^{-1} y s^\top}{s^\top y} \tag{9.29}$$

$$= \left(\mathrm{Id} - \frac{s y^\top}{s^\top y}\right) H^{-1} \left(\mathrm{Id} - \frac{y s^\top}{s^\top y}\right) + \frac{ss^\top}{s^\top y} \tag{9.30}$$

□

One very beautiful way to see this update is the following one.

Proposition 9.3 (BFGS as KL minimization)**.** *The BFGS update Equation (9.21) is the solution of the following problem:*

$$\min_H -\log\det(H) + \langle H_k^{-1}, H\rangle \quad \textit{subject to} \quad H = H^\top, Hs = y \ , \tag{9.31}$$

where the objective function, also known as the Stein loss, is equal to the KL divergence between two Gaussians having the same means, aka the Bregman divergence induced by the negative logdet.

Proof. Introduce the Lagrangian:

$$\mathcal{L}(H, \lambda, \Theta) = -\log\det H + \langle H_k^{-1}, H\rangle + \langle \lambda, Hs - y\rangle + \langle \Theta, H - H^\top\rangle \ . \tag{9.32}$$

Notice that $\langle \lambda, Hs\rangle$ is a scalar, equal to its trace $\operatorname{tr} \lambda^\top H s = \operatorname{tr} s\lambda^\top H = \langle \lambda s^\top, H\rangle$ where this time the scalar product is over matrices. Similarly, $\langle \Theta, H - H^\top\rangle = \langle \Theta, H\rangle - \langle \Theta, H^\top\rangle = \langle \Theta, H\rangle - \langle \Theta^\top, H\rangle = \langle \Theta - \Theta^\top, H\rangle$.

This gives us the two gradients we need to write the first-order optimality condition over H. Using that the gradient of logdet is the inverse, we obtain:

$$\nabla_H \mathcal{L} = 0 \iff H^{-1} = H_k^{-1} + \lambda s^\top + \Theta - \Theta^\top \ . \tag{9.33}$$

First, we get rid of Θ: H must be symmetric hence H^{-1} too; H_k is symmetric (it's the approximate Hessian at the previous iteration) and this gives:

$$\Theta - \Theta^\top = \frac{1}{2}(s\lambda^\top - \lambda s^\top) \ , \tag{9.34}$$

so

$$H^{-1} = H_k^{-1} + \frac{1}{2}(\lambda s^\top + s\lambda^\top) \ . \tag{9.35}$$

To find λ, we use the other condition, $Hs = y$ aka $s = H^{-1}y$. This gives

$$\frac{1}{2} s^\top y \lambda = (s - H_k^{-1} y) - \frac{1}{2}\lambda^\top y s \ . \tag{9.36}$$

This may seem impossible to solve as λ appears twice, but it nevertheless tells us something: there exists a scalar a such that

$$\lambda = -\frac{2}{s^\top y} H_k^{-1} y + as \ . \tag{9.37}$$

Plug back this expression to obtain H^{-1} as a function of a, then use $H^{-1}y = s$ to solve in a (it's tedious, but a good exercise). Then plugging back a into λ we obtain

$$\lambda s^\top = -\frac{2}{s^\top y} H_k^{-1} y s^\top + \frac{y^\top H_k^{-1} y + s^\top y}{(s^\top y)^2} s s^\top \ . \tag{9.38}$$

To conclude let's be a little lazy: $\frac{1}{2}(\lambda s^\top + s\lambda^\top)$ is the symmetric part of $\lambda s^\top$. As visible in Equation (9.38), the term in $ss^\top$ is already symmetric, so we only need to symmetrize the other one:

$$\frac{1}{2}(\lambda s^\top + s\lambda^\top) = \frac{y^\top H_k^{-1} y + s^\top y}{(s^\top y)^2} s s^\top - \frac{1}{s^\top y}\left(H_k^{-1} y s^\top + s y^\top H_k^{-1}\right) \ . \tag{9.39}$$

and we recognize the computation we've been through in Equation (9.28), which allows to conclude invoking Proposition 9.2. □

Proposition 9.4 (BFGS as weighted Frobenius norm minimization)**.** *Let W be any positive definite matrix satisfying $y_k = W s_k$. Then the inverse Hessian B_{k+1} solves:*

$$\min_B \|W^{1/2}(B - B_k)W^{1/2}\| \quad \textit{subject to} \quad B = B^\top,\ By = s \ . \tag{9.40}$$

Proof. The proof is similar to that of Proposition 9.3. Introduce the Lagrangian:

$$\mathcal{L}(B, \lambda, \Theta) = \frac{1}{2}\|W^{1/2}(B - B_k)W^{1/2}\| + \langle \lambda, By - s\rangle + \langle \Theta, B - B^\top\rangle \ . \tag{9.41}$$

The gradient with respect to B of the first term is $W(B - B_k)W$. For the second one, $\langle \lambda, By\rangle$ is a scalar, equal to its trace $\operatorname{tr} \lambda^\top B y = \operatorname{tr} y\lambda^\top B = \langle \lambda y^\top, B\rangle$ where this time the scalar product is over matrices. For the last one, $\langle \Theta, B - B^\top\rangle = \langle \Theta, B\rangle - \langle \Theta, B^\top\rangle = \langle \Theta, B\rangle - \langle \Theta^\top, B\rangle = \langle \Theta - \Theta^\top, B\rangle$.

Therefore:

$$\nabla_H \mathcal{L} = 0 \iff WBW = WB_kW + \lambda y^\top + \Theta - \Theta^\top \ . \tag{9.42}$$

Next, we get rid of Θ: B must be symmetric hence B^{-1} too; B_k is symmetric by assumption and so:

$$\Theta - \Theta^\top = \frac{1}{2}(y\lambda^\top - \lambda y^\top) \ , \tag{9.43}$$

so

$$WBW = WB_kW + \frac{1}{2}(\lambda y^\top + y\lambda^\top) \tag{9.44}$$

$$B = B_k + \frac{1}{2}W^{-1}(\lambda y^\top + y\lambda^\top)W^{-1} \ . \tag{9.45}$$

To find λ, we use the other condition, $By = BWs = s$. Applying (9.44) to s thus yields

$$WBWs = WBy = Ws = y = WB_k y + \frac{1}{2}(\lambda y^\top s + y\lambda^\top s) \tag{9.46}$$

Hence,

$$\frac{y^\top s}{2}\lambda = y - WB_k y - \lambda^\top s y \ , \tag{9.47}$$

so there exists a such that

$$\lambda = -\frac{2}{y^\top s} WB_k y + ay \ . \tag{9.48}$$

Since $By = s$,

$$s = B_k y + \frac{1}{2}W^{-1}(\lambda y^\top + y\lambda^\top)W^{-1}y \tag{9.49}$$

$$= B_k y + \frac{1}{2}W^{-1}\left(\left(-\frac{2}{y^\top s}WB_k y + ay\right)y^\top + y\left(-\frac{2}{y^\top s}y^\top B_k W + ay^\top\right)\right)s \tag{9.50}$$

$$= \frac{1}{2}W^{-1}\left(ayy^\top + y\left(-\frac{2}{y^\top s}y^\top B_k W + ay^\top\right)\right)s \tag{9.51}$$

$$= \frac{1}{2}asy^\top s - \frac{1}{y^\top s}sy^\top B_k W s + \frac{1}{2}asy^\top s \tag{9.52}$$

$$= ay^\top ss - \frac{1}{y^\top s}y^\top B_k W ss \tag{9.53}$$

$$= ay^\top ss - \frac{1}{y^\top s}y^\top B_k y s \tag{9.54}$$

Hence

$$a = \frac{y^\top s + y^\top B_k y}{(y^\top s)^2} \ . \tag{9.55}$$

So

$$\lambda y^\top = -\frac{2}{y^\top s}WB_k yy^\top + \frac{y^\top s + y^\top By}{(y^\top s)^2}yy^\top \tag{9.56}$$

$$\frac{1}{2}\left(\lambda y^\top + y\lambda^\top\right) = -\frac{1}{y^\top s}\left(WB_k yy^\top + yy^\top B_k W\right) + \frac{y^\top s + y^\top By}{(y^\top s)^2}yy^\top \ . \tag{9.57}$$

Substituting in (9.45) and using $W^{-1}y = s$,

$$B = B_k + \frac{1}{y^\top s}W^{-1}\left(\lambda y^\top + y\lambda^\top\right)W^{-1} \tag{9.58}$$

$$= B_k - \frac{1}{y^\top s}\left(B_k ys^\top + sy^\top B_k\right) + \frac{y^\top s + y^\top By}{(y^\top s)^2}ss^\top \tag{9.59}$$

$$= \left(\mathrm{Id} - \frac{sy^\top}{y^\top s}\right)B_k\left(\mathrm{Id} - \frac{sy^\top}{s^\top y}\right) + \frac{ss^\top}{y^\top s} \ . \tag{9.60}$$

□

9.2.3 DFP

The DFP update is the same as BFGS, swapping the roles of H, y, s with B, s, y:

$$B_{k+1} = B_k + \frac{s_k s_k^\top}{y_k^\top s_k} - \frac{B_k y_k y_k^\top B_k}{y_k^\top B_k y_k} \ . \tag{9.61}$$

9.3 Exercises

Exercise 9.1. ☕ *Show that any $n \times n$ matrix of rank 1 writes $xy^\top$ with $x, y \in \mathbb{R}^n$. Show that the cost of multiplying by a rank one matrix is $2n$ instead of the usual n^2.*

Exercise 9.2 (Woodbury inverse formula)**.** ☕☕ *Let $n, k \in \mathbb{N}$. Let $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{k\times n}$. Show that*

$$(\mathrm{Id} + UV)^{-1} = \mathrm{Id} - U(\mathrm{Id} + VU)^{-1}V \ . \tag{9.62}$$

Show that for any invertible A,

$$(A + UV)^{-1} = A^{-1} - A^{-1}U(\mathrm{Id} + VA^{-1}U)^{-1}VA^{-1} \ . \tag{9.63}$$

When $k = 1$ and U and V respectively become column and row vectors u and $v^\top$, the formula is known as the Sherman-Morrison formula:

$$(A + uv^\top)^{-1} = A^{-1} - \frac{1}{1 + v^\top A^{-1}u} A^{-1}uv^\top A^{-1} \ . \tag{9.64}$$

What is the naive cost of computing $A^{-1}uv^\top A^{-1}$? And the clever cost?

10 The finite sum structure: stochastic methods

10.1 Stochastic gradient descent

We consider problems with the *finite sum structure*:

$$\min_x f(x) = \sum_{i=1}^n f_i(x) \ . \tag{10.1}$$

Such problems notably arise from the maximum likelihood framework of Section 1.1 where each f_i is the negative log-likelihood of a data point. Computing the gradient of f normally requires computing the gradient of each f_i, that scales with n. The principle of *stochastic* gradient descent is to replace $\nabla f(x_t)$ at iteration $t \in \mathbb{N}$ of gradient descent by $\nabla f_{i_t}(x_t)$ for some i_t sampled uniformly in $[n]$:

$$x_{t+1} = x_t - \eta \nabla f_{i_t}(x_t) \ . \tag{10.2}$$

The update direction is n times less costly to compute, and on average correct since:

$$\mathbb{E}_{i \sim \mathcal{U}([n])}[\nabla f_i(x)] = \sum_{i'=1}^n \mathbb{P}(i = i') \nabla f_{i'}(x) = \frac{1}{n} \sum_{i'=1}^n \nabla f_{i'}(x) = \nabla f(x) \ . \tag{10.3}$$

Let's make this rigorous.

Proposition 10.1. *Let f be a μ-strongly convex function with (unique) minimizer x^*. Let $0 < \eta \leq \frac{1}{\mu}$ and assume that the iterates of stochastic descent, x_t, have bounded gradients in expectation:*

$$\forall j \in [p], \forall t \in \mathbb{N}, \ \mathbb{E}[\|\nabla f_j(x_t)\|^2] \leq B^2 \ . \tag{10.4}$$

Then

$$\mathbb{E}[\|x_t - x^*\|^2] \leq (1 - \eta\mu)^t \|x_0 - x^*\|^2 + \frac{\eta}{\mu} B^2 \ . \tag{10.5}$$

As usual a few comments before moving into the proof:

- There's randomness so our rate is in expectation. We cannot hope for "absolute" rates without assumption since we could get very unlucky and sample always the same data point. But that is unlikely to happen: the same kind of rates with high probability can also be obtained.
- The expectation in (10.4) is over the indices $(i_0, \dots, i_t)$ that are sampled.
- We have two bounds in the rate: an exponentially fast "forgetting" of the initial conditions, and an "irreducible" term: the upper bound does not go to 0 for fixed η. To make the second term small, we need η small, but this makes the first phase slower...
- The bound on the gradients is (this is my personal opinion) a circular reasoning (gradients are bounded if the algorithm converges, but it's needed to prove that the algorithm converges), and there's no real reason to think it holds a priori.

Proof.

$$\|x_{t+1} - x^*\|^2 = \|x_t - \eta \nabla f_{i_t}(x_t) - x^*\|^2 \tag{10.6}$$
$$= \|x_t - x^*\|^2 - 2\eta \langle f_{i_t}(x_t), x_t - x^* \rangle + \eta^2 \|\nabla f_{i_t}(x_t)\|^2 \ . \tag{10.7}$$

We can take expectation with respect to $(i_0, \ldots, i_t)$ on both sides, and since x_t is independent of i_t,

$$\mathbb{E}\|x_{t+1} - x^*\|^2 = \mathbb{E}\|x_t - x^*\|^2 - 2\eta \langle \nabla f(x_t), x_t - x^* \rangle + \eta^2 \mathbb{E}\|\nabla f_{i_t}(x_t)\|^2 \tag{10.8}$$
$$\leq \mathbb{E}\|x_t - x^*\|^2 - 2\eta \langle \nabla f(x_t), \mathbb{E}[x_t - x^*] \rangle + \eta^2 B^2 \ . \tag{10.9}$$

Then since $\nabla f(x^*) = 0$, by μ-strong convexity of f (Exercise 3.3),

$$\mu \|x_t - x^*\|^2 \leq \langle \nabla f(x_t), x_t - x^* \rangle \ . \tag{10.10}$$

Thus

$$\mathbb{E}\|x_{t+1} - x^*\|^2 \leq (1 - 2\eta\mu)\mathbb{E}\|x_t - x^*\|^2 + \eta^2 B^2 \tag{10.11}$$
$$\leq (1 - 2\eta\mu)^{t+1}\mathbb{E}\|x_0 - x^*\|^2 + \eta^2 B^2 \sum_{t'=0}^{t} (1 - 2\eta\mu)^{t'} \tag{10.12}$$
$$\leq (1 - 2\eta\mu)^{t+1}\mathbb{E}\|x_0 - x^*\|^2 + \frac{1}{2\eta\mu}\eta^2 B^2 \tag{10.13}$$
$$\tag{10.14}$$

□

10.2 Variance reduction

The problems of SGD are the following: if you start at optimum, you move away from it. You need to reduce the stepsize to stop moving too much, but then there's a hard tradeoff because you still want to go fast to the minimizer.

f_i μ strongly convex and L-smooth,

GD: Fixed learning rate, O(1 /k) or linear, no problem structure exploitation. SGD: decreasing stepsize, O(1/sqrt k) or O(1 / k). Can we get the best of both worlds? Variance is the issue.

	GD	SGD	Variance reduced
Cost per iteration	n	1	1
convex rate	sqrt	sqrt	sqrt
smooth convex rate	1/ k	$1/\sqrt{k}$	1 / k
smooth strongly convex rate	linear	1 / k	linear

The idea of variance reduction is the following: random variable X, estimate $\mathbb{E}X$ but with lower variance. Use a rv Y correlated with X with mean easier to estimate, introduce $\theta_\alpha = \alpha(X - y) + \mathbb{E}[Y]$ for $\alpha \in [0, 1]$. Unbiased if $\alpha = 1$ or $\mathbb{E}X = \mathbb{E}Y$.

Variance:

$$\begin{aligned}\mathbb{E}\theta_\alpha &= \mathbb{E}[(\alpha(X-Y)-\alpha\mathbb{E}[(X-Y)])^2] &(10.15)\\ &= \alpha^2\mathbb{E}[(X-\mathbb{E}X-(Y-\mathbb{E}Y))^2] &(10.16)\\ &= \alpha^2(\operatorname{Var}X+\operatorname{Var}Y-2\operatorname{cov}(X,Y)) &(10.17)\end{aligned}$$

can be lower than $VarX$.

Application of idea: SAGA. X is the stochastic gradient, Y is the past stored gradient.

$$\begin{cases} j\sim\mathcal{U}(n)\\ g_j^k=\nabla f_j(x^k)\\ thereststaysequal\\ x^k=x^{k-1}-\gamma[g_j^k-g_j^{k-1}+\frac{1}{n}\sum_{i=1}^n g_i^{k-1}]\end{cases} \tag{10.18}$$

Eventually, apply a prox.

A few comments: storage of gradients is expensive, for GLMs can be diminished. one must be careful regarding the update of the mean. sparse vs dense issue.

SAG: biased gradient update:

$$x^k=x^{k-1}-\gamma[\frac{g_j^k-g_j^{k-1}}{n}+\frac{1}{n}\sum_{i=1}^n g_i^{k-1}] \tag{10.19}$$

SVRG: snapshot, refrence point $\tilde{x}$.

$$x^k=x^{k-1}-\gamma[\nabla f_j(x^k)-\nabla f_j^{\tilde{x}}+\frac{1}{n}\sum_{i=1}^n\nabla f_i^{\tilde{x}}] \tag{10.20}$$

update frequency of $\tilde{x}$?

11 Mirror descent

Nick Harvey lecture notes: https://www.cs.ubc.ca//~nickhar/F18-531/Notes20.pdf SUper cool connection: MD is Natural Gradient descent on the dual Riemman manifold: https://arxiv.org/pdf/1310.7780.pdf

Key to the Mirror descent algorithm is the concept of Bregman divergence.

Definition 11.1 (Legendre function)**.** *A Γ_0 function J is Legendre iff:*

- *it is strictly convex on the interior of its domain*
- *it is essentially smooth, meaning that it is continuously differentiable on the interior of its domain, and $\|\nabla J(x_k)\| \to \infty$ for any sequence x_k converging to a point on the boundary of* dom J*.*

Definition 11.2 (Bregman divergence)**.** *Let J be a convex function. The Bregman divergence induced by J is $D_J : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined by:*

$$D_j(x, y) = J(x) - J(y) - \langle \nabla J(y), x - y \rangle . \tag{11.1}$$

For brevity, we may sometimes omit the J subscript. By convexity of J, it is positive. It is usually not symmetric. It is point-separating if J is strictly convex. It is not like a distance, it is like a square distance. It is the difference at x between J and its linear upper bound at x. It is convex in x but usually not in y

Proposition 11.3 (Generalization of Pythagoras)**.**

$$D_J(x, y) + D_J(z, x) - D_J(z, y) = \langle \nabla J(x) - \nabla J(y), x - z \rangle . \tag{11.2}$$

See also Section 12.2 for an analysis of the continuous time version.

11.1 The mirror descent algorithm

Assumptions: J Legendre, J ρ-strongly convex in some norm. The mirror descent algorithm is given by the following equation:

$$\nabla J(x_{k+1}) = \nabla J(x_k) - \eta \nabla f(x_k) . \tag{11.3}$$

Proposition 11.4. *Convergence rates are similar to that of subgradient descent.*

Proof.

$$\begin{aligned} f(x_k) - f(x^*) &\leq \langle \nabla f(x_k), x_k - x^* \rangle \\ &= \frac{1}{\eta} \langle J(x_k) - J(x_{k+1}), x_k - x^* \rangle \\ &= \frac{1}{\eta} \left[D(x_k, x_{k+1}) + D(x^*, x_k) - D(x^*, x_{k+1}) \right] \end{aligned}$$

Sum to telescope:

$$\sum_1^t f(x_k) - f(x^*) \leq \frac{1}{\eta}\left[\sum_1^t D(x_k, x_{k+1}) + D(x^*, x_1) - D(x^*, x_{t+1})\right]$$
$$\leq \frac{1}{\eta}\left[\sum_1^t D(x_k, x_{k+1}) + D(x^*, x_1)\right]$$

It remains to bound the first term, for which we use strong convexity. Strong convexity means that $D(x_k, x_{k+1})$ is lower bounded, which is not very useful here. So we make the opposite of a Bregman divergence appear:

$$\begin{aligned} D(x_k, x_{k+1}) &= J(x_k) - J(x_{k+1}) - \langle \nabla J(x_{k+1}), x_k - x_{k+1}\rangle \\ &= \langle \nabla J(x_{k+1}) - \nabla J(x_k), x_{k+1} - x_k\rangle - D(x_{k+1}, x_k) \\ &= \langle -\eta \nabla f(x_k), x_{k+1} - x_k\rangle - D(x_{k+1}, x_k) \\ &= -\eta L\|x_{k+1} - x_k\| - \frac{\mu}{2}\|x_{k+1} - x_k\|^2 \\ &= \frac{\eta^2 L^2}{2\mu} \end{aligned}$$

Note: handling projection is trivial because Bregman projection satisfies $D(x^*, x_{k+1}) \leq D(x^*, y_{k+1})$ where y_{k+1} is the intermediate step that gets projected into x_{k+1}.

Plug back and divide by t:

$$\frac{1}{t}\sum_1^t f(x_k) - f(x^*) \leq \eta\frac{L^2}{2\mu} + \frac{D(x^*, x_1)}{\eta t}$$

Minimize RHS in η gives $\eta = \sqrt{\frac{2\mu D}{L^2 t}}$ and an upper bound value of $\sqrt{\frac{2DL^2}{\mu t}}$. □

Better rates: see also https://www.cs.uic.edu/~zhangx/teaching/bregman.pdf

11.2 Mirror descent on the simplex: the exponentiated gradient algorithm

The following potential, called *entropy*, is particularly adapted to optimization on the simplex:

$$J(x) = \sum_1^d x_i \log x_i \ . \tag{11.4}$$

It is 1-strongly convex with respect to the ℓ_1 norm (a result known as Pinsker's inequality), and its associated Bregman divergence is

$$D(x,y) = \sum_1^d x_i \log x_i - y_i \log y_i - (1+\log y_i)(x_i - y_i) \tag{11.5}$$

$$= \sum_{i=1}^d x_i \log \frac{x_i}{y_i} - \sum_{i=1}^d x_i + \sum_{i=1}^d y_i \tag{11.6}$$

which is called the *Kullback-Leibler divergence* (usually considered for vectors x and y on the simplex, so the last two sums cancel). This Bregman divergence induces a nicely behaved projection onto the simplex. For $y \in \mathbb{R}^d_{++}$ (the case where y has vanishing entries is easy to handle similarly), let us compute

$$\operatorname*{argmin}_{x \in \Delta_d} \sum_{i=1}^d x_i \log \frac{x_i}{y_i} - \sum_{i=1}^d x_i + \sum_{i=1}^d y_i \tag{11.7}$$

This problem can be solved by introducing the Lagrangian, with $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}^d_+$

$$\mathcal{L}(x,\lambda,\mu) = \sum_{i=1}^d x_i \log \frac{x_i}{y_i} - \sum_{i=1}^d x_i + \lambda(\sum_{i=1}^d x_i - 1) - \sum_{i=1}^d \mu_i x_i \tag{11.8}$$

The saddle point conditions on $\mathcal{L}$ yield

$$\begin{cases} \mu_i x_i = 0 \\ \log \frac{x_i}{y_i} + 1 - 1 + \lambda - \mu_i = 0 \ \text{ aka } \ x_i = y_i \exp(\lambda - \mu_i) \\ \sum_{i=1}^d x_i = 1 \end{cases} \tag{11.9}$$

The second equation shows that x_i cannot have value 0, hence $1 = \sum_{i=1}^d x_i = \exp(\lambda) \sum_{i=1}^d y_i$, hence the solution is given by:

$$\operatorname*{argmin}_{x \in \Delta_d} D(x,y) = \left(\frac{y_i}{\sum_{i=1}^d y_i} \right)_{i \in [d]} \tag{11.10}$$

Recall that the mirror descent iterates are given by $x_{k+1} = \nabla J^*(\nabla J(x_k) - \eta \nabla f(x_k))$, and we have

$$x = \nabla J^*(u) \Leftrightarrow u = \nabla J(x) \Leftrightarrow u = 1 + \log x \Leftrightarrow x = \exp(u-1) \tag{11.11}$$

So, the mirror descent algorithm for the entropic potential is:

$$x_{k+1} = \exp(1 + \log x_k - \eta \nabla f(x_k) - 1) = x_k \exp(-\eta \nabla f(x_k)), \tag{11.12}$$

which is known as the exponentiated gradient algorithm. (TODO and normalization if projection)

11.3 Online learning and mirror descent

The online learning framework is the following: we don't minimize a function but, at time $t \in \mathbb{N}$, take a decision $x_t \in \mathbb{R}^d$, incur cost $f_t(x_t)$ (f_t possibly chosen in an adversarial fashion; we don't know it when picking x_t!), receive/observe $g_t \in \partial f_t(x_t)$ (information: how could we have done better for this stage), and can use it to take a new decision x_{t+1}.

How to measure the performance? The regret of an online algorithm

Definition 11.5 (Regret)**.** *The regret at horizon $T \in \mathbb{N}$ of an algorithm/a sequence of actions x_t is:*

$$R(T) = \sum_{t=1}^{T} f_t(x_t) - \min_x \sum_{t=1}^{T} f_t(x) \ , \tag{11.13}$$

that is, the difference between the incurred cost and the minimal cost for a fixed strategy which knows all the f_t's in hindsight.

Basic algorithm: subgradient descent. Works exactly as in the fixed f case under the same assumptions on f.

TODO example $f_t = x, 1 - x$ even/odd

12 Continuous view: gradient flows, mirror flows, and Nesterov

TODO merge this with below

Let's jump ahead to the continuous time version of gradient descent that we'll see later, the gradient flow. The gradient flow is an Ordinary Differential Equation (ODE):

$$\begin{aligned} x(0) &= x_0, \\ \dot{x}(t) &= -\nabla f(x(t)) \ . \end{aligned} \tag{12.1}$$

Suppose one wants to numerically approximate the solution to this equation. We pick a time discretization step $\eta > 0$, and define time instants $t_k = k\eta$ for $k \in \mathbb{N}$. We will construct a sequence x_k that should approximate the continuous version $x(t_k)$. We approximate the derivative $\dot{x}(t_k)$ by finite differences $\frac{x_{k+1}-x_k}{\eta}$.

Thus a discrete approximation of (12.1) is:

$$x_{k+1} - x_k = -\eta \nabla f(x_k) \ ; \tag{12.2}$$

this is a *forward* Euler discretization scheme, that gives the iterations of gradient descent.

On the other hand, one could chose a *backward* Euler scheme, also called *implicit* (because it does not give an explicit value of x_k):

$$x_{k+1} - x_k = -\eta \nabla f(x_{k+1}) \ . \tag{12.3}$$

Rewriting this as $(\mathrm{Id} + \eta \nabla f)(x_{k+1}) = x_k$, it becomes visible that this scheme is the proximal point algorithm!

12.1 Gradient flow

In this section, we take the limit of algorithms as the stepsize goes to zero and times become continuous. Indeed gradient descent can be seen as a forward Euler discretization of the following Ordinary Differential Equation (ODE):

$$\dot{x}(t) = -\nabla f(x(t)) \tag{12.4}$$

while the proximal point algorithm is a backward/implicit discretization, since $\mathrm{prox}_f = (\mathrm{Id} + \nabla f)^{-1}$ for a differentiable function f.

Studying the gradient flow is often easier than studying its discretized counterpart, gradient descent: the same results can usually be obtained, but the proofs are much simpler and require fewer assumptions. In particular, stepsizes and Lipschitz constants are not involved.

Proposition 12.1. *The function value $f(x(t))$ decreases along the flow. If f is convex, the gradient norm $\|f(x(t))\|$ also decreases along the flow.*

Proof. A direct calculation shows:

$$\frac{\mathrm{d}}{\mathrm{d}t} f(x(t)) = \langle \nabla f(x(t)), \frac{\mathrm{d}}{\mathrm{d}t} x(t) \rangle = -\|\nabla f(x(t))\|^2 \leq 0 \ . \tag{12.5}$$

If f is convex, its Hessian is semidefinite positive[13] and thus:

$$\frac{d}{\mathrm{d}t}\|\nabla f(x(t))\|^2 = -\nabla f(x(t))\nabla^2 f(x(t))\nabla f(x(t)) \leq 0 \ . \tag{12.6}$$

□

Proposition 12.2 (Convergence rates)**.** *Let x^* be a minimizer of f. Then*

$$f(x(t)) - f(x^*) \leq \frac{\|x(0) - x^*\|^2}{2t} \tag{12.7}$$

$$\|\nabla f(x(t))\|^2 \leq \frac{f(x(0))}{t} \ . \tag{12.8}$$

Proof. Again, differentiating a cleverly chosen functional gives:

$$\frac{\mathrm{d}}{\mathrm{d}t}\tfrac{1}{2}\|x(t) - x^*\|^2 = -\langle \nabla f(x(t)), x(t) - x^*\rangle \leq f(x^*) - f(x(t)) \ . \tag{12.9}$$

First, this shows that $x(t)$ gets closer to any minimizer as t increases since the RHS is negative. Second, since $f(x(t))$ is decreasing,

$$f(x(t)) - f(x^*) \leq \frac{1}{t}\int_0^t f(x(s)) - f(x^*)\,\mathrm{d}s \leq \frac{1}{2t}\|x(0) - x^*\|^2 - \frac{1}{2t}\|x(t) - x^*\|^2 \ , \tag{12.10}$$

which proves the first result. For the second one, since the squared gradient norm decreases and is equal to $-\frac{\mathrm{d}}{\mathrm{d}t}f(x(t))$ by Equation (12.5),

$$\|\nabla f(x(t))\|^2 \leq -\frac{1}{t}\int_0^t \frac{\mathrm{d}}{\mathrm{d}s}f(x(s)) = \frac{1}{t}(f(x(0)) - f(x(t))) \leq \frac{1}{t}f(x(0)) \ . \tag{12.11}$$

□

12.2 Mirror descent flow

Definition 12.3 (Mirror descent flow)**.** *For a Legendre function J, and a convex differentiable function f, the mirror descent flow is the solution of the ODE*

$$\nabla^2 J(x(t))\dot{x}(t) = -\nabla f(x(t))\,. \tag{12.12}$$

An explicit discretization of this flow with time step η gives the mirror descent equation (Equation (11.3))

Proposition 12.4. *We have:*

1. *for any minimizer x^* of f, $D(x^*, x(t))$ is decreasing*
2. *$f(x(t))$ decreases with t*
3. *$f(x(t)) - f(x^*) \leq \frac{D(x^*, x_0)}{t}$ for any minimizer x^**

[13] what if f is not twice differentiable?

Proof. 1. Let $z \in \mathbb{R}^d$.

$$\frac{\mathrm{d}}{\mathrm{d}t} D(z, x(t)) = \langle \nabla J(x), \dot{x} \rangle - \langle \nabla^2 J(x)\dot{x}, z - x \rangle - \langle \nabla J(x), -\dot{x} \rangle \tag{12.13}$$

$$= \langle -\nabla^2 J(x)\dot{x}, z - x \rangle \tag{12.14}$$

$$= \langle \nabla f(x), z - x \rangle \tag{12.15}$$

$$\leq f(z) - f(x)\,. \tag{12.16}$$

Now taking z to be a minimizer yields $\frac{\mathrm{d}}{\mathrm{d}t} D(z, x(t)) \leq 0$.

$$\dot{f}(x(t)) = \langle \nabla f(x(t)), \dot{x}(t) \rangle = -\langle \nabla^2 J(x(t))\dot{x}(t), \dot{x}(t) \rangle \leq 0 \tag{12.17}$$

because J is convex so $\nabla^2 J$ is p.s.d. everywhere.

2.

$$\frac{1}{t}\left(D(x^*, x) - D(x^*, x_0)\right) = \frac{1}{t}\int_0^t \frac{\mathrm{d}}{\mathrm{d}s} D(x^*, x(s))\,\mathrm{d}s \tag{12.18}$$

$$\leq \frac{1}{t}\int_0^t f(x^*) - f(x(s))\,\mathrm{d}s \tag{12.19}$$

$$\leq \frac{1}{t}\int_0^t f(x^*) - f(x(t))\,\mathrm{d}s \quad (\quad f(x(s)) \text{ decreases}) \tag{12.20}$$

$$= f(x^*) - f(x(t)) \tag{12.21}$$

In similar settings without decrease of f, one can also use an argument using $f(x^*) - \frac{1}{t}\int_0^t f(x(s))\,\mathrm{d}s \leq f(x^*) - f(\bar{x}(t))$ with $\bar{x}(t) = \frac{1}{t}\int_0^t x(s)\,\mathrm{d}s$, but here we can do better since f decreases.

□

12.3 An ODE modeling Nesterov acceleration

Can we do better than the $\mathcal{O}(1/t)$ rate of gradient flow? Su Boyd and Candès consider the following equation:

$$\ddot{x}(t) + \beta(t)\dot{x}(t) + \nabla(f(x(t)))\ . \tag{12.22}$$

By letting $\epsilon(t) = \frac{A}{2}\|x - x^* + C\dot{x}\|^2 + B(f(x(t)) - f^*)$, they have

$$\dot{\varepsilon}(t) = \frac{\dot{A}}{2}\|x - x^*\|^2 + (\dot{A}C + A(\dot{C} + 1 - C\beta))\langle x - x^*, \dot{x} \rangle - AC\langle x - x^*, \nabla f(x) \rangle \tag{12.23}$$

$$+ (\frac{\dot{A}C^2}{2} + AC(\dot{C} + 1 - C\beta))\|\dot{x}\|^2 + (B - AC^2)\langle \dot{x}, \nabla f(x) \rangle + \dot{B}(f(x) - f^*) \tag{12.24}$$

To simplify the expression, they take many terms equal to 0 by choosing $B = AC^2$, $\dot{A}C^2 = 0$ so $\dot{A} = 0$, $\dot{C} + 1 - C\beta = 0$, so $\dot{C} = 1/2$.

yielding:

$$f(x(t)) - f(x^*) = \mathcal{O}\left(\frac{1}{B(t)}\right) = \mathcal{O}\left(\frac{1}{t^2}\right)\ . \tag{12.25}$$

Discretization, Heavy ball vs Nesterov.

Possible discretization:

$$\frac{x_{k+1}-2x_k+x_{k-1}}{h^2}+\frac{\alpha}{kh}\frac{x_{k+1}-x_k}{h}+\nabla f(x_k)=0 \ . \tag{12.26}$$

$$x_{k+1}=x_k+(1-\tfrac{\alpha}{k})(x_k-x_{k-1})-h^2\nabla f(x_k) \ . \tag{12.27}$$

also $\frac{k-1}{k+2}$

Nesterov choice of momentum: $t_1=1$

$$\begin{cases} t_{k+1}=\frac{1+\sqrt{1+4t_k^2}}{2} \\ x_{k+1}=y_k-\frac{1}{L}\nabla f(y_k) \\ y_{k+1}=x_k+\frac{t_k-1}{t_{k+1}}(x_{k+1}-x_k) \end{cases} \tag{12.28}$$

The FISTA algorithm is the same, with a prox in the x_k step.

Heavy Ball (different result?):

$$x_{k+1}=x_k-\alpha\nabla f(x_k)+\beta(x_k-x_{k+1}) \tag{12.29}$$

with choices $\alpha=\frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$, $\beta=\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$.

12.4 Exercises

Exercise 12.1 (Preliminaries). ☕ *Let $f:\mathcal{X}\to\mathbb{R}$ be twice differentiable. Let $\theta:\mathbb{R}_+\to\mathcal{X}$ be differentiable. Show that*

$$\frac{d}{dt}f(\theta(t))=\langle\nabla f(\theta(t)),\dot\theta(t)\rangle \ ,$$
$$\frac{d}{dt}\nabla f(\theta(t))=\nabla^2 f(\theta(t))\dot\theta(t) \ .$$

Exercise 12.2 (Everything decreases in gradient flow). ☕ *Let $f:\mathbb{R}^d\to\mathbb{R}$ be a convex twice differentiable function. Let $x(t)$ be the gradient flow on f, defined as a solution of $\dot x(t)=-\nabla f(x(t))$.*
Show that $f(x(t))$ decreases.
Show that $\|\nabla f(x(t))\|$ decreases.
Show that $\|x(t)-x^*\|$ decreases for any minimizer x^* of f.

Exercise 12.3. ☕ *Do the results of Exercise 12.2 hold when f is not convex?*

13 Implicit regularization

Resources : Claire Boyer lecture notes: "This chapter heavily relies on the articles Hastie et al. (2019), Bartlett et al. (2021) and the lecture of Matus Telgarsky"

In classical statistical learning, it is often nice to control the norm of the solution in regression, or the margin in classification (Shalev Schwartz 2014 Understanding ML book). It is typically imposed by regularizing the objective function, using an additional square L2 term for example.

However, we have witnessed the advent of deep learning successes, where highly overparameterized, unregularized models which should have been prone to overfitting met stellar success. This has sparked new interest in the phenomenon of *implicit regularisation*, in which, amongst all solutions to an optimization problem, an algorithm always converges to a particular one – in this case, a "simple", "good" minimizer of the empirical risk that generalizes well.

13.1 Implicit bias of GD on least squares

Consider the least square problem:

$$\min_x \frac{1}{2}\|Ax - b\|^2 \tag{13.1}$$

and the iterations of gradient descent on this problem, started at 0:

$$\begin{cases} x_0 = 0 \\ x_{k+1} = x_k - \eta A^\top (Ax_k - b) \end{cases} \tag{13.2}$$

A simple recursion shows that $x_k = \sum_0^k (\mathrm{Id} - \eta A^\top A)^t \eta A^\top b$; if $\eta < 2/\|A\|^2$, the matrix series (called von Neumann series) converges[14], to $(\mathrm{Id} - (Id - \eta A^\top A)^\dagger)$ and so the iterates converge to $A^\dagger b$.

This solution is remarkable: it is the minimum norm solution to $A^\top Ax = A^\top b$[15].

There are many ways to show it (it can even be considered the definition of the pseudo inverse operator, I think). There a geometrical proofs and a nice proof by going to the dual TODO exercise.

Therefore, amongst all possible solutions to the linear system $A^\top Ax = A^\top b$, gradient descent "unknowingly" converges to the minimal norm one, without regularization.

What if one is interested in solutions that are minimal in other sense, e.g. minimize another functional than the Euclidean norm?

13.2 "Implicit" bias of mirror descent

J smooth and α-strongly convex, implicit bias of mirror descent on least squares? Mirror descent iterations:

[14] in the same way $\sum_0^{+\infty} \rho^k = (1-\rho)^{-1}$

[15] recall that in finite dimension there always exists such a solution, even when there is no solution to $Ax = b$; see Exercise 1.1

Proposition 13.1. *Consider the mirror descent iterations on least squares with potential smooth and strongly convex potential J:*

$$\nabla J(x_{k+1}) = \nabla J(x_k) - \eta A^*(Ax_k - b)$$

Then the iterates converge to the J minimal solution:

$$x_k \to \operatorname*{argmin}_x J(x) \quad s.t. \quad Ax = b \tag{13.3}$$

Proof. Explicit "min-J solution" problem and its dual:

$$\text{Primal}: \quad \hat{x} = \operatorname*{argmin}_x J(x) = j(x) + \tfrac{\alpha}{2}\|x\|_2^2 \qquad \text{s.t.} \quad Ax = b \tag{13.4}$$

$$\text{Dual}: \qquad \hat{\theta} = \operatorname*{argmin}_\theta J^*(-A^*\theta) + \langle b, \theta \rangle \tag{13.5}$$

$$\text{Link}: \qquad -A^*\hat{\theta} = \alpha\hat{x} + \nabla j(\hat{x}) \qquad \text{aka} \quad \hat{x} = \operatorname{prox}_{\frac{1}{\alpha}j}(-A^*\hat{\theta}/\alpha) \tag{13.6}$$

In fact, we only need strict convexity of J (in which case the interpretation as a proximal step for the primal variable x does not hold). J strictly convex makes $J*$ differentiable. J strongly convex so J^* smooth. Actually J^* Moreau envelope of j^*:

$$\begin{aligned} J^* &= (j + \tfrac{\alpha}{2}\|\cdot\|^2)^* \\ &= j^* \,\square\, \tfrac{1}{2\alpha}\|\cdot\|^2 \\ &= \inf_y j^*(y) + \tfrac{1}{2\alpha}\|\cdot - y\|^2 \quad \text{(Moreau envelope)} \end{aligned}$$

Since J is strongly convex, the dual is smooth and you can perform gradient descent on it. The gradient of J* writes as a prox in this case. Dual is smooth, do gradient descent on it:

$$\begin{aligned} \theta_{k+1} &= \theta_k - \eta[-A\nabla J^*(-A^*\theta_k) + b] \\ &= \theta_k - \eta[-A \operatorname{prox}_{\frac{1}{\alpha}j}(-A^*\theta_k) + b] \quad \text{(envelope theorem + Moreau decomposition)} \end{aligned}$$

Envelope theorem:

$$\begin{aligned} \nabla J^*(x) &= \tfrac{1}{\alpha}(x - \operatorname{prox}_{\alpha j}(x)) \\ &= \operatorname{prox}_{\frac{1}{\alpha}j}\left(x/\alpha\right) \quad \text{(Moreau decomposition formula)} \end{aligned}$$

The dual iterates θ_k solve the dual, the final step is to map them back to the primal using the primal-dual link equation. Inspired by the link equation, let

$$x_k \triangleq \operatorname{prox}_{\frac{1}{\alpha}j}(-A^*\theta_k/\alpha), \text{ meaning } -A^*\theta_k = \alpha x_k + \nabla j(x_k) = \nabla J(x_k)$$

Plugging back x_k in dual GD iterates gives mirror descent:

$$\begin{aligned} -A^*\theta_{k+1} &= -A^*\theta_k + \eta A^*[-A\nabla J^*(-A^*\theta_k) + b] \\ \nabla J(x_{k+1}) &= \nabla J(x_k) - \eta A^*(Ax_k - b) \quad \text{(mirror descent)} \end{aligned}$$

Since $\theta_k \to \hat{\theta}$, $x_k \to \hat{x}$. □

13.3 Logistic regression on separable data

Motivation: controlling norm avoids overfitting, controls *stability* of solution. Typical example: Tikhonov, bound singular value.

$$(A^t A)^\dagger A^\top b = \sum_1^R \frac{1}{\sigma_i} v_i u_i^\top b$$

vs

$$(A^t A + \lambda \mathrm{Id})^\dagger A^\top b = \sum_1^R \frac{\sigma_i}{(\sigma_i + \lambda)^2} v_i u_i^\top b$$

Iterative regularization Engl 1996
Implicit bias of GD on OLS. Generalization to Mirror descent
TODO: give SVD as homework

1. the geometrical view with SVD for OLS: projection onto Span $A^\top$

Ref off the convex path: http://www.offconvex.org/2020/11/27/reg_dl_not_norm/

13.4 Matrix factorization: can everything be explained in terms of norms?

Deep linear nets for matrix factorization, ref http://www.offconvex.org/2019/07/10/trajectories-linear-nets/

Neyshabur thesis on implicit bias in DL: https://arxiv.org/pdf/1709.01953.pdf

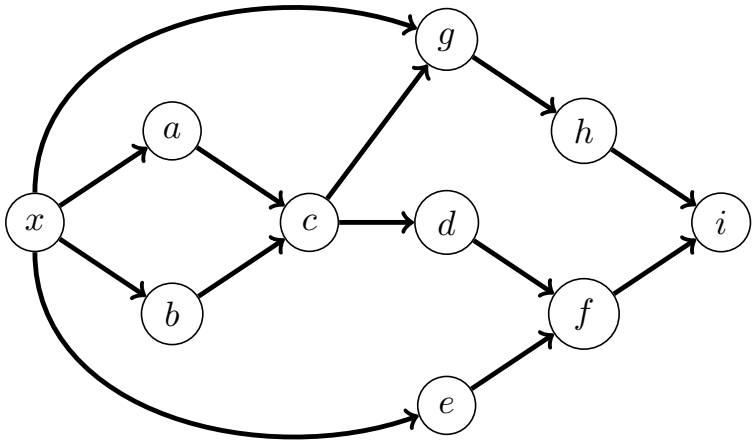

Figure 6: Directed acyclic graph representing the dependency graph of Algorithm 1

14 Automatic and implicit differentiation

First-order optimization methods require computing gradients of given functions. So far, all the functions were simple (least squares, logistic regression) and their gradient could be computed by hand.

We now turn to examples where the computation requires more advanced tools.

14.1 Forward and backward automatic differentiation

Note: AD acts on programs, not on functions.

Consider the following scalar function:

$$f(x) = \exp(x^2 - 3x)\sin(x) + \cos(x^2 - 3x)x \tag{14.1}$$

It can be implemented through the following program in Algorithm 1, which is itself representable by a Directed Acyclic Graph (DAG) in Figure 6.

Algorithm 1 Program implementing f

$a = x^2$
$b = 3x$
$c = a + b$
$d = \exp(c)$
$e = \sin(x)$
$f = de$
$g = \cos(c)$
$h = gx$
$i = hf$

In this section, we will introduce three ways to compute gradient/Jacobians of loss functions

Tuning the weights of a neural network requires computing the Jacobian of the loss with respect to various parameters. Automatic differentiation, in particular Autodiff, backpropagation

Autodiff for a one layer NN:

$$f(x) = W_2\sigma(W_1 x) \tag{14.2}$$

Technique: vectorize everything and devectorize. Code snippet. Emphasize vector Jacobian product and Jacobian vector product (directional derivatives)

14.2 Implicit differentiation for bilevel optimization

$$\min_y F(x^*(y), y) \quad \text{s.t.} \quad x^*(y) = \operatorname*{argmin}_x F(x, y) \tag{14.3}$$

One could think of performing gradient descent on F, requiring the computation of the *hypergradient*:

$$\nabla_y F(x^*(y), y) = \nabla_1 F(x^*(y))^\top \mathcal{J}x^*(y) + \nabla_2 F(x^*(y), u) \ . \tag{14.4}$$

The main challenge is the differentiation of the inner problem solution with respect to y. We can write the first-order optimality condition for the inner problem, and differentiate with respect to y:

$$\nabla_1 F(x^*(y), y) = 0 \tag{14.5}$$

$$\nabla_1^2 F(x^*(y), y)\mathcal{J}x^*(y) + \nabla_{12}^2 F(x^*(y), y) = 0 \tag{14.6}$$

$$\mathcal{J}x^*(y) = -[\nabla_1^2 F(x^*(y), y)]^{-1}\nabla_{12}^2 F(x^*(y), y) \tag{14.7}$$

Need to solve a linear system, of which size? The number of parameters. What if G is not smooth? Alternative: differentiate a fixed point scheme. Give example on the Lasso.

15 Variational inequalities

In all this section K is a convex bounded subset of $\mathbb{R}^d$. The function $F : \mathbb{R}^d \to \mathbb{R}^d$ is called an operator.

Definition 15.1 (Strong and weak solutions to variational inequality problems)**.** *We say that $x^* \in \mathbb{R}^d$ is a* strong solution *to the variational inequality problem, or simply to the variational inequality, associated to F if $x^* \in K$ and*

$$\forall y \in K,\ \langle F(x^*), x^* - y\rangle \leq 0\ . \tag{15.1}$$

It is a weak solution *if it belong to K and*

$$\forall y \in K,\ \langle F(y), x^* - y\rangle \leq 0\ . \tag{15.2}$$

In the unconstrained case, a strong solution is simply a point such that $F(x^*) = 0$. Variational inequalities generalize convex (constrained) minimization problems: to solve $\min_C f(x)$ is to find a strong solution to the VI associated to $F = \nabla f$.

Under certain conditions, strong and weak solutions are the same; these conditions involve a special property on F, monotonicity.

Definition 15.2 (Monotone operator)**.** *The operator F is monotone if*

$$\forall x, y,\ \langle F(x) - F(y), x - y\rangle \geq 0\ . \tag{15.3}$$

The most famous class of monotone operators is probably the one composed of the convex functions' gradients (see Exercise 2.9).

Proposition 15.3 (equivalence under monotonicity of F)**.** *Let F be monotone. Then any strong solution to the VI is a weak solution to the VI. The converse is also true if F is continuous and K is convex, a result known as Minty's lemma*[16]*.*

Proof. One direction is easy. To prove that any weak solution is a strong solution, let x^* be a weak solution, and let $y \in K$. Let $t \in]0, 1]$. Since K is convex, $x^* + t(y - x^*) \in K$, and so:

$$\langle F(x^* + t(x^* - y)), x^* - x^* + t(x^* - y)\rangle \leq 0 \tag{15.4}$$
$$t\langle F(x^* + t(x^* - y)), x^* - y\rangle \leq 0 \tag{15.5}$$
$$\langle F(x^* + t(x^* - y)), x^* - y\rangle \leq 0\ . \tag{15.6}$$

Taking the limit as $t \to 0$, by continuity of F, yields that x^* is a strong solution since y was any point. □

Variational inequalities generalize variational optimization problems, but there is no objective anymore.

They handle more complex problems, such as min-max problems arising in game theory.

[16] everything here is in finite dimension, but beware that the result may not hold in infinite dimension. Can you spot where this plays a role in the proof?

Example 15.4 (Rock paper scissors)**.** *The rock-paper-scissors game is described by the payoff matrix* $A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix}$*. If Rock, Paper and Scissors choices are represented by 1, 2 and 3 respectively, then if player 1 plays* i *and player 2 pays* j*, player 1 receives payoff* A_{ij} *(and player 2 receives* $-A_{ij}$*).*

If Player 1 (P1) chooses a random strategy represented by a vector p *in the 3-d simplex* Δ_3 *and Player 2 (P2) chooses* $q \in \Delta_3$*, then the expected payoff of P1 is:*

$$p^\top A q = \sum_{i,j=1}^{3} p_i A_{ij} q_j \ . \tag{15.7}$$

The game for P1 is to solve $\max_p \min_q p^\top A q$*, while for P2 it is to solve* $\max_q \min_p -p^\top A q$*, or equivalently* $\min_q \max_p p^\top A q$*.*

A result, due to von Neumann and reminiscent of Section 7, states that the two objectives are the same:

$$\max_{p\in\Delta_3} \min_{q\in\Delta_3} p^\top A q = \min_{q\in\Delta_3} \max_{p\in\Delta_3} p^\top A q \ . \tag{15.8}$$

More generally, VIs are useful to model optimization problems of the form:

$$\min_u \max_v f(u, v) \ , \tag{15.9}$$

where f is differentiable, convex in u, and concave in v. Indeed, such a problem is equivalent to the VI $F = (\nabla_u f, -\nabla_v f)$.

To solve Problem (9), one may be tempted to perform gradient descent on u and gradient ascent on v simultaneously:

$$\begin{cases} u_{k+1} = u_k - \eta \nabla_u f(u_k) \\ v_{k+1} = v_k + \eta \nabla_v f(v_k) \ . \end{cases} \tag{15.10}$$

but unfortunately, this fails to converge even in very simple cases.

Example 15.5. *Consider the 2D problem* $\min_{u\in\mathbb{R}} \max_{v\in\mathbb{R}} uv$*, whose solution is* $(0, 0)$*. The iterates* (15.10) *on this problem read:*

$$\begin{cases} u_{k+1} = u_k - \eta v_k \\ v_{k+1} = v_k + \eta u_k \ . \end{cases} \tag{15.11}$$

It is easy to check that $\|(u_{k+1}, v_{k+1})\| = (1 + \eta)\|(u_k, v_k)\|$*. Hence, the iterates will always diverge (worse: they get further away from the solution at every iteration). This is visible on Figure 7.*

We must therefore come up with a better method!

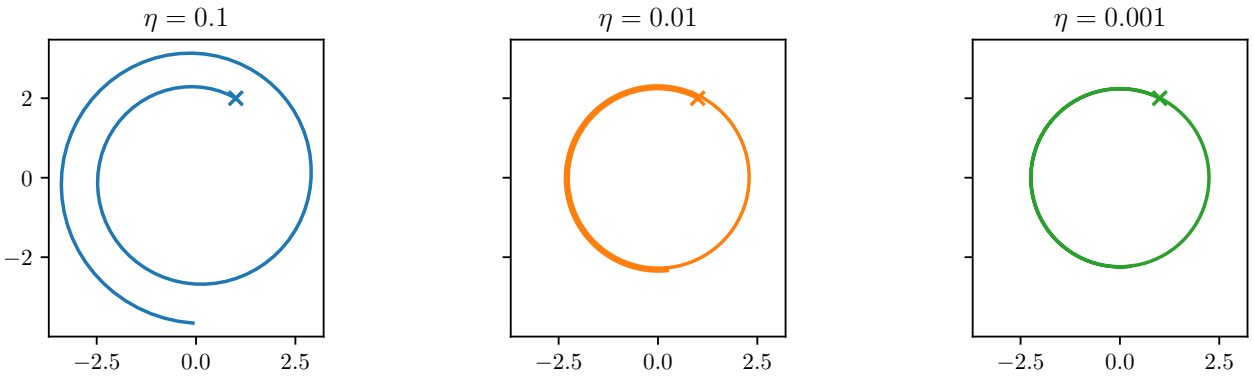

Figure 7: Gradient method for the saddle point problem $\min_{u\in\mathbb{R}} \max_{v\in\mathbb{R}} uv$, initialized at $(1, 1)$. No matter how small η, the iterates never converge. In fact, they always go to infinity.

15.1 Solving VIs: the extragradient method

Suppose we want to solve a VI with an iterative algorithm. What is a relevant measure of error, since there is no objective involved? Guided by the definition of a weak solution, we will use the following value to quantify the quality of x:

$$\sup_{y\in K} \langle F(y), x - y\rangle. \tag{15.12}$$

Proposition 15.6 (Convergence rate of the extragradient method)**.** *The extragradient method iterates are given by*[17]*:*

$$\begin{cases} x_k = z_{k-1} - \eta F(z_{k-1}) \\ z_k = z_{k-1} - \eta F(x_k) = z_{k-1} - \eta F(z_{k-1} - \eta F(z_{k-1})) \end{cases} \tag{15.13}$$

has the following convergence rates on the ergodic iterates $\bar{x}_t = \frac{1}{t}\sum_1^t x_k$*:*

- *if* F *is bounded in norm by* G *and* $\eta = ...$*, err* $\bar{x}_t \leq TODO$
- *if* F *is* β *Lipschitz and* $\eta < 1/\beta$*, rate is* $/T$

Proof. In the whole proof, $y \in K$ is fixed. Let $t \in \mathbb{N}$, $k \in [\![1, t]\!]$ By monotonicity of F,

$$\langle F(y), x_k - y\rangle \leq \langle F(x_k), x_k - y\rangle \ . \tag{15.14}$$

Now, we want to use the definitions of x_k and z_k, and so we must make $x_{k+1} - x_k$ and TODO appears. We thus write:

$$\begin{aligned} \langle F(x_k), x_k - y\rangle &= \langle F(x_k), z_k - y\rangle + \langle F(z_{k-1}), x_k - z_k\rangle + \langle F(x_k) - F(z_{k-1}), x_k - z_k\rangle \\ &= \frac{1}{\eta}\langle z_{k+1} - z_k, z_k - y\rangle + \frac{1}{\eta}\langle z_{k-1} - x_k, x_k - z_k\rangle + \langle F(x_k) - F(z_{k-1}), x_k - z_k\rangle \ . \end{aligned} \tag{15.15}$$

[17] note: we cannot talk about extragradient *descent* since there is nothing being minimized here

We use the parallelogram identity on the two first terms of the right-hand side:

$$\langle z_{k+1} - z_k, z_k - y\rangle = \frac{1}{2}\left(\|z_{k+1} - y\|^2 - \|z_{k+1} - z_k\|^2 - \|z_k - y\|^2\right) \tag{15.16}$$

$$\langle z_{k-1} - x_k, x_k - z_k\rangle = \frac{1}{2}\left(\|z_{k+1} - z_k\| + \|z_{k+1} - x_k\|^2 + \|x_k - z_k\|^2\right) \tag{15.17}$$

Summing, the second term of the RHS of the first line and the first term of the RHS of the second line cancel out. So the RHS in (15.15) is equal to:

$$\frac{1}{2\eta}\left(\|z_{k+1} - y\|^2 - \|z_k - y\|^2\right) - \frac{1}{2\eta}\left(\|z_{k+1} - x_k\|^2 + \|x_k - z_k\|^2\right) + \langle F(x_k) - F(z_{k-1}), x_k - z_k\rangle \tag{15.18}$$

The first term will telescope when we sum over k. It remains to handle the last one, involving F.

Bounded case: by applying Cauchy-Schwartz inequality, the triangular inequality, and Young inequality, we obtain

$$\langle F(x_k) - F(z_{k-1}), x_k - z_k\rangle \leq 2G\|x_k - z_k\| \leq 2\eta G^2 + \frac{\|x_k - z_k\|^2}{2\eta} \ . \tag{15.19}$$

Plug back in RHS, two terms cancel out. Summing over k from 1 to t, dividing by t and choosing η accordingly concludes the proof.

Lipschitz case:

$$\langle F(x_k) - F(z_{k-1}), x_k - z_k\rangle \leq \beta\|x_k - z_{k-1}\|\|x_k - z_k\| \leq \frac{\beta}{2}\|x_k - z_{k-1}\|^2 + \frac{\beta}{2}\|x_k - z_k\|^2 \ . \tag{15.20}$$

(15.18) is thus upper bounded by

$$\frac{1}{2\eta}\left(\|z_{k+1} - y\|^2 - \|z_k - y\|^2\right) + \left(\frac{1}{2\eta} - \frac{\beta}{2}\right)\left(\|z_{k+1} - x_k\|^2 + \|x_k - z_k\|^2\right) \ . \tag{15.21}$$

The second term is negative provided $\eta \leq 1/\beta$; summing, telescoping and dividing by t concludes similarly. □

16 Additional resources

Bansal and Gupta (2017) for a review of potential-based convergence proofs.

References

Baptiste Goujaud, Adrien Taylor, and Aymeric Dieuleveut. Provable non-accelerations of the heavy-ball method. *arXiv preprint arXiv:2307.11291*, 2023.

Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *2015 European control conference (ECC)*, pages 310–315. IEEE, 2015.

Vassilis Apidopoulos, Nicolò Ginatta, and Silvia Villa. Convergence rates for the heavy-ball continuous dynamics for non-convex optimization, under polyak–łojasiewicz condition. *Journal of Global Optimization*, 84(3):563–589, 2022.

J-F Aujol, Ch Dossal, and A Rondepierre. Convergence rates of the heavy-ball method under the łojasiewicz property. *Mathematical Programming*, 198(1):195–254, 2023.

Hedy Attouch and Jalal Fadili. From the ravine method to the nesterov method and vice versa: a dynamical system perspective. *SIAM Journal on Optimization*, 32(3):2074–2101, 2022.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Ernest K Ryu and Stephen Boyd. Primer on monotone operator methods. *Appl. comput. math*, 15(1):3–43, 2016.

Ernest K Ryu and Wotao Yin. *Large-scale convex optimization: algorithms & analyses via monotone operators*. Cambridge University Press, 2022.

Patrick L Combettes and Jean-Christophe Pesquet. Fixed point strategies in data science. *IEEE Transactions on Signal Processing*, 69:3878–3905, 2021.

Arian Berdellima. On a notion of averaged operators in cat (0) spaces. *arXiv preprint arXiv:2010.05726*, 2020.

Nobuhiko Ogura and Isao Yamada. Non-strictly convex minimization over the fixed point set of an asymptotically shrinking nonexpansive mapping. 2002.

Patrick L Combettes. Fejér-monotonicity in convex optimization. *Encyclopedia of optimization*, 2:106–114, 2001.

Nikhil Bansal and Anupam Gupta. Potential-function proofs for first-order methods. *arXiv preprint arXiv:1712.04581*, 2017.