4AI 11 – Optimization for Machine Learning

Pascal Bianchi, Olivier Fercoq, Anne Sabourin, Radu Dragomir

December 9, 2024

Contents

Chapter 1

Introduction: Optimization, machine learning and convex analysis

1.1 Optimization problems in Machine Learning

Training a machine learning model usually requires solving a minimization problem. We search for a model whose error on the training data is as small as possible. We also want the model to perform well on new data: to do so, we often add a regularization term to prevent *overfitting.* In general, non-convex problems are difficult, whereas convex ones are easier to solve. Here, we are mostly going to focus on convex problems, as they benefit from strong theoretical guarantees. However, many of the algorithms and tools that we develop can also be applied to the non-convex setting.
Let us give some examples of optimization problems that arise in machine learning.

Example 1.1.1 (Linear regression). We have n vectors $z_1 \dots z_n \in \mathbb{R}^p$ (each coordinate is called a *feature*) and n target values $y_1 \dots y_n \in \mathbb{R}$. We want to find a vector $x \in \mathbb{R}^p$ such that $x^\top z_i \approx y_i$ for each $i = 1 \dots n$. To do so, we can minimize the sum of squared errors

$$\sum_{i=1}^{n} (z_i^\top x - y_i)^2 = \|Zx - Y\|^2, \quad \text{where } Z = \begin{bmatrix} z_1^\top \\ \vdots \\ z_n^\top \end{bmatrix} \in \mathbb{R}^{n\times p},\, Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n.$$

There are several approaches for finding a good x.

(a) **Ordinary Least Squares:** we simply minimize the sum of squared errors and solve

$$\min_{x\in\mathbb{R}^p} \|Zx - Y\|^2,$$

(b) **Ridge:** sometimes, there are many solutions to the problem above. For statistical reasons, we prefer one which has a small norm: it is better for generalization on unseen data. We can then choose to penalize the problem with the ℓ_2 norm, and we get the ridge (or Tikhonov) regularized regression:

$$\min_{x\in\mathbb{R}^p} \|Zx - Y\|^2 + \lambda\|x\|_2^2,$$

(c) **Lasso:** if we penalize the ℓ_1 norm instead, this enforces the solution to be *sparse*, which can be desirable if we want the model to depend on a small number of features. The optimization problem is then

$$\min_{x\in\mathbb{R}^p} \|Zx - Y\|^2 + \lambda\|x\|_1,$$

Example 1.1.2 (Linear classification).
We want to predict a binary variable $y \in \{-1,+1\}$ from *features* $z \in \mathbb{R}^p$. The simplest method for performing this task is linear classification. Linear classifiers are functions of the form

$$h : z \mapsto \operatorname{sign}(\langle x, z\rangle + x_0) \qquad (x \in \mathbb{R}^p, x_0 \in \mathbb{R}).$$

A classifier h is thus determined by a vector $\mathbf{x} = (x, x_0)$ in $\mathbb{R}^{p+1}$. The vector x is the normal vector to an hyperplane which separates the space into two regions, inside which the predicted labels are respectively "+1" and "−1".
To train the classifier, we use a dataset $\mathcal{D} = \{(z_1, y_1), \dots, (z_n, y_n)\}$, $y_i \in \{-1, 1\}$, $z_i \in \mathbb{R}^p$, where the z_i's are the data's *features* and the y_i's are the labels which represent the class of observation i. The samples are obtained by independent realizations of a vector $(z, y) \sim \mathbb{P}$, of unknown distribution $\mathbb{P}$.
The goal is to learn a classifier which, in average, is not wrong by much: that means that we want $\mathbb{P}(h(Z) = Y)$ to be as big as possible.
To quantify the classifier's error/accuracy, the reference loss function is the '0-1 loss':

$$L_{01}(\mathbf{x}, z, y) = \begin{cases} 0 & \text{if } -y\left(\langle x, z\rangle + x_0\right) \leq 0 \quad (h(z) \text{ and } y \text{ of same sign}), \\ 1 & \text{otherwise.} \end{cases}$$

In general, the implicit goal of machine learning methods for supervised classification is to solve (at least approximately) the following problem:

$$\min_{\mathbf{x} \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} L_{0,1}(\mathbf{x}, z_i, y_i) \tag{1.1.1}$$

i.e. to minimize the *empirical risk.*
As the cost L is not convex in $\mathbf{x}$ (and not even continuous), the problem (1.1.1) is *hard.* Classical machine learning methods choose to minimize instead a function that is similar to the objective (1.1.1): we replace the cost 0-1 by a *convex substitute*, and then to add a penalty term which penalizes "complexity" of x, so that we find a solution that generalizes well on unseen data. More precisely, the problem to be solved numerically is

$$\min_{x \in \mathbb{R}^p, x_0 \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \varphi(-y_i(x^\top z_i + x_0)) + \lambda \mathcal{P}(x), \tag{1.1.2}$$

where $\mathcal{P}$ is the penalty and φ is a convex substitute to the cost 0-1.
Different choices of penalties and convex subsitutes are available, yielding a range of methods for supervised classification :

- For $\varphi(u) = \max(0, 1 + u)$ (Hinge loss), $\mathcal{P}(x) = \|x\|^2$, we get Support Vector Machines (SVM):

$$\min_{x \in \mathbb{R}^p, x_0 \in \mathbb{R}} \sum_{i=1}^{n} \max(0, 1 - y_i(x^\top z_i + x_0)) + \lambda \|x\|^2.$$

- In the separable case (*i.e.* when there is a hyperplane that separates the two classes), introduce the "convex indicator function" (also called *characteristic function*),

$$\iota_A(x) = \begin{cases} 0 & \text{if } x \in A, \\ +\infty & \text{if } x \in A^c, \end{cases} \qquad (A \subset \mathcal{X})$$

and set

$$\varphi(u) = \iota_{\mathbb{R}^-}(u).$$

The solution to the problem is the maximum margin hyperplane:

$$\begin{aligned}
&\min_{x\in\mathbb{R}^p, x_0\in\mathbb{R}} \sum_{i=1}^{n} \iota(-y_i(x^\top z_i + x_0)) + \lambda\|x\|^2 \\
&= \min_{x\in\mathbb{R}^p, x_0\in\mathbb{R}} \{\lambda\|x\|^2, \text{ st: } -y_i(x^\top z_i + x_0) \le 0, \forall i, 1 \le i \le n\}
\end{aligned}$$

- For $\varphi(u) = \log(1+\exp(u))$ (Logistic loss) and $\mathcal{P}(x) = \|x\|^2$, we get logistic regression:

$$\min_{x\in\mathbb{R}^p, x_0\in\mathbb{R}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i(x^\top z_i + x_0))\right) + \lambda\|x\|^2.$$

To summarize, the common denominator of all these versions of example 1.1.2 is as follows:

- The statistical risk of a classifier x is defined by $J(x) = \mathbb{E}_{(z,y)\sim\mathbb{P}}[L_{01}(x,z,y)]$. It is the expectation of the 0-1 error on the *true* distribution $\mathbb{P}$. We are looking for x which minimizes J.

- $\mathbb{P}$ is unknown, and therefore so is J. However, a training sample $\mathcal{D} \sim \mathbb{P}$ is available. Therefore, we approximate the true statistical by the **empirical risk**

$$J_n(x) = \frac{1}{n}\sum_{i=1}^{n} L_{01}(x, z_i, y_i)$$

- The cost L_{01} is replaced by a convex surrogate L_φ, so that we obtain a convex function

$$J_{n,\varphi}(x) = \frac{1}{n}\sum_{i=1}^{n} L_\varphi(x, z_i, y_i)$$

- To prevent overfitting, we add a convex penalty term $\mathcal{P}$. In the end, the problem to be solved is

$$\min_{x\in\mathcal{X}} J_{n,\varphi}(x) + \lambda\mathcal{P}(x). \tag{1.1.3}$$

In the remaining of the course, the focus is on that last point: how to solve the convex minimization problem (1.1.3)?

Example 1.1.3 (Hyperparameter optimization). A natural question when considering Problem (1.1.3) is: what value should λ take? One of the goals of the regularization is to improve the generalization performance of the model. We want the model to behave well on new data points $(z_j^{\text{valid}}, y_j^{\text{valid}})_{1\le j\le m}$ where we will evaluate the parameters returned by the algorithm that solves (1.1.3).

We can formalize this question into a bilevel optimization problem, that is an optimization problem that involves (1.1.3) as an inner problem.

$$\min_{\lambda \geq 0} \frac{1}{m} \sum_{j=1}^{m} L(\hat{x}^{(\lambda)}, z_j^{\text{valid}}, y_j^{\text{valid}})$$

$$\hat{x}^{(\lambda)} \in \arg\min_{x} \frac{1}{n} \sum_{i=1}^{n} L_\varphi(x, z_i, y_i) + \lambda \mathcal{P}(x)$$

We will then solve a sequence of convex optimization problems indexed by λ and solving each of them fast will matter for the overall performance.

Example 1.1.4 (Feed-forward neural network for classification). For more difficult classification problems, we might want to consider models which are more expressive than linear functions. A feed-forward neural network (or *multilayer perceptron*) with L layers is a nonlinear function on $\mathbb{R}^p$ of the form

$$g(z; \theta) = g_L \circ g_{L-1} \circ \cdots \circ g_1(z).$$

z is the input and θ is the set of all parameters of the neural network. For each $i = 1 \dots L$ the i-th layer is described by the function

$$g_i(u) = \sigma^{(i)}(W^{(i)} u + b^{(i)}), \quad u \in \mathbb{R}^{d_{i-1}}$$

where

- $W^{(i)} \in \mathbb{R}^{d_i \times d_{i-1}}$ *weight matrix* and $b^{(i)} \in \mathbb{R}^{d_i}$ the *bias* of layer i,
- d_i is the *width*, or number of *neurons*, of layer i (and we set $d_0 = p$),
- $\sigma^{(i)}$ is a nonlinear activation function applied componentwise. Classical examples are ReLU ($\sigma^{(i)}(u) = \max(u, 0)$) or $\sigma^{(i)} = \tanh$.

The parameters of the neural network are $\theta = \{W^{(1)}, b^{(1)}, \dots W^{(L)}, b^{(L)}\}$. For given values of θ, the neural network defines a nonlinear classifier as the function

$$h(z) = \text{sign}(g(z; \theta)).$$

We want to find the best parameters θ for a given classification. As in Example 1.1.2, we train the classifier on a dataset $\{(z_i, y_i)\}_{1 \leq i \leq n}$ by minimizing the empirical loss

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \varphi\big(-y_i \, g(z_i; \theta)\big) + \lambda \mathcal{P}(\theta),$$

where φ is a convex surrogate of the 0-1 loss and $\mathcal{P}$ an additional regularization term. Because of the nonlinearity of function $g(\cdot; \theta)$ and the large size of θ, this minimization problem is much more challenging than that of linear classification. In particular, it is not **convex**.

1.2 General formulation of the problem

In this course, we only consider optimization problems which are defined on a finite dimension space $\mathcal{X} = \mathbb{R}^n$. These problems can be written, without loss of generality, as follows:

$$\begin{gathered} \min_{x \in \mathcal{X}} f(x) \\ s.t. \text{ (such that / under constraint that)} \\ g_i(x) \leq 0 \text{ for } 1 \leq i \leq p, \quad F_i(x) = 0 \text{ for } 1 \leq i \leq m\,. \end{gathered} \tag{1.2.1}$$

The function f is the *objective function* (or *target*),
the vector

$$C(x) = (g_1(x), \dots, g_p(x), F_1(x), \dots, F_m(x))$$

is the (functional) constraint vector.
The region

$$K = \{x \in \mathcal{X} : \; g_i(x) \leq 0, \; 1 \leq i \leq p, \quad F_i(x) = 0, \, 1 \leq i \leq m\}$$

is the set of *feasible* points.

- If $K = \mathbb{R}^n$, this is an *unconstrained* optimization problem.
- Problems where $p \geq 1$ and $m = 0$, are referred to as *inequality contrained* optimization problems.
- If $p = 0$ and $m \geq 1$, we speak of *equality contrained* optimization.
- When f and the constraints are regular (differentiable), the problem is called *differentiable* or *smooth*.
- If f or the constraints are not regular, the problem is called *non-differentiable* or *non-smooth*.
- If f and the contraints are convex, we have a *convex* optimization problem (more details later).

Solving the general problem (1.2.1) consists in finding

- a minimizer $x^* \in \arg\min_K f$ (if it exists, *i.e.* if $\arg\min_K f \neq \emptyset$),
- the *value* $f(x^*) = \min_{x \in K} f(x)$,

We can rewrite the constrained problem as an unconstrained problem, thanks to the infinite indicator function ι introduced earlier. Let's name g and (resp) F the vectors of the inequality and (resp) equality contraints.
For $x, y \in \mathbb{R}^n$, we write $x \preceq y$ if $(x_1 \leq y_1, \dots, x_n \leq y_n)$ and $x \npreceq y$ otherwise. The problem (1.2.1) is equivalent to :

$$\min_{x \in E} f(x) + \iota_{g \preceq 0, F=0}(x) \tag{1.2.2}$$

Let's notice that, even if the initial problem is smooth, the new problem isn't anymore !

1.3 Algorithms

In this section, we assume for simplicity that the problem is unconstrained.

Approximate solutions. Most of the time, Problem (1.2.1) cannot be solved analytically. However, numerical algorithms can provide an approximate solution. Finding an ϵ-approximate solution (**ϵ-solution**) consists in finding $\hat{x} \in K$ such that, if the "true" minimum x^* exists, we have

- $\|\hat{x} - x^*\| \leq \epsilon$,

  and/or

- $|f(\hat{x}) - f(x^*)| \leq \epsilon$.

"Black box" model A standard framework for optimization is the **black box**. That is, we want to optimize a function in a situation where:

- The algorithm can only access f by successive calls to an *oracle* $\mathcal{O}(x)$.
  Typically, $\mathcal{O}(x) = f(x)$ (0-order oracle) or $\mathcal{O}(x) = (f(x), \nabla f(x))$ (1-order oracle), or $\mathcal{O}(x) = (f(x), \nabla f(x), \nabla^2 f(x))$ ($2nd$-order oracle), etc.
- At iteration k, the algorithm only has the information $\mathcal{O}(x_1), \ldots, \mathcal{O}(x_k)$ as a basis to compute the next point x_{k+1}.
- The algorithm stops at time k if a criterion $T_\epsilon(x_k)$ is satisfied: the latter ensures that x_k is an ϵ-solution.

Performance of an algorithm Performance is measured in terms of computing resources needed to obtain an approximate solution.
This obviously depends on the considered problem. A **class of problems** is:

- A class of objective functions (regularity conditions, convexity or other)
- A condition on the starting point x_0 (for example, $\|x - x_0\| \leq R$)
- An oracle.

Definition 1.3.1 (oracle complexity)**.** The **oracle complexity** of an algorithm $\mathcal{A}$, for a class of problems C and a given precision ϵ, is the minimal number $N_\mathcal{A}(\epsilon)$ such that, for all objective functions and any initial point $(f, x_0) \in C$, we have:

$$N_\mathcal{A}(f, \epsilon) \leq N_\mathcal{A}(\epsilon)$$

where : $N_\mathcal{A}(f, \epsilon)$ is the number of calls to the oracle that are needed for $\mathcal{A}$ to give an ϵ-solution. The oracle complexity, as defined here, is a *worst-case* complexity. The computation time depends on the oracle complexity, but also on the number of required arithmetical operations at each call to the oracle. The total number of arithmetic operations to achieve an ϵ-solution in the worst case, is called *arithmetic complexity.* In practice, it is the arithmetic complexity which determines the computation time, but it is easier to prove bounds on the oracle complexity .

1.4 A first glance at convergence proofs

In this course, we are going to study several optimization algorithms. Suppose we want to minimise a convex and C^2 function $f : \mathbb{R} \to \mathbb{R}$:

$$\min_{x \in \mathbb{R}} f(x)$$

We shall assume that there exists $\mu > 0$ and $L > 0$ such that for all x, $\mu \leq f''(x) \leq L$. A simple algorithm to solve such a problem is the gradient descent algorithm, also called steepest descent:

$$\begin{aligned} & x_0 \in \mathbb{R} \\ & \forall k \in \mathbb{N}, x_{k+1} = x_k - \gamma f'(x_k) \end{aligned}$$

where $\gamma > 0$ is a real number call step size.
We can study its convergence using Picard's fixed point theorem.

Theorem 1.4.1. *If $T : \mathbb{R}^d \to \mathbb{R}^d$ satisfies that $\exists 0 < \rho < 1, \forall x \in \mathbb{R}^d, \forall y \in \mathbb{R}^d$,*

$$\|T(x) - T(y)\| \leq \rho \|x - y\|$$

then T has a unique fixed point x^* such that $x^* = T(x^*)$.
Moreover, every sequence of the form $x_{k+1} = T(x_k)$ converges to x^* with a linear convergence rate given by

$$\|x_k - x^*\| \leq \rho^k \|x_0 - x^*\|.$$

Proposition 1.4.1. *If f is C^2 and there exists $\mu > 0$ and $L > 0$ such that for all x, $\mu \leq f''(x) \leq L$ and if $0 < \gamma < 2/L$, then the gradient descent algorithms converges to x^* such that $f'(x^*) = 0$.*

Proof. Let us define $T : x \mapsto x - \gamma f'(x)$.
$T'(x) = 1 - \gamma f''(x) \in [1 - \gamma L, 1 - \gamma\mu]$, so by the mean value theorem, T is Lipschitz continuous with constant $\max(|1 - \gamma\mu|, |1 - \gamma L|)$. This constant is smaller than 1 as soon as $\gamma < 2/L$ so that T is a contraction. We can thus apply Picard's fixed point theorem.
The algorithm $x_{k+1} = T(x_k)$, which is exactly the gradient descent algorithm, converges to x^* such $T(x^*) = x^*$. Moreover, $T(x^*) = x^* \Leftrightarrow x^* - \gamma f'(x^*) = x^* \Leftrightarrow f'(x^*) = 0$. □

1.5 Preview of the rest of the course

A natural idea to solve general problem (1.2.1) is to start from an arbitrary point x_0 and to propose the next point x_1 in a region where f "has a good chance" to be smaller.
If f is differentiable, one widely used method is to follow "the line of greatest slope", i.e. move in the direction given by $-\nabla f$.
What's more, if there is a local minimum x^*, we then have $\nabla f(x^*) = 0$. So a similar idea to the previous one is to set the gradient equal to zero.
Here we have made implicit assumptions of regularity, but in practice some problems can arise.

- Under which assumptions is the necessary condition '$\nabla f(x) = 0$' sufficient for x to be a local minimum?
- Under which assumptions is a local minimum a global one?
- What if f is not differentiable?
- How should we proceed when E is a high-dimensional space?
- What if the new point x_1 leaves the admissible region K?

The appropriate framework to answer the first two questions is convex analysis. The lack of differentiability can be bypassed by introducing the concept of *subdifferential. Duality* methods solve a problem related to ((1.2.1)), called *dual problem.* The dual problem can often be easier to solve (*ex:* if it belongs to a space of smaller dimension). Typically, once the dual solution is known, the primal problem can be written as a unconstrained problem that is easier to solve than the initial one. For example, *proximal* methods can be used to solve constrained problems.

N.B The exercises and chapters marked with * are for students interested in technical details and can be skipped in a first read.

To go further ...
A panorama in Boyd and Vandenberghe (2009), chapter 4, more rigor in Nesterov (2004)'s introduction chapter.

Chapter 2

Elements of convex analysis

2.1 Convexity

2.1.1 Convex sets and convex functions

Definition 2.1.1 (Convex set)**.** A set $K \subset$ is **convex** if

$$\forall (x,y) \in K^2, \forall t \in [0,1], \quad t\,x + (1-t)\,y \in K.$$

Exercise 2.1.1.

1. Show that a ball, a vector subspace or an affine subspace of $\mathbb{R}^n$ are convex.

2. Show that any intersection of convex sets is convex.

In constrained optimization problems, it is useful to define cost functions with value $+\infty$ outside the admissible region. For all $f : \mathcal{X} \to [-\infty, +\infty]$, the *domain* of f, denoted by $\mathrm{dom}(f)$, is the set of points x such that $f(x) < +\infty$.
A function f is called **proper** if $\mathrm{dom}(f) \neq \emptyset$ (i.e $f \not\equiv +\infty$) and if f *never* takes the value $-\infty$.

Definition 2.1.2. Let $f : \mathcal{X} \to [-\infty, +\infty]$. The **epigraph of** f, denoted by $\mathrm{epi}\, f$, is the subset of $\mathcal{X} \times \mathbb{R}$ defined by:

$$\mathrm{epi}\, f = \{(x,t) \in \mathcal{X} \times \mathbb{R} : \quad t \geq f(x)\ \}.$$

Definition 2.1.3 (Convex function)**.** $f : \mathcal{X} \to [-\infty, +\infty]$ is **convex** if its epigraph is convex.

Proposition 2.1.1. *A function $f : \mathcal{X} \to [-\infty, +\infty]$ is convex if and only if*

$$\forall (x,y) \in \mathcal{X}^2,\ \forall t \in (0,1), \quad f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

Proof. Assume that f satisfies the inequality. Let (x,u) and (y,v) be two points of the epigraph: $u \geq f(x)$ and $v \geq f(y)$. In particular, $(x,y) \in \mathrm{dom}(f)^2$. Let $t \in]0,1[$. The inequality implies that $f(tx + (1-t)y) \leq tu + (1-t)v$. Thus, $t(x,u) + (1-t)(y,v) \in \mathrm{epi}(f)$, which proves that $\mathrm{epi}(f)$ is convex.
Conversely, assume that $\mathrm{epi}(f)$ is convex. If $x \notin \mathrm{dom}\, f$ or $y \notin \mathrm{dom}\, f$, the inequality is trivial. So let us consider $(x,y) \in \mathrm{dom}(f)^2$. For (x,u) and (y,v) two points in $\mathrm{epi}(f)$, and $t \in [0,1]$, the point $t(x,u) + (1-t)(y,v)$ belongs to $\mathrm{epi}(f)$. So, $f(t(x + (1-t)y) \leq tu + (1-t)v$.

- If $f(x)$ and $f(y)$ are $> -\infty$, we can choose $u = f(x)$ and $v = f(y)$, which demonstrates the inequality.

- If $f(x) = -\infty$, we can choose u arbitrary close to $-\infty$.Letting u go to $-\infty$, we obtain $f(t(x + (1-t)y) = -\infty$, which demonstrates here again the inequality we wanted to prove. □

Exercise 2.1.2. Show that:

1. Affine functions are convex.
2. If f is convex, then $\mathrm{dom}(f)$ is convex.
3. If f_1, f_2 are convex and $a, b \in \mathbb{R}_+$, then $af_1 + bf_2$ is convex.
4. If f is convex and $x, y \in \mathrm{dom}\, f$, then for all $t \geq 1$, $z_t = x + t(y-x)$ satisfies the inequality $f(z_t) \geq f(x) + t(f(y) - f(x))$.
5. If f is convex, proper, with $\mathrm{dom}\, f = \mathcal{X}$, and if f is bounded, then f is constant.

In the following, the **upper hull** (or pointwise maximum) of a family $(f_i)_{i\in I}$ of convex functions will play a key role. By definition, the upper hull of the family is the function $x \mapsto \sup_i f_i(x)$.

Proposition 2.1.2. *Let $(f_i)_{i\in I}$ be a family of convex functions $\mathcal{X} \to [-\infty, +\infty]$, with I* any *set of indices. Then function $x \mapsto \sup_{i\in I} f_i(x)$ is convex.*

Proof. Let $f = \sup_{i\in I} f_i$ be the upper hull of the family.

(a) $\mathrm{epi}\, f = \bigcap_{i\in I} \mathrm{epi}\, f_i$. Indeed,

$$(x,t) \in \mathrm{epi}\, f \Leftrightarrow \forall i \in I, t \geq f_i(x) \Leftrightarrow \forall i \in I, (x,t) \in \mathrm{epi}\, f_i \Leftrightarrow (x,t) \in \cap_i \mathrm{epi}\, f_i.$$

(b) Any intersection of convex sets $K = \cap_{i\in I} K_i$ is convex (exercice 2.1.1)

(a) and (b) show that $\mathrm{epi}\, f$ is convex, *i.e.* that f is convex. □

Proposition* 2.1.3. *Let $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a* **jointly** *convex function. Then the function*

$$\begin{aligned} f : \mathcal{X} &\to \mathbb{R} \\ x &\mapsto \inf_{y\in\mathcal{Y}} F(x,y) \end{aligned}$$

is convex.

Proof. Let $u, v \in \mathcal{X}$ and $\alpha \in (0,1)$. We need to show that $f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha) f(v)$.

$$\begin{aligned} \alpha f(u) + (1-\alpha) f(v) &= \alpha \inf_{y_u \in \mathcal{Y}} F(u, y_u) + (1-\alpha) \inf_{y_v\in\mathcal{Y}} F(v, y_v) && \text{(definition of } f) \\ &= \inf_{y_u\in\mathcal{Y}, y_v\in\mathcal{Y}} \alpha F(u, y_u) + (1-\alpha) F(v, y_v) && \text{(separable problems)} \\ &\geq \inf_{y_u\in\mathcal{Y}, y_v\in\mathcal{Y}} F(\alpha u + (1-\alpha)v, \alpha y_u + (1-\alpha) y_v) && \text{(joint convexity of } F) \\ &= \inf_{y\in\mathcal{Y}} F(\alpha u + (1-\alpha)v, y) && \text{(change of variable)} \\ &= f(\alpha u + (1-\alpha)v) \end{aligned}$$

A valid change of variable is $(y, y') = (\alpha y_u + (1-\alpha) y_v, y_v)$. It is indeed invertible since we have $\alpha \in (0,1)$. □

2.1.2 Characterizations of convexity

We now provide other characterizations of convexity, which can be easier to verify for functions that are once or twice differentiable. Here, we assume that the domain of f is $\mathbb{R}^n$ (f is finite everywhere).

Proposition 2.1.4 (First-order characterization of convexity)**.** *Let f be a differentiable function on $\mathbb{R}^n$. Then f is convex if and only if*

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle. \tag{2.1.1}$$

Proof. Assume first that (2.1.1) holds. Let $x, y \in \mathbb{R}^n$ and $t \in (0,1)$. Define $z = tx + (1-t)y$. We then have

$$\begin{aligned} f(x) &\leq f(z) + \langle \nabla f(z), x - z \rangle, \\ f(y) &\leq f(z) + \langle \nabla f(z), y - z \rangle. \end{aligned}$$

Multiplying the first equation by t, the second one by $(1-t)$ and adding them gives

$$tf(x) + (1-t)f(y) \geq f(z).$$

This proves that f is convex, thanks to Proposition 2.1.1.
Let us now prove the other implication. Assume that f is convex and let $x, y \in \mathbb{R}^n$. By Proposition 2.1.1, we have for every $t \in (0,1)$ that

$$(1-t)f(x) + tf(y) \geq f(x + t(y-x)),$$

and by re-arranging we get

$$f(y) \geq f(x) + \frac{f(x + t(y-x)) - f(x)}{t}.$$

By definition of the gradient, the limit of the right-hand term as $t \to 0$ is $f(x) + \langle \nabla f(x), y - x \rangle$. This proves (2.1.1). □

For twice differentiable function, a simpler condition is available.

Proposition 2.1.5. *(Second-order characterization of convexity) Let f be a twice differentiable function on $\mathbb{R}^n$. Then f is convex if and only if its Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for every $x \in \mathbb{R}^n$.*

To prove this result, we first need a lemma related to integration.

Lemma 2.1.1. *Let f be a differentiable function on $\mathbb{R}^n$. Then for any $x, y \in \mathbb{R}^n$ we have*

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle = \int_0^1 \langle \nabla f(x + t(y-x)) - \nabla f(x), y - x \rangle \, dt,$$

and if f is twice differentiable, we also have

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle = \int_0^1 \int_0^t \big\langle \nabla^2 f\big(x + s(y-x)\big)(y-x), y - x \big\rangle \, ds \, dt. \tag{2.1.2}$$

Proof. Set $\varphi(t) = f(x + t(y - x))$ for all $t \in [0, 1]$. Note that $\varphi(t) = f(g(t))$ where $g(t) = x + t(y - x)$. By the chain rule,

$$\varphi'(t) = \left\langle \nabla f(g(t)), g'(t) \right\rangle = \langle \nabla f(x + t(y - x)), y - x \rangle \tag{2.1.3}$$

So $\varphi'(0) = \langle \nabla f(x), y - x\rangle$. Since $\varphi(0) = f(x)$ and $\varphi(1) = f(y)$, we have

$$f(y) - f(x) - \langle \nabla f(x), y - x\rangle = \varphi(1) - \varphi(0) - \varphi'(0)\,.$$

As φ is a primitive of φ', $\varphi(1) = \varphi(0) + \int_0^1 \varphi'(t)dt$. Hence, using (2.1.3)

$$\begin{aligned}\varphi(1) - \varphi(0) - \varphi'(0) &= \int_0^1 \varphi'(t)dt - \varphi'(0)\\ &= \int_0^1 \langle \nabla f(x + t(y - x)), y - x\rangle\, dt - \int_0^1 \langle \nabla f(x), y - x\rangle\, dt\\ &= \int_0^1 \langle \nabla f(x + t(y - x)) - \nabla f(x), y - x\rangle\, dt.\end{aligned}$$

This proves the first equality. Consider then the vector function $\Psi : s \mapsto \nabla f(x + s(y - x))$. If f is twice differentiable, then Ψ is differentiable and its Jacobian is given by $J_\Psi(s) = \nabla^2 f\left(x + s\left(y - x\right)\right)\left(y - x\right)$. By integration we get

$$\nabla f(x + t(y - x)) - \nabla f(x) = \int_0^t J_\Psi(s)ds = \int_0^t \nabla^2 f(x + s(y - x))(y - x)ds.$$

To conclude, we substitute this in the first formula and use linearity of integration. □

Proof of Proposition 2.1.5. If $\nabla^2 f$ is positive semidefinite everywhere (recall that this means that $\langle \nabla^2 f(x)u, u\rangle \geq 0$ for every $x, u \in \mathbb{R}^n$), then the right hand side of (2.1.2) is nonnegative, which proves that $f(y) - f(x) - \langle \nabla f(x), y - x\rangle \geq 0$ for every $x, y \in \mathbb{R}^n$. Therefore, f is convex by Proposition 2.1.4.
The reverse implication is left as an exercice. Hint: use the second-order Taylor expansion

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

and reason by contradiction: if $\nabla^2 f(x)$ is not positive semidefinite, then it has a negative eigenvalue. □

2.2 Strong convexity and Lipschitz gradient continuity

In optimization, we will often need to assume two key properties for our objective function: μ-strong convexity and L-Lipschitz gradient continuity. In this section, we assume that f is finite everywhere: $\operatorname{dom} f = \mathbb{R}^n$.

Definition 2.2.1 (Strong convexity)**.** For $\mu > 0$, we say that function $f : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex if the function $f - \frac{\mu}{2}\|\cdot\|^2$ is convex.

We give equivalent characterizations for strong convexity. Recall that for symmetric matrices $A, B \in \mathbb{R}^{n\times n}$, we write $A \succeq B$ iff $A - B$ is positive semidefinite.

Proposition 2.2.1 (Characterizations of strong convexity)**.** *The following properties are equivalent to strong convexity:*

(i) (if f is differentiable) $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \quad \forall x, y \in \mathbb{R}^n$,

(ii) (if f is twice differentiable) $\nabla^2 f(x) \succeq \mu I_n, \quad \forall x \in \mathbb{R}^n$.

Proof. This follows from the definition and Propositions 2.1.4 and 2.1.5, noting that the gradient of the function $x \to \frac{\mu}{2}\|x - y\|^2$ is $\mu(x - y)$ and its Hessian is μI_n. □

Definition 2.2.2 (Gradient Lipschitz continuity)**.** For $L \geq 0$, we say that a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ has L-Lipschitz gradients if $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for every $x, y \in \mathbb{R}^n$.

We recall that the spectral norm $\|A\|_2$ of a symmetric matrix A is its largest absolute eigenvalue.

Proposition 2.2.2. *The following propsitions are equivalent to L-Lipschitz gradient continuity:*

(i) $-\frac{L}{2}\|x - y\|^2 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle \leq \frac{L}{2}\|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n$,

and, if f is twice differentiable:

(ii) $-LI_n \preceq \nabla^2 f(x) \preceq LI_n, \quad \forall x \in \mathbb{R}^n$,

(iii) $\|\nabla^2 f(x)\|_2 \leq L, \quad \forall x \in \mathbb{R}^n$.

If additionnally f is convex, the left-hand side inequality of (i) and (ii) can be omitted.

Proof. If f has a Lipschitz gradient, then by Lemma 2.1.1 we have

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

This proves (i). The rest of the proof is left as an exercice. To prove (i) implies (ii), use second-order Taylor expansion. To prove (ii) implies (iii), use the fact that (ii) is equivalent to every eigenvalue of $\nabla^2 f(x)$ being in $[-L, L]$. To prove that (iii) implies gradient Lipschitz continuity, use the identity $\nabla f(y) - \nabla f(x) = \int_0^1 \nabla^2 f(x + t(y - x))(y - x) dt$. □

2.3 Fermat's rule, optimality conditions.

A point x is called a **minimizer** of f on $\mathbb{R}^n$ if $f(x) \leq f(y)$ for all $y \in \mathbb{R}^n$. The set of minimizers of f is denoted $\arg\min(f)$. A crucial property of convex functions is that the set of minimizers is exactly the set of points where the gradient is 0.

Theorem 2.3.1 (Fermat's rule)**.** *If f is convex and differentiable on $\mathbb{R}^n$, then*

$$x \in \arg\min f \; \Leftrightarrow \nabla f(x) = 0.$$

Recall that if f is a differentiable nonconvex function, a *necessary* condition (not a sufficient one) for $\bar{x}$ to be a minimizer of f, is that $\nabla f(\bar{x}) = 0$. If f is convex, then this is also sufficient. Besides, local minima for any function f are not necessarily global ones in general. In the convex case, everything works fine:

Proposition 2.3.1. *Let x be a local minimum of a convex function f. Then, x is a global minimizer.*

Proof. The local minimality assumption means that there exists an open ball $V \subset \mathbb{R}^n$, such that $x \in V$ and that, for all $u \in V$, $f(x) \le f(u)$.
Let $y \in \mathbb{R}^n$ and t such that $u = x + t(y - x) \in \mathbb{R}^n$. Then using the convexity of f, we get $f(u) \le tf(y) + (1-t)f(x)$. Re-organizing, we get

$$f(y) \ge t^{-1}\big(f(u) - (1-t)f(x)\big) \ge f(x). \qquad \square$$

2.4 Lower semi-continuity

In this course, we will consider functions with infinite values. Such function cannot be continuous. However, some kind of continuity would be very desirable. For infinite-valued convex function, lower semi-continuity is the good generalization of continuity.

Definition 2.4.1 (Reminder: $\liminf$: **limit inferior**).
The **limit inferior** of a sequence $(u_n)_{n\in\mathbb{N}}$, where $u_n \in [-\infty, \infty]$, is

$$\liminf(u_n) = \sup_{n\ge 0}\Big(\inf_{k\ge n} u_k\Big).$$

Since the sequence $V_n = \inf_{k\ge n} u_k$ is non decreasing, an equivalent definition is

$$\liminf(u_n) = \lim_{n\to\infty}\Big(\inf_{k\ge n} u_k\Big).$$

Definition 2.4.2 (Lower semicontinuous function). A function $f : \mathcal{X} \to [-\infty, \infty]$ is called **lower semicontinuous (l.s.c.) at** $x \in \mathcal{X}$ if for all sequence (x_n) which converges to x,

$$\liminf f(x_n) \ge f(x).$$

The function f is said to be **lower semicontinuous**, if it is l.s.c. at x, for all $x \in \mathcal{X}$.

The interest of l.s.c. functions becomes clear in the next result

Proposition 2.4.1 (epigraphical characterization). *Let $f : \mathcal{X} \to [-\infty, +\infty]$, any function f is l.s.c. if and only if its epigraph is closed.*

Proof. If f is l.s.c., and if $(x_n, t_n) \in \operatorname{epi} f \to (\bar{x}, \bar{t})$, then, $\forall n, t_n \ge f(x_n)$. Consequently,

$$\bar{t} = \liminf t_n \ge \liminf f(x_n) \ge f(\bar{x}).$$

Thus, $(\bar{x}, \bar{t}) \in \operatorname{epi} f$, and $\operatorname{epi} f$ is closed.
Conversely, if f is *not* l.s.c., there exists an $x \in \mathcal{X}$, and a sequence $(x_n) \to x$, such that $f(x) > \liminf f(x_n)$, *i.e.*, there is an $\epsilon > 0$ such that $\forall n \ge 0$, $\inf_{k\ge n} f(x_k) \le f(x) - \epsilon$. Thus, for all n, $\exists k_n \ge k_{n-1}$, $f(x_{k_n}) \le f(x) - \epsilon$. We have built a sequence $(w_n) = (x_{k_n}, f(x) - \epsilon)$, each term of which belongs to $\operatorname{epi} f$, and which converges to a limit $\bar{w} = (f(x) - \epsilon)$ which is outside the epigraph. Consequently, $\operatorname{epi} f$ is not closed. $\square$

Lower semi-continuity is a very desirable property for a function we want to optimize thanks to the following proposition.

Proposition 2.4.2. *Let f be a l.s.c function such that $\lim_{\|x\|\to+\infty} f(x) = +\infty$. Then there exists x^* such that $f(x^*) = \inf_{x\in\mathcal{X}} f(x)$.*

Proof. Let $(x_n)_{n\geq 0}$ be a minimizing sequence, that is a sequence of $\mathcal{X}$ such that we have $\lim_{n\to\infty} f(x_n) = \inf_{x\in\mathcal{X}} f(x)$.
Suppose that (x_n) were unbounded. Then there would exist a subsequence $(x_{\phi(n)})$ such that $\lim_{n\to\infty}\|x_{\phi(n)}\| \to +\infty$. By the assumptions on f, this implies that $\lim_{n\to\infty} f(x_n) = +\infty$ which contradicts the fact that $(x_n)_{n\geq 0}$ is a minimizing sequence.
Thus (x_n) is bounded and we can extract from it a subsequence $(x_{\phi(n)})$ converging to, say, x^*. As f is l.s.c., we get $\inf_{x\in\mathcal{X}} f(x) = \lim_{n\to\infty} f(x_{\phi(n)}) = \liminf f(x_{\phi(n)}) \geq f(x^*) \geq \inf_{x\in\mathcal{X}} f(x)$. □

A nice property of the family of l.s.c. functions is its stability with respect to point-wise suprema.

Proposition 2.4.3. *Let $(f_i)_{i\in I}$ a family of l.s.c. functions. Then, the upper hull $f = \sup_{i\in I} f_i$ is l.s.c.*

Proof. Let C_i denote the epigraph of f_i and $C = \operatorname{epi} f$. As already shown (proof of proposition 2.1.2), $C = \cap_{i\in I} C_i$. Each C_i is closed, and any intersection of closed sets is closed, so C is closed and f is l.s.c. □

2.5 Subdifferential

A classical property of convex function is that they are above their tangents. In a multi-dimensional setting, tangents become tangent hyperplanes.
When the function is not differentiable, we generalize the notion of gradient as follows.

Definition 2.5.1 (Subdifferential)**.** Let $f : \mathcal{X} \to [-\infty, +\infty]$ and $x \in \operatorname{dom}(f)$. A vector $\phi \in \mathcal{X}$ is called a **subgradient** of f at x if:

$$\forall y \in \mathcal{X}, \quad f(y) - f(x) \geq \langle \phi, y - x\rangle \ .$$

The **subdifferential** of f in x, denoted by $\partial f(x)$, is the whole set of the subgradients of f at x. By convention, $\partial f(x) = \emptyset$ if $x \notin \operatorname{dom}(f)$.

Interest: Gradient methods in optimization can still be used in the non-differentiable case, choosing a subgradient in the subdifferential.
For differentiable convex functions, the subdifferential reduces to the usual differential.

Proposition 2.5.1. *Let $f : \mathcal{X} \to (-\infty, \infty]$ be a convex function, differentiable in x. Then $\partial f(x) = \{\nabla f(x)\}$*

Proof. If f is differentiable at x, by Proposition 2.1.4 we have $\nabla f(x) \in \partial f(x)$. Let $\phi \in \partial f(x)$ and $t \neq 0$. Then for all $y \in \operatorname{dom}(f)$, $f(y) - f(x) \geq \langle \phi, y - x\rangle$. Applying this inequality to $y = x + t(\phi - \nabla f(x))$ leads to :

$$\frac{f\left(x + t(\phi - \nabla f(x))\right) - f(x)}{t} \geq \langle \phi, \phi - \nabla f(x)\rangle \ .$$

As $t \to 0$, the left term converges to $\langle \nabla f(x), \phi - \nabla f(x)\rangle$ by definition of the directional derivative. Finally,

$$\langle \nabla f(x) - \phi, \phi - \nabla f(x)\rangle \geq 0,$$

i.e. $\|\phi - \nabla f(x)\|^2 \leq 0$ and therefore $\phi = \nabla f(x)$. □

Fermat's rule extends nicely to subdifferentials.

Theorem 2.5.1 (Fermat's rule, nonsmooth case)**.** *If f is convex then*

$$x \in \arg\min f \;\Leftrightarrow\; 0 \in \partial f(x).$$

Proof. $x \in \arg\min f \;\Leftrightarrow\; \forall y, f(y) \geq f(x) + \langle 0, y - x\rangle \Leftrightarrow 0 \in \partial f(x)$. □

In order to clarify in what cases the subdifferential is non-empty, we need two more definitions:

Definition* 2.5.2. A set $A \subset \mathcal{X}$ is called an **affine space** if, for all $(x, y) \in A^2$ and for all $t \in \mathbb{R}$, $x + t(y - x) \in A$. The **affine hull** $\mathcal{A}(C)$ of a set $C \subset \mathcal{X}$ is **the smallest affine space** that contains C.

Definition* 2.5.3. Let $C \subset \mathbf{E}$. The **topology relative to** C is a topology on $\mathcal{A}(C)$. The open sets in this topology are the sets of the kind $\{V \cap \mathcal{A}(C)\}$, where V is open in $\mathbf{E}$.

Definition* 2.5.4. Let $C \subset \mathcal{X}$. The **relative interior** of C, denoted by $\text{relint}(C)$, is the interior of C for the topology relative to C. In other words, it consists of the points x that admit a neighborhood V, open in $\mathbf{E}$, such that $V \cap \mathcal{A}(C) \subset C$.

Clearly, $\text{int}(C) \subset \text{relint}(C)$. What's more, one can show that if C is convex, then $\text{relint}(C) \neq \emptyset$.

Proposition* 2.5.2. *Let $f : \mathcal{X} \to [-\infty, +\infty]$ be a convex function and $x \in \text{relint}(\text{dom}\, f)$. Then $\partial f(x)$ is non-empty.*

Proof. The proof is a bit technical and uses the concept of separating hyperplane Bauschke and Combettes (2011). □

Remark 2.5.1 (the question of $-\infty$ values)**.**
If $f : \mathcal{X} \to [-\infty, +\infty]$ is convex and if $\text{relint}\,\text{dom}\, f$ contains a point x such that $f(x) > -\infty$, then f *never* takes the value $-\infty$. So f is proper.

Exercise 2.5.1. * Show this point, using proposition 2.5.2.

Example 2.5.1. The absolute-value function $x \mapsto |x|$ defined on $\mathbb{R} \to \mathbb{R}$ admits as a subdifferential the sign application, defined by :

$$\text{sign}(x) = \begin{cases} \{1\} & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ \{-1\} & \text{ifi } x < 0\,. \end{cases}$$

Exercise 2.5.2. Determine the subdifferentials of the following functions, at the considered points :

1. In $\mathcal{X} = \mathbb{R}$, $f(x) = \iota_{[0,1]}$, at $x = 0, x = 1$ and $0 < x < 1$.
2. In $\mathcal{X} = \mathbb{R}^2$, $f(x) = \iota_{B(0,1)}$ (closed Euclidian ball), at x such that $\|x\| < 1$ and at x such that $\|x\| = 1$.
3. $\mathcal{X} = \mathbb{R}$,
$$f(x) = \begin{cases} +\infty & \text{if } x < 0 \\ -\sqrt{x} & \text{if } x \geq 0 \end{cases}$$
at $x = 0$, and $x > 0$.
4. $\mathcal{X} = \mathbb{R}^n$, $f(x) = \|x\|$, determine $\partial f(x)$, for any $x \in \mathbb{R}^n$.
5. $\mathcal{X} = \mathbb{R}$, $f(x) = x^3$. Show that $\partial f(x) = \emptyset$, $\forall x \in \mathbb{R}$. Explain this result.

Exercise 2.5.3. Let $f : \mathbb{R}^n \to \mathbb{R}$, differentiable. Show that: f is convex, if and only if

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \langle \nabla f(y) - \nabla f(x), y - x\rangle \geq 0.$$

2.6 Operations on subdifferentials

Until now, we have seen examples of subdifferential computations on basic functions, but we haven't mentioned how to derive the subdifferentials of more complex functions, such as sums or linear transforms of basic ones. A basic fact from differential calculus is that, when all the terms are differentiable, $\nabla(f+g) = \nabla f + \nabla g$. Also, if M is a linear operator, then we have the equality $\nabla(g \circ M)(x) = M^* \nabla g(Mx)$. Under qualification assumptions, these properties are still valid in the convex case, up to replacing the gradient by the subdifferential and point-wise operations by set operations. But first, we need to define operations on sets.

Definition 2.6.1 (addition and transformations of sets)**.** Let $A, B \subset \mathcal{X}$. The Minkowski sum and difference of A and B are the sets

$$A + B = \{x \in \mathcal{X} :\ \exists a \in A, \exists b \in B,\ x = a + b\}$$
$$A - B = \{x \in \mathcal{X} :\ \exists a \in A, \exists b \in B,\ x = a - b\}$$

Let $\mathcal{Y}$ another space and M any mapping from $\mathcal{X}$ to $\mathcal{Y}$. Then MA is the image of A by M,

$$MA = \{y \in \mathcal{Y} :\ \exists a \in A,\ y = Ma\}.$$

Proposition 2.6.1. *Let $f : \mathcal{X} \to (-\infty, +\infty]$ be a convex function, $g : \mathcal{Y} \to \mathbb{R}$ a convex differentiable function and $M : \mathcal{X} \to \mathcal{Y}$ a linear operator.*

$$\forall x \in \mathcal{X}, \partial(f + g \circ M)(x) = \partial f(x) + \{M^* \nabla g(Mx)\}$$

Proof. We first show that $\partial f(\,\cdot\,) + \{M^* \nabla g(Mx)\} \subseteq \partial(f + g \circ M)(\,\cdot\,)$. Let $x \in \mathcal{X}$ and let $\phi \in \partial f(x) + \{M^* \nabla g(Mx)\}$, which means that $\phi = u + M^* \nabla g(Mx)$ where $u \in \partial f(x)$. In particular, the latter subdifferential is not empty, which implies that $x \in \operatorname{dom} f$. By definition of u and convexity of g, for $y \in \mathcal{X}$,

$$\begin{cases} f(y) - f(x) \geq \langle u, y - x \rangle \\ g(My) - g(Mx) \geq \langle \nabla g(Mx), M(y - x) \rangle = \langle M^* \nabla g(Mx), y - x \rangle \,. \end{cases}$$

Adding the two inequalities,

$$(f + g \circ M)(y) - (f + g \circ M)(x) \geq \langle \phi, y - x \rangle \,.$$

Thus, $\phi \in \partial(f + g \circ M)(x)$ and $\partial f(x) + M^* \partial g(Mx) \subset \partial(f + g \circ M)(x)$.

For the converse inclusion, let $\phi \in \partial(f + g \circ M)(x)$. By definition of the subdifferential, for all $y \in \mathcal{X}$, $f(y) + g(My) \geq f(x) + g(Mx) + \langle \phi, y - x \rangle$. In particular, for all $h \in [0, 1]$,

$$f(x + h(y - x)) + g(M(x + h(y - x))) \geq f(x) + g(Mx) + h\langle \phi, y - x \rangle.$$

As f is convex, $f(x + h(y - x)) \leq hf(y) + (1 - h)f(x)$ and so, dividing by h,

$$f(y) \geq f(x) - \frac{1}{h}\big(g(M(x + h(y - x))) - g(x)\big) + \langle \phi, y - x \rangle.$$

We let h tend to 0 and we obtain $f(y) \geq f(x) + \langle -\nabla(g \circ M)(x) + \phi, y - x \rangle$. Said otherwise, $\phi - \nabla(g \circ M)(x) \in \partial f(x)$. We conclude using the chain rule. $\square$

When both functions are not differentiable, the previous theorem requires and additional assumption to apply.

Proposition* 2.6.2. *Let $f : \mathcal{X} \to (-\infty, +\infty]$, $g : \mathcal{Y} \to (-\infty, \infty]$ two convex functions and let $M : \mathcal{X} \to \mathcal{Y}$ a linear operator.*

$$\forall x \in \mathcal{X}, \partial f(x) + M^* \partial g(Mx) \subseteq \partial(f + g \circ M)(x)$$

Moreover, if $0 \in \operatorname{relint}(\operatorname{dom} g - M \operatorname{dom} f)$, then

$$\forall x \in \mathcal{X}, \partial(f + g \circ M)(x) = \partial f(x) + M^* \partial g(Mx)$$

Proof. The general proof is omitted. It makes use of the Fenchel-Young inequality and of the strong duality theorem that will be given in Chapters 6 and 7.
Remark that when g is differentiable, $\operatorname{dom} g = \mathcal{Y}$ and thus the condition is trivially satisfied. □

Chapter 3

Algorithms for smooth unconstrained minimization

In this chapter, we study unconstrained problems of the form

$$\min_{x \in \mathbb{R}^n} f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be **differentiable**.

3.1 How to compute gradients?

Optimization algorithms heavily rely on gradients. Hence, given a function $f : \mathbb{R}^n \to \mathbb{R}$, we would like to compute its gradient. By definition, $\nabla f(x)$ is the unique vector of $\mathbb{R}^n$ such that

$$f(x+h) = f(x) + \langle \nabla f(x), h \rangle + o(\|h\|),$$

where the notation $o(\|h\|)$ denotes any function of the form $\|h\|\epsilon(h)$ with $\lim_{h \to 0} \epsilon(h) = 0$.
Let us recall three methods to calculate this vector.

3.1.1 Using the definition

We compute $f(x+h)$ and write it as a sum of $f(x)$, a term linear in h and a term that is negligible with respect to $\|h\|$. By the unicity of the differential, this allows to identify the gradient.
Example. We consider $f(x) = \|Ax - b\|^2$.

$$\begin{aligned} f(x+h) = \|A(x+h) - b\|^2 &= \|Ax - b\|^2 + 2\langle Ax - b, Ah \rangle + \|Ah\|^2 \\ &= f(x) + 2\langle A^\top (Ax - b), h \rangle + o(\|h\|). \end{aligned}$$

Indeed, we have $\|Ah\| \leq \|A\|_F \|h\|^2 = o(\|h\|)$. Thus, $\nabla f(x) = 2A^\top (Ax - b)$.

3.1.2 Using partial derivatives

We know that the gradient is the vector of all the partial derivatives. Hence, we can compute $\frac{\partial f}{\partial x_i}(x)$ for all i and reconstruct the vector.
Example. Let us consider the function $f(x) = \|Ax - b\|^2$ where $A \in \mathbb{R}^{m \times n}$. We can write

$$f(x) = \sum_{j=1}^{m} \Big(\sum_{i=1}^{n} A_{j,i} x_i - b_j \Big)^2$$

and so

$$\frac{\partial f}{\partial x_k}(x) = 2\sum_{j=1}^{m} A_{j,k}\Big(\sum_{i=1}^{n} A_{j,i}x_i - b_j\Big) \ .$$

We recognise the components of the vector

$$\nabla f(x) = 2A^\top(Ax - b) \ .$$

3.1.3 Using the chain rule

Let $g : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^p$. The chain rule states that the Jacobian matrix of the function $f \circ g$ at x is given by

$$J_{f\circ g}(x) = J_f(g(x)) \times J_g(x) \ .$$

We recall that

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix}$$

is the unique linear map such that

$$g(x+h) = g(x) + J_g(x)h + o(\|h\|) \ .$$

The chain rule allows us to combine simple functions in order to obtain complex functions. In is at the basis of automatic differentiation and the resolution of neural network models.
When $f : \mathbb{R}^m \to \mathbb{R}$ and $g(x) = Ax$ where A is a $m \times n$ matrix, the formula simplifies as

$$\nabla(f \circ A)(x) = A^\top \nabla f(Ax) \ .$$

Example. We consider $f(x) = \|Ax - b\|^2$.
Let us remark that $f(x) = h(Ax)$ where $h(y) = \|y - b\|^2$.
Since $h(y+h) = \|y+h-b\|^2 = \|y-b\|^2 + 2\langle y-b, h\rangle + \|h\|^2$, we know that $\nabla h(y) = 2(y-b)$.
Using the chain rule, we get $\nabla f(x) = \nabla(h \circ A)(x) = A^\top \nabla h(Ax) = 2A^\top(Ax - b)$.

Remark 3.1.1 (Computing Hessians.)**.** For computing the Hessian $\nabla^2 f$ of a twice differentiable function, similar strategies can be used, once we know the gradient.

1. Using the definition: for $x \in \mathbb{R}^n$, the Hessian $\nabla^2 f(x)$ is the unique matrix satisfying for every h

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

2. Differential of gradient: the Hessian is the Jacobian of the gradient, i.e. $\nabla^2 f(x) = J_{\nabla f}(x)$. We can then use the definition of the Jacobian, as well as the chain rule when useful.

3. Partial derivatives: the Hessian has coordinates for $i, j \in \{1 \dots n\}$

$$\left[\nabla^2 f(x)\right]_{ij} = \frac{\partial f}{\partial x_i \partial x_j}.$$

3.2 Gradient method

The gradient method is the most basic optimization method for a differentiable function f. It consists in a sequence $(x_k)_{k\in\mathbb{N}}$ of points in $\mathbb{R}^n$ defined by induction from $x_0 \in \mathbb{R}^n$ by

$$\boxed{x_{k+1} = x_k - \gamma_k \nabla f(x_k)}$$

where for all k, γ_k is a positive coefficient called the step size. The choice of the step size is key for performance: if it is too small, the method is slow, and too big, the iterates diverge.

Constant step sizes

First, we prove a result for gradient descent with constant step size. In this case, we need to assume that f has a L-Lipschitz continuous gradient, meaning that

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

If f is C^2, this is equivalent to assuming that the Hessian is bounded in spectral norm, i.e. $\|\nabla^2 f(x)\|_2 \leq L$ for every $x \in \mathbb{R}^n$ (see Section 2.2).

The next theorem states that if the step size is set to $1/L$ and f is **convex**, then gradient descent converges at a rate proportional to $1/k$. It moreover f is μ-**strongly convex**, we get a faster rate that we call “linear”.

Theorem 3.2.1 (Gradient method, smooth convex case)**.** *Let f be a convex differentiable function that has a minimizer x^* and whose gradient is L-Lipschitz continuous. The gradient method with constant step size $\gamma_k = \frac{1}{L}$ satisfies for $k \geq 1$*

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

If moreover f is μ-strongly convex, then

$$f(x_k) - f(x^*) \leq L\left(1 - \frac{\mu}{L}\right)^k \|x_0 - x^*\|^2$$
$$\|x_k - x^*\|^2 \leq 2\left(1 - \frac{\mu}{L}\right)^k \|x_0 - x^*\|^2$$

Proof. The convergence rate of gradient method follows from the more general result of Theorem 4.2.1 for the proximal gradient method (in the special case where $g \equiv 0$). We will prove it in Chapter 4.

In the strongly convex case, we can also use the Picard fixed point theorem. Indeed, the map $x \mapsto x - \frac{1}{L}\nabla f(x)$ is a contraction (see Exercice 3 from the first exercice sheet for a proof). □

Remark 3.2.1. Note that if f has a Lipschitz continuous gradient with constant L, then it is also L'-Lipschitz continuous for any $L' \geq L$. Therefore any fixed step size γ satisfying $\gamma \in (0, \frac{1}{L}]$ also works. The convergence rate will then be $\frac{\|x_0 - x^*\|^2}{2\gamma k}$ for the convex case.

Corollary 3.2.1. *Let $\epsilon > 0$. If we run the gradient method for $K = \lceil \frac{L\|x_0 - x^*\|^2}{2\epsilon} \rceil$ iterations, then we can guarantee that $f(x_K) - f(x^*) \leq \epsilon$. We say that we have found an ϵ-solution.*

Proof. $f(x_K) - f(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2K} \leq \epsilon$. □

Line search

Considering constant step sizes makes the proof easier but has drawbacks in practice:

- One needs to compute the Lipschitz constant of the gradient of f L, which may be a non-negligible amount of work.
- Some functions, like $(x \mapsto x^4)$ simply do not have a Lipschitz gradient globally. However, the gradient can be locally Lipschitz (this is always the case for C^2 functions).
- Even if the function has a Lipschitz gradient; the theoretical estimation of the Lipschitz constant might be too conservative. It could take into account regions where the curvature is large but that are never visited by the algorithm.

A solution to these three issues is a line search procedure. The idea of line search is to choose γ_k adaptively using local information.

Exact line search. We take

$$\gamma_k = \arg\min_{\gamma \in \mathbb{R}_+} f(x_k - \gamma \nabla f(x_k)).$$

This method is most efficient when we have a closed formula for the 1-dimensional optimization problem.

Taylor-based line search. In the proof of convergence of Theorem 4.2.1, we only need the Lipschitz continuity of ∇f in order to ensure that

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

The Taylor-based line search chooses a step size γ_k such that for the tentative update defined by $x^+(\gamma_k) = x_k - \gamma_k \nabla f(x_k)$, we have

$$f(x^+(\gamma_k)) \leq f(x_k) + \langle \nabla f(x_k), x^+(\gamma_k) - x_k \rangle + \frac{1}{2\gamma_k}\|x_k - x^+(\gamma_k)\|^2$$

using the following algorithm.
We set $b > 0, a \in (0,1)$ and we find the first nonnegative integer l such that

$$f(x^+(ba^l)) \leq f(x_k) + \langle \nabla f(x_k), x^+(ba^l) - x_k \rangle + \frac{1}{2ba^l}\|x_k - x^+(ba^l)\|^2 \tag{3.2.1}$$

Then we set $\gamma_k = ba^l$. Clearly, if such an l exists, the desired inequality will hold.

Proposition 3.2.1. *If f has a L-Lipschitz gradient, then the Taylor-based line search will terminate with $l < \frac{log(ba^{-1}L)}{\log(a^{-1})}$ and $ba^l < aL$*

Proof. If ∇f is L-Lipschitz, then it is also L'-Lipschitz for all $L' \geq L$. Hence, as soon as $1/(ba^l) \geq L$ (which will eventually happen since $1/a > 1$), (3.2.1) will hold and the line search will terminate. Just before, we had $1/(ba^{l-1}) < L$, so, $1/(ba^l) < a^{-1}L$. We get the bound on l by passing to the log. □

Note that we do not need to know this Lipschitz constant in order to run the line search.
Classical choices for the parameters are $a = 0.5$ and $b = 2\gamma_{k-1}$.

Armijo's line search. This line search is the most famous one. Given $a \in (0,1)$, $b > 0$ and $\beta \in (0,1)$, determine the first integer l such that

$$f(x^+(ba^l)) \leq f(x_k) + \beta\langle \nabla f(x_k), x^+(ba^l) - x_k\rangle$$

In the case of gradient descent, $\langle \nabla f(x_k), x^+(\gamma) - x_k\rangle = -\gamma\|\nabla f(x_k)\|^2$ so we can see that the Taylor-based line search is equivalent to an Armijo's line search with $\beta = 1/2$.

3.3 Accelerated gradient method

The standard gradient method is simple but can suffer from excessively slow convergence. A variant known as *Nesterov's accelerated gradient*, adds a *momentum* term to the gradient update to speed up the method.

The method starts by initializing $x_0 \in \mathbb{R}^n$ as well as an auxiliary variable $y_0 = x_0$ and then performs the update

$$\boxed{\begin{aligned} y_{k+1} &= x_k - \gamma_k \nabla f(x_k) \\ x_{k+1} &= y_{k+1} + \frac{k}{k+3}(y_{k+1} - y_k) \end{aligned}}$$

The intuitive idea is that adding the inertia term proportional to the previous displacement $y_{k+1} - y_k$ allows to accelerate convergence by "eliminating oscillations". Again, the step size γ_k can be either set constant to $1/L$ or chosen through line search.

The next theorem states that this method enjoys a faster convergence rate than standard gradient descent for **convex** functions with Lipschitz continuous gradient.

Theorem 3.3.1 (Accelerated gradient method, smooth convex case)**.** *Let f be a convex differentiable function that has a minimizer x^* and whose gradient is L-Lipschitz continuous. The accelerated gradient method with constant step size $\gamma_k = \frac{1}{L}$ satisfies for $k \geq 1$*

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

The proof is more involved and left out of the scope of this course. The accelerated gradient method enjoys **optimal oracle complexity**. This means that no other algorithm that uses only the first-order oracle $x \mapsto (f(x), \nabla f(x))$ has a better worst-case complexity for this class of problems.

Note that this is a theoretical argument based on a worst-case scenario. On a specific function, some other method might perform better. However, it is usually observed that adding a momentum term to gradient descent allows a significant speedup in practice.

Finally, we mention that there is also a variant of the accelerated method adapted to μ-strongly convex functions: it enjoys a convergence rate proportional to $\left(1 - \sqrt{\frac{\mu}{L}}\right)^k$ instead of $\left(1 - \frac{\mu}{L}\right)^k$ for standard gradient descent, which can be a significant improvement when $\mu \ll L$.

3.4 Second-order methods

Looking at the convergence rate of the gradient method for strongly convex functions, we see that it depends on the **condition number** $\kappa = L/\mu$. For *ill-conditioned problems* for which $\kappa \gg 1$, the convergence is slow. One remedy is to rescale the gradient step with a suitable matrix. This is the spirit of the Newton and Quasi-Newton methods.

3.4.1 Newton's method

Newton's method uses the Hessian matrix in order to ensure convergence in a smaller number of iterations. As shown in 0EL01 – Optimisation et analyse numérique – the method enjoys quadratic convergence locally.

Theorem 3.4.1. *If f is three times continuously differentiable and if x_0 is chosen close enough to a local minimum x^* where the Hessian matrix of f is positive definite, then the sequence x_k generated by Newton's method*

$$\boxed{x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)}$$

converges to x^* and there exists $M > 0$ such that

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

Out of the region of quadratic convergence, Newton's method may diverge. Hence, the method should be combined with a line search procedure similar to (4.2.2) in order to ensure convergence even if x_0 is not close to x^*.
Newton's method with line search is given by

$$\begin{aligned} & x^+(\gamma) = x_k - \gamma(\nabla^2 f(x_k))^{-1} \nabla f(x_k) \\ & x_{k+1} = x^+(\gamma_k) \end{aligned}$$

where $\gamma_k \in]0, 1]$ is such that

$$f(x^+(\gamma_k)) \leq f(x_k) + \langle \nabla f(x_k), x^+(\gamma_k) - x_k \rangle + \frac{1}{2\gamma_k}(x^+(\gamma_k) - x_k)^\top \nabla^2 f(x_k)(x^+(\gamma_k) - x_k) \ .$$

We should also make sure that the line search selects $\gamma_k = 1$ when possible in order to keep the quadratic convergence.

3.4.2 Variable metric methods

Newton's method requires very few iterations to converge, but each iteration can be very costly. Computing the Newton step $x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$ requires solving a linear system of size $n \times n$, which usually needs $\mathcal{O}(n^3)$ operations. This complexity is prohibitive when n is larger than a few thousands, which is often the case in machine learning.
Variable metric methods seek to preserve the benefit of using second-order information but at at lesser computational cost. They replace the Hessian $\nabla^2 f(x_k)$ by a simpler positive definite matrix H_k, and perform the update

$$\boxed{x_{k+1} = x_k - H_k^{-1} \nabla f(x_k)}$$

There are several options for choosing H_k.

- **Constant:** if $H_k = H$ for every k, then H is called a *preconditioner*. It is usually chosen empirically such that H is as close as possible to $\nabla^2 f(x)$ for x in a certain domain.

- **Diagonal:** if $H_k = \mathrm{Diag}(h_1^{(k)} \dots h_n^{(k)})$, it is very easy to invert. Some methods compute the coefficients $h_i^{(k)}$ adaptively as a function of the past gradients. This is the case for the `AdaGrad` and `Adam` methods which are popular for neural network training (more details in Chapter 5 on stochastic methods).

- **Quasi-Newton:** these methods mimic the behavior of the Newton method by estimating the inverse Hessian in a smart way. They compute a matrix H_k^{-1} which behaves like $\nabla^2 f(x_k)^{-1}$ on the previously seen directions, using only gradient information. Among them, the `BFGS` algorithm is particularly popular: it is known as one of the best state-of-the art methods for unconstrained smooth problems.

Chapter 4

Algorithms for nonsmooth problems

In this chapter, we turn to problems which are not differentiable. Depending on the structure, we can use two types of algorithms: subgradient methods (slow but general) or proximal gradient methods (efficient but requires a special structure).

4.1 Subgradient method

We study here the problem

$$\min_{x \in \mathbb{R}^n} F(x)$$

where $F : \mathbb{R}^n \to \mathbb{R}$ is convex but not differentiable. Here, we cannot use the gradient $\nabla F(x)$ as it is not defined everywhere. We can however use a generalization: the subdifferential $\partial F(x)$ (see Section 2.5). In return, we shall set smaller, diminishing step-sizes to ensure that the algorithm converges. The method writes

Algorithme 1 : Subgradient method

```
select $g_k \in \partial F(x_k)$
$x_{k+1} = x_k - \gamma_k g_k$
```

where for all k, γ_k is a positive coefficient. The next proposition shows that the method works if the step size satisfies a decrease condition.

Proposition 4.1.1. *Let $F : \mathbb{R}^n \to \mathbb{R}$ be a convex function that has a minimizer x^* and γ_k be a sequence such that $\frac{\sum_{l=0}^k \gamma_l^2}{\sum_{l=0}^k \gamma_l} \to 0$ when $k \to +\infty$. Suppose that there exists $C \geq 0$ such that any subgradient g of f satisfies $\|g\| \leq C$. Then the subgradient method satisfies*

$$F(\bar{x}_k^\gamma) - F(x^*) \to 0$$

where $\bar{x}_k^\gamma = \dfrac{\sum_{l=0}^k \gamma_l x_l}{\sum_{j=0}^k \gamma_j}$ is a convex combination of all previous iterates.

Proof. This is a special case of Theorem 5.2.1, which does apply even if there is nothing random in the function. □

The subgradient method is general but is known to be quite slow.

4.2 Proximal gradient method

When the problem has more favorable structure, we can turn to the proximal gradient method, which is more efficient. The proximal gradient method is a method designed to solve composite problems of the type

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + g(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and has Lipschitz gradients, and $g : \mathbb{R}^n \to [-\infty, +\infty]$ is a possibly nonsmooth convex function which is *simple enough*, in a sense that we will explain now.
Note that g may take infinite values to encode constraints (see Chapter 2).

Definition 4.2.1. The proximal operator of a convex lower-semicontinuous function $g : \mathbb{R}^n \to [-\infty, +\infty]$ is defined as

$$\operatorname{prox}_g(x) = \arg\min_{y\in\mathbb{R}^n} g(y) + \frac{1}{2}\|x-y\|^2.$$

Because of the strong convexity of $(y \mapsto g(y) + \frac{1}{2}\|x-y\|^2)$, there is a unique minimizer in the problem defining the proximal operator.
We assume g is simple enough so that the proximal operator of g can be computed easily, ideally in closed form. Two examples can be found in the exercise sheet: the proximal operator of the 1-norm $g(x) = \|x\|_1$ is the element-wise soft-thresholding operator and the proximal operator of a convex indicator function $g(x) = \iota_C(x)$ is the projection on C.
The proximal gradient method initializes $x_0 \in \mathbb{R}^n$ and performs the update

$$\boxed{x_{k+1} = \operatorname{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))}$$

for given step sizes $\gamma_k > 0$.
The advantage of this method is to deal with simple non-differentiable functions in a way that preserves the good convergence properties of gradient descent. Indeed, for the problems where the proximal operator of g is easy to compute, the proximal gradient method will be much faster than the subgradient method. In particular, we do not need to set a diminishing step size.

Constant step size. As for standard gradient descent, we can choose the step size to be $1/L$, where L is the Lipschitz constant of the smooth component f.

Theorem 4.2.1. *Let f be a convex differentiable function whose gradient is L-Lipschitz continuous, g be a convex lower semi-continuous function and x^* a minimizer of $F = f + g$. The proximal gradient method with constant step size $\gamma_k = \frac{1}{L}$ satisfies*

$$F(x_k) - F(x^*) \le \frac{L\|x_0 - x^*\|^2}{2k}$$

If moreover f is μ-strongly convex, then denoting $\Delta_0 = F(x_0) - F(x^*) + \frac{L}{2}\|x_0 - x^*\|^2$,

$$F(x_k) - F(x^*) \le \left(1 - \frac{\mu}{L}\right)^k \Delta_0$$
$$\|x_k - x^*\|^2 \le \frac{2}{L}\left(1 - \frac{\mu}{L}\right)^k \Delta_0$$

To prove this theorem, we will need a lemma on the proximal operator.

Lemma 4.2.1. *Let $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function and $\gamma > 0$. Define the proximal operator of γg as $\mathrm{prox}_{\gamma g}(x) = \arg\min_{y \in \mathbb{R}^n} \gamma g(y) + \frac{1}{2}\|y - x\|^2$. Denoting $p = \mathrm{prox}_{\gamma g}(x)$, we have for all z,*

$$g(p) + \frac{1}{2\gamma}\|x - p\|^2 \leq g(z) + \frac{1}{2\gamma}\|x - z\|^2 - \frac{1}{2\gamma}\|p - z\|^2. \tag{4.2.1}$$

Proof. By Fermat's rule applied to the convex function $y \to \gamma g + \frac{1}{2}\|y - x\|^2$ and the definition of p, we have $0 \in \partial g(p) + \frac{1}{\gamma}\{p - x\}$.
Define the function

$$h : y \mapsto g(y) + \frac{1}{2\gamma}\|x - y\|^2 - \frac{1}{2\gamma}\|p - y\|^2 = g(y) + \frac{1}{2\gamma}\|x\|^2 - \frac{1}{2\gamma}\|p\|^2 + \frac{1}{\gamma}\langle y, p - x\rangle.$$

h is convex and $0 \in \partial h(p)$, therefore p minimizes h (Fermat's rule) and $h(p) \leq h(z)\ \forall z$. □

Proof of the theorem. We proceed with the proof of the theorem. Using the gradient Lipschitz property, we have by Proposition 2.2.2(i) that

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

We then use Lemma 4.2.1 with $x = x_k - \frac{1}{L}\nabla f(x_k)$, and $\mathrm{prox}_{\frac{1}{L}g}(x_k) = x_{k+1}$ by definition of the algorithm. We have then for every $z \in \mathbb{R}^n$

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

Adding the two inequalities and rearranging yields

$$\begin{aligned} f(x_{k+1}) + g(x_{k+1}) \leq{}& f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &+ g(z) + \frac{L}{2}\|x_k - \frac{1}{L}\nabla f(x_k) - z\|^2 - \frac{L}{2}\|x_{k+1} - z\|^2 - \frac{L}{2}\|x_{k+1} - x_k + \frac{1}{L}\nabla f(x_k)\|^2 \end{aligned}$$

We develop the squared norms involving a $\frac{1}{L}\nabla f(x_k)$ term and we get:

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

By setting $z = x_k$, we obtain monotonicity: $f(x_{k+1}) + g(x_{k+1}) \leq f(x_k) + g(x_k)$.
By setting $z = x^*$, we obtain

$$\begin{aligned} f(x_{k+1}) + g(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x^* - x_k\rangle + g(x^*) + \frac{L}{2}\|x_k - x^*\|^2 - \frac{L}{2}\|x_{k+1} - x^*\|^2 \\ &\leq f(x^*) + g(x^*) + \frac{L}{2}\|x_k - x^*\|^2 - \frac{L}{2}\|x_{k+1} - x^*\|^2 . \end{aligned}$$

We sum for k between 0 and $K - 1$.

$$\sum_{k=0}^{K-1}(f(x_{k+1}) + g(x_{k+1})) \leq K(f(x^*) + g(x^*)) + \frac{L}{2}\|x_0 - x^*\|^2 - \frac{L}{2}\|x_K - x^*\|^2$$

By monotonicity, we also have

$$K(f(x_K) + g(x_K)) \leq \sum_{k=0}^{K-1}(f(x_{k+1}) + g(x_{k+1}))$$

Combining the two previous inequalities and rearranging yields the result for the convex case. We leave the strongly convex case as an exercise. □

Interpretation of the proximal gradient method. Let us provide an intuitive explanation for the method. Applying the Taylor-Lagrange inequality to f for $x = x_k$ and adding $g(y)$ yields

$$f(y) + g(y) \leq f(x_k) + \langle f(x_k), y - x_k \rangle + \frac{L}{2}\|y - x_k\|^2 + g(y), \quad \forall y \in \mathbb{R}^n.$$

This gives a global upper majorant of $F = f + g$ which is tight for $y = x_k$. Then the proximal gradient method computes x_{k+1} as the minimizer of this majorant. Indeed, one can show (and you can verify this as an exercice) that

$$\operatorname{prox}_{\frac{1}{L}g}\left(x_k - \frac{1}{L}\nabla f(x_k)\right) = \arg\min_{y \in \mathbb{R}^n} \left\{ f(x_k) + \langle f(x_k), y - x_k \rangle + \frac{L}{2}\|y - x_k\|^2 + g(y) \right\}$$

Special case: projected gradient method. Let C be a closed convex subset of $\mathbb{R}^n$, and assume we want to solve the constrained problem

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

Introducing the indicator function ι_C, this rewrites as the composite problem

$$\min_{x \in \mathbb{R}^n} f(x) + \iota_C(x).$$

Applying the proximal gradient method with $g = \iota_C$ gives the **projected gradient method**:

$$\boxed{x_{k+1} = \operatorname{Proj}_C\left(x_k - \gamma_k \nabla f(x_k)\right)}$$

Here, $\operatorname{Proj}_C(y) = \arg\min_{x \in C} \|x - y\|^2$ denotes the Euclidean projection on C.

Adaptive step sizes. As for the standard gradient method, we can set the step size γ_k adaptively using the Taylor-based line search. We need to choose γ_k such that for $x^+(\gamma_k) = \operatorname{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))$, we have

$$f(x^+(\gamma_k)) \leq f(x_k) + \langle \nabla f(x_k), x^+(\gamma_k) - x_k \rangle + \frac{1}{2\gamma_k}\|x_k - x^+(\gamma_k)\|^2. \tag{4.2.2}$$

Armijo's line search can be generalized in the same way.

Remark 4.2.1. A slightly better rate can be obtained by taking $\gamma = \frac{2}{L+\mu}$, provided μ is known Nesterov (2004).

Chapter 5

Stochastic gradient descent

5.1 Algorithm

Let us consider a mesurable function

$$\begin{aligned} f : \mathcal{X} \times \Xi &\to \mathbb{R} \\ (x,t) &\mapsto f(x,t) \end{aligned}$$

and a random variable ξ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in Ξ. We suppose that $\mathbb{E}(|f(x,\xi)|) < +\infty$.
The stochastic gradient method has been designed to solve the optimization problem

$$\min_{x \in \mathcal{X}} \mathbb{E}(f(x,\xi)) \tag{5.1.1}$$

The challenge is that the law of ξ is not supposed to be known and we cannot compute $\mathbb{E}$. Instead, it is revealed through (ξ_k), a sequence of i.i.d. samples of ξ. Given a sequence of step sizes γ_k, the algorithm reads

$$\boxed{x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})}$$

where $\nabla f(x_k, \xi_{k+1})$ is the gradient of $(x \mapsto f(x, \xi_{k+1}))$ at x_k.

Remark 5.1.1. If $(x \mapsto f(x, \xi_{k+1}))$ is not differentiable, one can use a subgradient of the function instead of its gradient.

Example 5.1.1 (Empirical Risk Minimization). In this context, we are given N data points, each of which is associated with a loss function f_i, $1 \leq i \leq N$. A typical model in machine learning consists in minimizing the empirical risk given by

$$\min_x \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

This corresponds to Problem (5.1.1) with $\xi = I \sim U(\{1, \dots, N\})$. The expectation is computable but N may be so large that this takes a long time. Indeed, denoting $C_{\nabla f}$ the cost of computing $\nabla f_i(x)$, each iteration of gradient descent costs $NC_{\nabla f}$. Running stochastic gradient on this problem leads to an algorithm with very low complexity per iteration, namely $C_{\nabla f}$. This algorithm is thus often used in practice.

$$\begin{cases} \text{Generate } I_{k+1} \sim U(\{1, \dots, N\}) \\ x_{k+1} = x_k - \gamma_k \nabla f_{I_{k+1}}(x_k) \end{cases}$$

Example 5.1.2 (Least Mean Squares). We are given a random variable $\xi = (X, Y)$ where $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}$. Least Mean Squares (LMS) is a regression problem in expectation

$$\min_{w \in \mathbb{R}^n} \frac{1}{2}\mathbb{E}[(Y - X^\top w)^2]$$

Stochastic gradient on this problems writes

$$w_{k+1} = w_k - \gamma_k (X_{k+1}^\top w_k - Y_{k+1}) X_{k+1}$$

This algorithm is also used and analysed when (ξ_k) is not i.i.d. Macchi and Eweda (1983).

5.2 Convergence

We denote $F(x) = \mathbb{E}(f(x, \xi))$.

Theorem 5.2.1. *Suppose that:*

- $(x \mapsto f(x,\xi))$ *is convex and differentiable for all* ξ*,*
- *there exists* $C > 0$ *such that* $\mathbb{E}(\|\nabla f(x,\xi)\|^2) \le C$ *for all* x
- *there exists* $x^* \in \arg\min F$*,*
- *the sequence* γ_k *is deterministic.*

The iterates of the stochastic gradient algorithm $x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$ *satisfy the convergence guarantee*

$$\mathbb{E}\Big[F(\bar{x}_k^\gamma) - F(x^*)\Big] \le \frac{\mathbb{E}[\|x_0 - x^*\|^2] + C\sum_{l=0}^k \gamma_l^2}{2\sum_{l=0}^k \gamma_l}$$

where $\bar{x}_k^\gamma = \dfrac{\sum_{l=0}^k \gamma_l x_l}{\sum_{j=0}^k \gamma_j}$ *is a convex combination of all previous iterates.*

Proof. Let us first remark that $\mathbb{E}[\nabla f(x,\xi)] \in \partial F(x)$ for all x. Indeed,

$$\begin{aligned}
&f(y,\xi) \ge f(x,\xi) + \langle \nabla f(x,\xi), y - x\rangle \\
&F(y) = \mathbb{E}[f(y,\xi)] \ge \mathbb{E}[f(x,\xi)] + \mathbb{E}[\langle \nabla f(x,\xi), y - x\rangle] = F(x) + \langle \mathbb{E}[\nabla f(x,\xi)], y - x\rangle\ .
\end{aligned}$$

Now, we denote by $\mathbb{E}_k$ the expectation knowing $(\xi_1, \dots, \xi_k)$. Note that x_k is measurable with respect to $(\xi_1, \dots, \xi_k)$, so that

$$\begin{aligned}
\mathbb{E}_k[\|x_{k+1} - x^*\|] &= \mathbb{E}_k[\|x_k - x^*\| + 2\langle x_{k+1} - x_k, x_k - x^*\rangle + \|x_{k+1} - x_k\|^2] \\
&= \|x_k - x^*\| - 2\gamma_k\langle \mathbb{E}_k[\nabla f(x_k, \xi_{k+1})], x_k - x^*\rangle + \gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2] \\
&\le \|x_k - x^*\| + 2\gamma_k\langle \mathbb{E}_k[\nabla f(x_k, \xi_{k+1})], x^* - x_k\rangle + \gamma_k^2 C \\
&\le \|x_k - x^*\| + 2\gamma_k(F(x^*) - F(x_k)) + \gamma_k^2 C\ .
\end{aligned}$$

We reorganise and apply total expectation:

$$\mathbb{E}[\gamma_k(F(x_k) - F(x^*))] \le -\frac{1}{2}\mathbb{E}[\|x_{k+1} - x^*\|^2] + \frac{1}{2}\mathbb{E}[\|x_k - x^*\|^2] + \frac{\gamma_k^2 C}{2}$$

We sum for l between 0 and k:

$$\mathbb{E}[\sum_{l=0}^{k}\gamma_l(F(x_l)-F(x^*))] \leq -\frac{1}{2}\mathbb{E}[\|x_{k+1}-x^*\|^2]+\frac{1}{2}\mathbb{E}[\|x_0-x^*\|^2]+\sum_{l=0}^{k}\frac{\gamma_l^2 C}{2}$$

The result follows by convexity of F:

$$\mathbb{E}\Big[F(\bar{x}_l^\gamma)-F(x^*)\Big] \leq \frac{1}{\sum_{j=0}^{k}\gamma_j}\mathbb{E}[\sum_{l=0}^{k}\gamma_l(F(x_l)-F(x^*))] \leq \frac{\mathbb{E}[\|x_0-x^*\|^2]+C\sum_{l=0}^{k}\gamma_l^2}{2\sum_{j=0}^{k}\gamma_j} \qquad \square$$

5.3 Step size sequence

We know that $\mathbb{E}\Big[F(\bar{x}_l^\gamma)-F(x^*)\Big] \leq \frac{\mathbb{E}[\|x_0-x^*\|^2]+C\sum_{l=0}^{k}\gamma_l^2}{2\sum_{l=0}^{k}\gamma_l}$. A natural question is: which sequence (γ_k) should we take?
We would like $\sum_{j=1}^{k}\gamma_j \to +\infty$ and $\frac{\sum_{l=1}^{k}\gamma_l^2}{\sum_{j=1}^{k}\gamma_j} \to 0$. Such a sequence can be for instance taken as $\gamma_k = \frac{\gamma_0}{(k+1)^\alpha}$ with $0<\alpha<1$. Then,

$$\sum_{j=0}^{k}\gamma_j = \sum_{j=0}^{k}\frac{\gamma_0}{(j+1)^\alpha} \geq \sum_{j=0}^{k}\int_{j+1}^{j+2}\frac{\gamma_0}{t^\alpha}dt = \int_1^{k+2}\frac{\gamma_0}{t^\alpha}dt = \frac{\gamma_0}{1-\alpha}\Big[t^{1-\alpha}\Big]_1^{k+2} = \frac{\gamma_0}{1-\alpha}\Big((k+2)^{1-\alpha}-1\Big)$$

$$\sum_{j=0}^{k}\gamma_j^2 = \sum_{j=0}^{k}\frac{\gamma_0}{(j+1)^{2\alpha}} \leq \gamma_0+\sum_{j=1}^{k}\int_j^{j+1}\frac{\gamma_0}{t^{2\alpha}}dt = \gamma_0+\int_1^{k+1}\frac{\gamma_0}{t^{2\alpha}}dt = \begin{cases}\gamma_0(1+\ln(k+1)) & \text{if } \alpha=1/2\\ \gamma_0(1+\frac{(k+1)^{1-2\alpha}-1}{1-2\alpha}) & \text{if } \alpha\neq 1/2\end{cases}$$

We obtain the following cases:

	$\frac{1}{\sum_{j=0}^{k}\gamma_j}$	$\frac{\sum_{l=1}^{k}\gamma_l^2}{\sum_{j=1}^{k}\gamma_j}$
$0<\alpha<1/2$	$O\Big(\frac{1}{k^{1-\alpha}}\Big)$	$O\Big(\frac{1}{k^\alpha}\Big)$
$\alpha=1/2$	$O\Big(\frac{1}{k^{1/2}}\Big)$	$O\Big(\frac{\ln(k)}{k^{1/2}}\Big)$
$1/2<\alpha<1$	$O\Big(\frac{1}{k^{1-\alpha}}\Big)$	$O\Big(\frac{1}{k^{1-\alpha}}\Big)$

The best rate is obtained with $\alpha=1/2$, that is $\gamma_k=\frac{\gamma_0}{\sqrt{k}}$. With this choice, we have

$$\mathbb{E}[F(\bar{x}_k^\gamma)-F(x^*)] \in O\Big(\frac{\ln(k)}{\sqrt{k}}\Big).$$

Remark 5.3.1. If we know the number of iterations K we are going to perform, we can set a constant step size $\gamma_k=\frac{a}{\sqrt{K}}$ and obtain a guarantee $\mathbb{E}[F(\bar{x}_K^\gamma)-F(x^*)] \in O\Big(\frac{1}{\sqrt{K}}\Big)$

Remark 5.3.2. When F is μ-strongly convex, we can show that a step size decreasing as $\gamma_k=\frac{a}{\mu k}$ gives an improved rate $\mathbb{E}[F(\bar{x}_K^\gamma)-F(x^*)] \in O\Big(\frac{1}{\mu k}\Big)$.

5.4 Tradeoffs of large scale learning

We've seen in the previous section that when solving empirical risk minimization and when compared to gradient descent, the stochastic gradient descent method has a much better complexity per iteration but on the other hand, it requires more iterations to reach a given precision. There is indeed a tradeoff between those two quantities and knowing which algorithm is better suited to a given problem requires understanding what are the statistical benefits of increasing the size of the training set Bottou and Bousquet (2007).

Given the unknown probability distribution $\mathcal{D}$ of the i.i.d. dataset $(\xi_i)_{1\leq i\leq N}$ such that $\xi_i \sim \mathcal{D}$ for all i, we are interested in an ideal statistical estimator defined as

$$x^* \in \arg\min F(x) = \mathcal{E}_{\xi\sim\mathcal{D}}[f(x,\xi)]$$

As $\mathcal{D}$ is unknown, we resort to empirical risk minimization (ERM):

$$x_N^* \in \arg\min_{x\in X} F_N(x) = \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i)$$

Then, the optimization algorithm solving ERM return a point x_k that approximates x_n^*.
In total, the error which is made between x^* and x_k is

$$\mathbb{E}[F(x_k) - F(x^*)] = \underbrace{\mathbb{E}[F_N(x_k) - F_N(x_N^*)]}_{\text{optimisation error } \mathcal{E}_{\text{opt}}} + \underbrace{\mathbb{E}[F_N(x_N^*) - F(x^*)] + \mathbb{E}[F(x_k) - F_N(x_k)]}_{\text{estimation error } \mathcal{E}_{\text{est}}}$$

Statistical theory tells us that there exists a constant c such that $\mathcal{E}_{\text{est}} \leq c\sqrt{\frac{d}{N}}$, where d is the number of parameters, that is the dimension of the optimization variable x. We can argue that here is no need to search for an optimization error that would be much smaller that the estimation error. In the end, the total error would remain of the same order of magnitude and the gain would not compensate the effort. In the following table, we compare the cost of gradient descent and stochastic gradient when asking for a precision of the order of $\sqrt{\frac{d}{N}}$.

estimation	$\mathcal{E}_{\text{est}} \leq c\sqrt{\frac{d}{N}}$	
step size	gradient descent $\gamma = 1/L$	stochastic gradient $\gamma = \frac{a}{\sqrt{k}}$
optimization cost	$\mathcal{E}_{\text{opt}} \leq \frac{C_1}{k}$	$\mathcal{E}_{\text{opt}} \leq \frac{C_2}{\sqrt{k}}$
cost for 1 iteration	Nd	d
total cost for $\mathcal{E}_{\text{opt}} \approx \mathcal{E}_{\text{est}}$	$C_3 Nd\sqrt{\frac{N}{d}} = C_3 N^{3/2} d^{1/2}$	$C_4 d(\sqrt{\frac{N}{d}})^2 = C_4 N$

Table 5.1: Comparison of gradient descent and stochastic gradient descent when the number of samples N is large.

We can see that in the regime where we have many samples, even if stochastic gradient descent has a slower asymptotic rate than gradient descent, the total complexity of finding an estimator x_k with an error comparable to the estimation error is much lower for stochastic gradient.
However, a drawback of stochastic gradient descent is its sensitivity to the choice of the step size sequence $(\gamma_k)_{k\geq 0}$. The previous study only tells us how it should depend with respect to the number of iterations: indeed, setting it properly would require knowing the distance to the minimizer and we usually do not have this information. This can have a tremendous impact

on the actual behavior of the algorithm, so that intensive research has been done in order to find adaptive ways to set the step size sequence. A famous algorithm in this line of research is the ADAM algorithm Kingma and Ba (2015), often used for the resolution of neural network models.

5.5 Nonconvex objective*

Theorem 5.5.1. *Suppose that:*

- *$(x \mapsto f(x, \xi))$ is differentiable for all ξ with a L-Lipschitz gradient,*
- *there exists $C > 0$ such that $\mathbb{E}(\|\nabla f(x, \xi)\|^2) \leq C$ for all x,*
- *the sequence γ_k is deterministic.*

The iterates of the stochastic gradient algorithm $x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$ satisfy the convergence guarantee

$$\mathbb{E}\Big[\min_{0 \leq l \leq k} \|\nabla F(x_l)\|^2\Big] \leq \frac{2(F(x_0) - \inf F) + CL\sum_{l=0}^k \gamma_l^2}{2\sum_{l=0}^k \gamma_l} .$$

Proof. By Taylor-Lagrange inequality,

$$\begin{aligned} F(x_{k+1}) &\leq F(x_k) + \langle \nabla F(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &\leq F(x_k) - \gamma_k \langle \nabla F(x_k), \nabla f(x_k, \xi_{k+1}) \rangle + \frac{L\gamma_k^2}{2}\|\nabla f(x_k, \xi_{k+1})\|^2 \end{aligned}$$

where we use the fact that $x_{k+1} - x_k = -\gamma_k \nabla f(x_k, \xi_{k+1})$. We apply the conditional expectation $\mathbb{E}_k$:

$$\begin{aligned} \mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \gamma_k \|\nabla F(x_k)\|^2 + \frac{L}{2}\gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2] \\ \gamma_k \|\nabla F(x_k)\|^2 &\leq F(x_k) - \mathbb{E}_k[F(x_{k+1})] + \frac{L}{2}\gamma_k^2 C \end{aligned}$$

We then apply total expectation and sum for l between 0 and k

$$\mathbb{E}\Big[\sum_{l=0}^k \gamma_l \|\nabla F(x_l)\|^2\Big] \leq F(x_0) - \mathbb{E}[F(x_{k+1})] + \frac{L}{2}\sum_{l=0}^k \gamma_l^2 C$$

The result follows by remarking that $\|\nabla F(x_l)\|^2 \geq \min_{0 \leq l' \leq k} \|\nabla F(x_l')\|^2$ for all l and $\mathbb{E}[F(x_{k+1})] \geq \inf F$. □

Chapter 6

Dual problem

6.1 Lagrangian function

In this chapter, we consider the convex optimization problem

$$\begin{aligned} \text{minimize over } \mathbb{R}^n : \quad & f(x) \\ \text{under the constraints} : \quad & g(x) \preceq 0 \\ & A(x) = 0 \end{aligned} \tag{6.1.1}$$

($i.e.$ minimize $f(x)$ over $\mathbb{R}^n$, under the constraint $g(x) \preceq 0$ and $A(x) = 0$),
where $f : \mathbb{R}^n \to (-\infty, \infty]$ is convex and proper; $g(x) = (g_1(x), \dots, g_p(x))$, each $g_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function $(1 \le i \le p)$; $A : \mathbb{R}^n \to \mathbb{R}^m$ is an affine function and $\iota_{g \preceq 0} = \iota_{g^{-1}(\mathbb{R}^p_-)}$.
Using convex indicator functions

$$\iota_C(x) = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{if } x \notin C \end{cases}$$

this can also be written as

$$\text{minimize over } \mathbb{R}^n : \quad f(x) + \iota_{g \preceq 0}(x) + \iota_{A=0}(x) \ . \tag{6.1.2}$$

Under these conditions, the function $x \mapsto f(x) + \iota_{g \preceq 0}(x) + \iota_{A=0}(x)$ is convex.

Definition 6.1.1 (primal value, primal optimal point)**.** The **primal value** associated to (6.1.2) is the infimum

$$p = \inf_{x \in \mathbb{R}^n} f(x) + \iota_{g \preceq 0}(x) + \iota_{A=0}(x).$$

A point $x^* \in \mathbb{R}^n$ is called **primal optimal** if

$$p = f(x^*) + \iota_{g \preceq 0}(x^*) + \iota_{A=0}(x^*).$$

Notice that, under our assumption, $p \in [-\infty, \infty]$. Also, there is no guarantee about the existence of a primal optimal point, and no guarantee that the primal value is attained either.

Since (6.1.2) may be difficult to solve, it is useful to see this as an 'inf sup' problem, and solve a 'sup inf' problem instead (see definition 6.2.1 below). To make this precise, we introduce the Lagrangian function.

Definition 6.1.2. The **Lagrangian function** associated to problem (6.1.2) is the function

$$\begin{aligned} L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p &\longrightarrow [-\infty, +\infty] \\ (x, \phi_E, \phi_I) &\mapsto f(x) + \langle \phi_E, A(x) \rangle + \langle \phi_I, g(x) \rangle - \iota_{\mathbb{R}^p_+}(\phi_I) \end{aligned}$$

(where $\mathbb{R}^p_+ = \{\phi \in \mathbb{R}^p, \phi \succeq 0\}$).

The link with the initial problem comes next:

Lemma 6.1.1 (constrained objective as a supremum)**.** *The constrained objective is the supremum (over $\phi = (\phi_E, \phi_I)$) of the Lagrangian function,*

$$\forall x \in \mathbb{R}^n, \;\; f(x) + \iota_{g \preceq 0}(x) + \iota_{A=0}(x) = \sup_{\phi \in \mathbb{R}^{m+p}} L(x, \phi)$$

Proof. We give the proof in the case $m = 0$ (only inequality constraints). The general case is similar.
Distinguish the cases $g(x) \preceq 0$ and $g(x) \npreceq 0$.
(a) If $g(x) \npreceq 0, \exists i \in \{1, \dots, p\} : g_i(x) > 0$. Choosing $\phi_t = te_i$ (where $\mathbf{e} = (e_1, \dots, e_p)$ is the canonical basis of $\mathbb{R}^p$)), $t \geq 0$, then $\lim_{t\to\infty} L(x, \phi_t) = +\infty$, whence $\sup_{\phi \in \mathbb{R}^p_+} L(x, \phi) = +\infty$. On the other hand, in such a case, $\iota_{g \leq 0}(x) = +\infty$, whence the result.

(b) If $g(x) \preceq 0$, then $\forall \phi \in \mathbb{R}^p_+, \langle \phi, g(x) \rangle \leq 0$, and the supremum is attained at $\phi = 0$. Whence, $\sup_{\phi \succeq 0} L(x, \phi) = \sup_{\phi \in \mathbb{R}^p} L(x, \phi) = f(x)$.
On the other hand, $\iota_{g \preceq 0}(x) = 0$, so $f(x) + \iota_{g \preceq 0}(x) = f(x)$. The result follows. □

Equipped with lemma 6.1.1, the primal value associated to problem (6.1.2) writes

$$p = \inf_{x \in \mathbb{R}^n} \sup_{\phi \in \mathbb{R}^{m+p}} L(x, \phi). \tag{6.1.3}$$

One natural idea is to exchange the order of inf and sup in the above problem. Before proceeding, the following simple lemma allows to understand the consequence of such an exchange.

Proposition 6.1.1. *Let $F : A \times B \to [-\infty, \infty]$ any function. Then,*

$$\sup_{y \in B} \inf_{x \in A} F(x, y) \leq \inf_{x \in A} \sup_{y \in B} F(x, y).$$

Proof. $\forall (\bar{x}, \bar{y}) \in A \times B$,

$$\inf_{x \in A} F(x, \bar{y}) \leq F(\bar{x}, \bar{y}) \leq \sup_{y \in B} F(\bar{x}, y).$$

Taking the supremum over $\bar{y}$ in the left-hand side we still have

$$\sup_{\bar{y} \in B} \inf_{x \in A} F(x, \bar{y}) \leq \sup_{y \in B} F(\bar{x}, y).$$

Now, taking the infimum over $\bar{x}$ in the right-hand side yields

$$\sup_{\bar{y} \in B} \inf_{x \in A} F(x, \bar{y}) \leq \inf_{\bar{x} \in A} \sup_{y \in B} F(\bar{x}, y).$$

up to a simple change of notation, this is the expected result. □

6.2 Dual problem

Definition 6.2.1 (Dual problem, dual function, dual value)**.**
The **dual value** associated to (6.1.3) is

$$d = \sup_{\phi \in \mathbb{R}^{m+p}} \inf_{x \in \mathbb{R}^n} L(x, \phi_I, \phi_E).$$

The function

$$\mathcal{D}(\phi) = \inf_{x \in \mathbb{R}^n} L(x, \phi)$$

is called the **Lagrangian dual function**. Thus, the **dual problem** associated to the primal problem (6.1.2) is

$$\text{maximize over } \mathbb{R}^{m+p} : \quad \mathcal{D}(\phi).$$

A vector $\lambda \in \mathbb{R}^p_+$ is called **dual optimal** if

$$d = \mathcal{D}(\lambda).$$

Without any further assumption, there is no reason for the two values (primal and dual) to coincide. However, as a direct consequence of Proposition 6.1.1, we have :

Proposition 6.2.1 (Weak duality)**.** *Let p and d denote respectively the primal and dual value for problem (6.1.2). Then,*

$$d \leq p\ .$$

Proof. Apply Proposition 6.1.1. □

Definition 6.2.2 (Saddle point)**.** Let $F : A \times B \to [-\infty, \infty]$ any function, and A, B two sets. The point $(x^*, y^*) \in A \times B$ is called a **saddle point** of F if, for all $(x, y) \in A \times B$,

$$F(x^*, y) \leq F(x^*, y^*) \leq F(x, y^*).$$

Proposition 6.2.2. *Let $F : A \times B \to [-\infty, \infty]$. F has a saddle point (x^*, y^*) if and only if*

$$\sup_{y \in B} \inf_{x \in A} F(x, y) = \inf_{x \in A} F(x, y^*) = F(x^*, y^*) = \sup_{y \in B} F(x^*, y) = \inf_{x \in A} \sup_{y \in B} F(x, y).$$

Proof. Suppose F has a saddle point (x^*, y^*). As $\forall y$, $F(x^*, y) \leq F(x^*, y^*)$, we take the supremum in y to get $\sup_{y \in B} F(x^*, y) \leq F(x^*, y^*)$.

$$\begin{aligned}
\sup_{y \in B} \inf_{x \in A} F(x, y) &\leq \inf_{x \in A} \sup_{y \in B} F(x, y) && \text{(Prop. 6.1.1)} \\
&\leq \sup_{y \in B} F(x^*, y) && \text{(def of inf)} \\
&\leq F(x^*, y^*) && \text{(saddle point)} \\
&\leq \inf_{x \in A} F(x, y^*) && \text{(saddle point)} \\
&\leq \sup_{y \in B} \inf_{x \in A} F(x, y) && \text{(def of sup)}
\end{aligned}$$

Hence all inequalities are equalities and we get the result.
The converse implication is straightforward using the two inner equalities. □

Chapter 7

Strong duality theorem

7.1 Fenchel-Legendre Conjugate*

Definition 7.1.1. Let $f : \mathcal{X} \to [-\infty, +\infty]$. The **Fenchel-Legendre conjugate** of f is the function $f^* : \mathcal{X} \to [-\infty, \infty]$, defined by

$$f^*(\phi) = \sup_{x \in \mathcal{X}} \langle \phi, x \rangle - f(x)\,, \qquad \forall \phi \in \mathcal{X}.$$

Notice that

$$f^*(0) = -\inf_{x \in \mathcal{X}} f(x).$$

Figure 7.1 provides a graphical representation of f^*. You should get the intuition that, in the differentiable case, if the maximum is attained in the definition of f^* at point x_0, then $\phi = \nabla f(x_0)$, and $f^*(\phi) = \langle \nabla f(x_0), x_0 \rangle - f(x_0)$. This intuition will be proved correct in proposition 7.1.2.

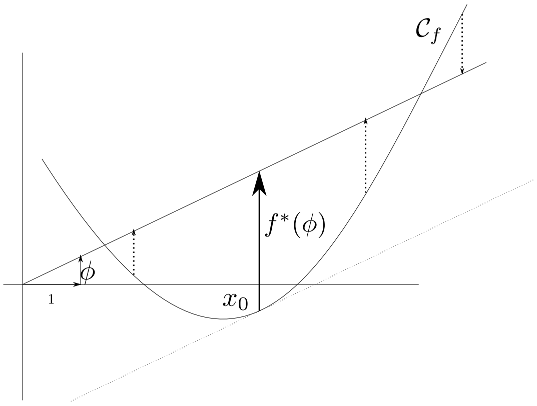

Figure 7.1: Fenchel Legendre transform of a smooth function f. The maximum positive difference between the line with slope $\tan(\phi)$ and the graph $\mathcal{C}_f$ of f is reached at x_0.

Exercise 7.1.1.
Prove the following statements.
General hint: If $h_\phi : x \mapsto \langle \phi, x \rangle - f(x)$ reaches a maximum at x^*, then $f^*(\phi) = h_\phi(x^*)$. Furthermore, h_ϕ is concave (if f is convex). If h_ϕ is differentiable, it is enough to find a zero of its gradient to obtain a maximum.
Indeed, $x \in \arg\min(-h_\phi) \Leftrightarrow 0 \in \partial(-h_\phi)$, and, if $-h_\phi$ is differentiable, $\partial(-h_\phi) = \{-\nabla h_\phi\}$.

1. If $\mathcal{X} = \mathbb{R}$ and f is a quadratic function (of the kind $f(x) = (x-a)^2 + b$), then f^* is also quadratic.
2. In $\mathbb{R}^n$, let A by a symmetric, definite positive matrix and $f(x) = \langle x, Ax \rangle$ (a quadratic function). Show that f^* is also quadratic.
3. $f : \mathcal{X} \to [-\infty, +\infty]$. Show that $f = f^* \Leftrightarrow f(x) = \frac{1}{2}\|x\|^2$.

   Hint: For the 'if' part: show first that $f(\phi) \geq \langle \phi, \phi \rangle - f(\phi)$.

   Then, show that $f(\phi) \leq \sup_x \langle \phi, x \rangle - \frac{1}{2}\|x\|^2$. Conclude.
4. $\mathcal{X} = \mathbb{R}$, for

   $$f(x) = \begin{cases} 1/x & \text{if } x > 0; \\ +\infty & \text{otherwise .} \end{cases}$$

   we have,

   $$f^*(\phi) = \begin{cases} -2\sqrt{-\phi} & \text{if } \phi \leq 0; \\ +\infty & \text{otherwise .} \end{cases}$$
5. $\mathcal{X} = \mathbb{R}$, if $f(x) = \exp(x)$, then

   $$f^*(\phi) = \begin{cases} \phi \ln(\phi) - \phi & \text{if } \phi > 0; \\ 0 & \text{if } \phi = 0; \\ +\infty & \text{if } \phi < 0. \end{cases}$$

Notice that, if $f(x) = -\infty$ for some x, then $f^* \equiv +\infty$.
Nonetheless, under 'reasonable' conditions on f, the Legendre transform enjoys nice properties, and even f can be recovered from f^* (through the equality $f = f^{**}$).

Proposition 7.1.1 (Properties of f^*)**.**
Let $f : \mathcal{X} \to [-\infty, +\infty]$ be any function.

1. *f^* is always convex, and l.s.c.*
2. *If $\operatorname{dom} f \neq \emptyset$, then $-\infty \notin f^*(\mathcal{X})$*
3. *If f is convex and proper, then f^* is convex, l.s.c., proper.*

Proof.

1. Fix $x \in \mathcal{X}$ and consider the function $h_x : \phi \mapsto \langle \phi, x \rangle - f(x)$. From the definition, $f^* = \sup_{x \in \mathcal{X}} h_x$. Each h_x is affine, whence convex. Using proposition 2.1.2, f^* is also convex. Furthermore, each h_x is continuous, whence l.s.c, so that its epigraph is closed. Lemma 2.4.3 thus shows that f^* is l.s.c.
2. From the hypothesis, there is an x_0 in $\operatorname{dom} f$. Let $\phi \in \mathcal{X}$. The result is immediate:

   $$f^*(\phi) \geq h_{x_0}(\phi) = f(x_0) - \langle \phi, x_0 \rangle > -\infty.$$
3. In view of points 1. and 2., it only remains to show that $f^* \not\equiv +\infty$. Let $x_0 \in \operatorname{relint}(\operatorname{dom} f)$. According to proposition 2.5.2, there exists a subgradient ϕ_0 of f at x_0. Moreover, since f is proper, $f(x_0) < \infty$. From the definition of a subgradient,

   $$\forall x \in \operatorname{dom} f, \langle \phi_0, x - x_0 \rangle \leq f(x) - f(x_0).$$

Whence, for all $x \in \mathcal{X}$,

$$\langle \phi_0, x \rangle - f(x) \leq \langle \phi_0, x_0 \rangle - f(x_0),$$

thus, $\sup_x \langle \phi_0, x \rangle - f(x) \leq \langle \phi_0, x_0 \rangle - f(x_0) < +\infty$.

Therefore, $f^*(\phi_0) < +\infty$. □

Proposition 7.1.2 (Fenchel-Young). *Let $f : \mathcal{X} \to [-\infty, \infty]$. For all $(x, \phi) \in \mathcal{X}^2$, the following inequality holds:*

$$f(x) + f^*(\phi) \geq \langle \phi, x \rangle ,$$

With equality if and only if $\phi \in \partial f(x)$.

Proof. The inequality is an immediate consequence of the definition of f^*. The condition for equality to hold (*i.e.*, for the converse inequality to be valid), is obtained with the equivalence

$$f(x) + f^*(\phi) \leq \langle \phi, x \rangle \;\Leftrightarrow\; \forall y,\; f(x) + \langle \phi, y \rangle - f(y) \leq \langle \phi, x \rangle \;\Leftrightarrow\; \phi \in \partial f(x)\,.$$

□

Exercise 7.1.2. * Let $f : \mathcal{X} \to (-\infty, +\infty]$ a proper, convex, l.s.c. function. Show that

$$\partial(f^*) = (\partial f)^{-1}$$

where, for $\phi \in \mathcal{X}$, $(\partial f)^{-1}(\phi) = \{x \in \mathcal{X} : \; \phi \in \partial f(x)\}$.
Hint: Use Fenchel-Young inequality to show one inclusion, and the property $f = f^{**}$ for the other one.

7.2 Equality constraints

Theorem 7.2.1. *Let A be an affine function and f be a convex function. Let us consider the problem with equality constraints*

$$\min_{x \in \mathbb{R}^n} f(x) + \iota_{A=0}(x),$$

the associated Lagrangian

$$L(x, \phi) = f(x) + \langle \phi, A(x) \rangle$$

and the dual problem

$$\sup_{\phi \in \mathbb{R}^m} \mathcal{D}(\phi)$$

where $\mathcal{D}(\phi) = \inf_{x \in \mathbb{R}^n} f(x) + \langle \phi, A(x) \rangle$.
If $0 \in \operatorname{relint}(A(\operatorname{dom} f))$ (constraint qualification condition), then

1. $\inf_{x \in \mathbb{R}^n} f(x) + \iota_{A=0}(x) < +\infty$

2. $\inf_{x \in \mathbb{R}^n} f(x) + \iota_{A=0}(x) = \sup_{\phi \in \mathbb{R}^m} \mathcal{D}(\phi)$ *(*i.e.*, the duality gap is zero).*

3. *(Dual attainment at some λ):*

$$\exists \lambda \in \mathbb{R}^m, \textit{such that } \; d = \mathcal{D}(\lambda).$$

If moreover, $\exists x^* \in \mathbb{R}^n$ such that $\min_{x \in \mathbb{R}^n} f(x) + \iota_{A=0}(x) = f(x^*) + \iota_{A=0}(x^*)$, then (x^*, λ) is a saddle point of L.

Note that if $\operatorname{dom} f = \mathbb{R}^n$, then the constraints are automatically qualified.

**Proof of the theorem, introduction.* This proof is a bit technical and should be read only by readers interested in understanding theory deeper.
The idea of the proof is to apply Proposition 2.5.2 on the value function

$$\mathcal{V}(b) = \inf_{x\in\mathbb{R}^n} f(x) + \iota_{\{b\}}(A(x)).$$

Note that $\mathcal{V}(0)$ is the value of the primal problem. $\mathcal{V}$ is convex since it is the infimum of a jointly convex function (Proposition 2.1.3). We are now going to compute $\mathcal{V}^*$ and see its link with the dual function.

Lemma 7.2.1. *For all $\phi \in \mathbb{R}^m$, $\mathcal{V}^*(-\phi) = -\mathcal{D}(\phi)$.*

Proof. For $\phi \in \mathbb{R}^m$, by definition of the Fenchel conjugate,

$$\begin{aligned}\mathcal{V}^*(-\phi) &= \sup_{y\in\mathbb{R}^m} \langle -\phi, y\rangle - \mathcal{V}(y)\\ &= \sup_{y\in\mathbb{R}^m} \langle -\phi, y\rangle - \inf_{x\in\mathbb{R}^n} \left[f(x) + \iota_{\{y\}}(A(x))\right]\\ &= \sup_{y\in\mathbb{R}^m} \langle -\phi, y\rangle + \sup_{x\in\mathbb{R}^n} \left[- f(x) - \iota_{\{y\}}(A(x))\right]\\ &= \sup_{y\in\mathbb{R}^m} \sup_{x\in\mathbb{R}^n} \langle -\phi, y\rangle - f(x) - \iota_{\{y\}}(A(x))\\ &= \sup_{x\in\mathbb{R}^n} \left[\sup_{y\in\mathbb{R}^m} \underbrace{\langle -\phi, y\rangle - \iota_{\{y\}}(A(x))}_{\varphi_x(y)}\right] - f(x). \end{aligned} \tag{7.2.1}$$

For a fixed $x \in \operatorname{dom} f$, consider the function $\varphi_x : y \mapsto \langle -\phi, y\rangle - \iota_{\{y\}}(A(x))$. As

$$\varphi(y) = \begin{cases} -\infty & \text{if } y \neq A(x)\\ \langle -\phi, A(x)\rangle & \text{otherwise,}\end{cases}$$

(7.2.1) becomes

$$\begin{aligned}\mathcal{V}^*(-\phi) &= \sup_{x\in\mathbb{R}^n} \langle -\phi, A(x)\rangle - f(x) = - \inf_{x\in\mathbb{R}^n} \underbrace{f(x) + \langle \phi, A(x)\rangle}_{L(x,\phi)}\\ &= -\mathcal{D}(\phi)\end{aligned}$$

□

Corollary 7.2.1. *The dual function $\mathcal{D}$ is concave and upper semi-continuous.*

Proof. Proposition 7.1.1. □

Proof of the theorem, continued. From Lemma 7.2.1, we deduce

$$\begin{aligned}\mathcal{V}^{**}(0) &= \sup_{\phi\in\mathbb{R}^p} -\mathcal{V}^*(\phi) = \sup_{\phi\in\mathbb{R}^p} -\mathcal{V}^*(-\phi) \quad \text{(by symmetry of } \mathbb{R}^p\text{)}\\ &= \sup_{\phi\in\mathbb{R}^p} \mathcal{D}(\phi)\end{aligned}$$

Hence, $\mathcal{V}^{**}(0)$ is the value of the dual problem. We have, $\mathcal{V}(0) \geq \mathcal{V}^{**}(0)$. Said otherwise, $\inf_{x\in\mathbb{R}^n} f(x) + \iota_{A=0}(x) \geq \sup_{\phi\in\mathbb{R}^p} \mathcal{D}(\phi)$ and we recover weak duality.

Now remark that $\operatorname{dom}\mathcal{V} = \{b \in \mathbb{R}^m \ : \ \exists x \in \operatorname{dom} f, A(x) = b\} = A(\operatorname{dom} f)$. So the constraint qualification condition $0 \in \operatorname{relint}(A(\operatorname{dom} f))$ is equivalent to $0 \in \operatorname{relint}(\operatorname{dom}\mathcal{V})$ and we can apply Proposition 2.5.2: $\partial\mathcal{V}(0) \neq \emptyset$.

To show the dual attainment, we take $\lambda \in \partial\mathcal{V}(0) \neq \emptyset$. Equality in Fenchel-Young (Proposition 7.1.2) writes: $\mathcal{V}(0) + \mathcal{V}^*(\lambda) = \langle \lambda, 0\rangle = 0$. Thus, we have

$$\begin{aligned}\mathcal{V}(0) &= -\mathcal{V}^*(\lambda)\\ &= \mathcal{D}(-\lambda)\\ &\leq \sup_\phi \mathcal{D}(\phi) = \mathcal{V}^{**}(0) \leq \mathcal{V}(0)\end{aligned}$$

Hence, all the inequalities are equalities:

$$\inf_{x\in\mathbb{R}^n} f(x) + \iota_{A=0}(x) = \mathcal{V}(0) = -\mathcal{V}^*(\lambda) = \mathcal{D}(-\lambda) = \sup_\phi \mathcal{D}(\phi).$$

This shows that the duality gap is 0 and that $-\lambda$ is a dual optimum.

Now, if there exists a primal optimum x^*,

$$\mathcal{V}(0) = f(x^*) + \iota_{A=0}(x^*) = \sup_{\phi\in\mathbb{R}^m} L(x^*, \phi) \geq L(x^*, -\lambda) \geq \inf_x L(x, -\lambda) = \mathcal{D}(-\lambda)$$

and we conclude using the fact that $\mathcal{V}(0) = \mathcal{D}(-\lambda)$ and Proposition 6.2.2. □

Remark 7.2.1. The proof shows that the negative subgradients of the value function at 0 are optimal dual points. Hence, we can interpret the optimal dual points as the sensitivity of the primal objective to changes in the constraint.

7.3 Inequality constraints

Theorem 7.3.1. *Let us consider the problem with inequality constraints*

$$\min_{x\in\mathbb{R}^n} f(x) + \iota_{g\leq 0}(x),$$

the associated Lagrangian

$$L(x,\phi) = f(x) + \langle \phi, g(x)\rangle - \iota_{\mathbb{R}^p_+}(\phi) \tag{7.3.1}$$

and the dual problem

$$\sup_{\phi\in\mathbb{R}^p} \mathcal{D}(\phi)$$

where $\mathcal{D}(\phi) = \inf_{x\in\mathbb{R}^n} f(x) + \langle\phi, g(x)\rangle - \iota_{\mathbb{R}^p_+}(\phi)$.

If $\exists x_0 \in \operatorname{dom} f$ *such that for all* j, $g_j(x_0) < 0$ *(Slater's constraint qualification condition), then*

1. $\inf_{x\in\mathbb{R}^n} f(x) + \iota_{g\leq 0}(x) < +\infty$
2. $\inf_{x\in\mathbb{R}^n} f(x) + \iota_{g\leq 0}(x) = \sup_{\phi\in\mathbb{R}^p} \mathcal{D}(\phi)$ *(*i.e., *the duality gap is zero).*
3. *(Dual attainment at some* λ*):*

$$\exists\lambda \in \mathbb{R}^p_+, \textit{such that } \ d = \mathcal{D}(\lambda).$$

If moreover, $\exists x^* \in \mathbb{R}^n$ *such that* $\min_{x\in\mathbb{R}^n} f(x) + \iota_{g\leq 0}(x) = f(x^*) + \iota_{g\leq 0}(x^*)$*, then* (x^*, λ) *is a saddle point of* L.

**Proof.* The proof is similar to the equality case. The main difference is in the domain of the value function:

$$\operatorname{dom} \mathcal{V} = \{b \in \mathbb{R}^p \ : \ \exists x \in \operatorname{dom} f, g(x) \leq b\}.$$

Slater's condition is exactly saying that $0 \in \operatorname{int} \operatorname{dom} \mathcal{V}$. □

Theorem 7.3.2 (Karush-Kuhn-Tucker conditions)**.** *If (x^*, ϕ^*) is a saddle point of the Lagrangian function (7.3.1) of the optimization problem with inequality constraints, then*

$$\begin{aligned} &0 \in \sum_{j=1}^p \phi_j^* \partial g_j(x^*) + \partial f(x^*) \\ &g(x^*) \leq 0 \ , \qquad \phi^* \geq 0 \ , \qquad \langle \phi^*, g(x^*) \rangle = 0 \end{aligned}$$

Proof. The Karush-Kuhn-Tucker conditions can be recovered by writing Fermat's rule for the inf-sup conditions and the fact that $\operatorname{dom} g_j = \mathbb{R}^n$ for all j:

$$\begin{aligned} &0 \in \partial_x L(x^*, \phi^*) = \sum_{j=1}^p \phi_j^* \partial g_j(x^*) + \partial f(x^*) \\ &0 \in \partial_\phi(-L)(x^*, \phi^*) = -g(x^*) + \partial \iota_{\mathbb{R}^p_+}(\phi^*). \end{aligned}$$

For this second condition, we need to compute $\partial \iota_{\mathbb{R}^p_+}$. First note that for all $\phi \in \mathbb{R}^p$, $\iota_{\mathbb{R}^p_+}(\phi) = \sum_{j=1}^p \iota_{\mathbb{R}_+}(\phi_j)$ and so $\partial \iota_{\mathbb{R}^p_+}(\phi) = \partial \iota_{\mathbb{R}_+}(\phi_1) \times \ldots \times \partial \iota_{\mathbb{R}_+}(\phi_n)$.
It is a good exercise to show that

$$\partial \iota_{\mathbb{R}_+}(\phi_j) = \begin{cases} \{0\} & \text{if } \phi_j > 0, \\ \mathbb{R}_- & \text{if } \phi_j = 0, \\ \emptyset & \text{if } \phi_j < 0. \end{cases}$$

We obtain that $\forall j$, $g_j(x^*) = 0$ if $\phi_j^* > 0$, $g_j(x^*) \leq 0$ if $\phi_j^* = 0$ and that ϕ_j^* is never strictly negative. This can be written as $g_j(x^*)\phi_j^* = 0$ for all j. □

Exercise 7.3.1 (Examples of duals, Borwein and Lewis (2006), chap.4)**.** Compute the dual of the following problems. In other words, calculate the dual function $\mathcal{D}$ and write the problem of maximizing the latter as a convex minimization problem.

1. Linear program

   $$\inf_{x \in \mathbb{R}^n} \langle c, x \rangle$$
   $$\text{under constraint } Gx \preceq b$$

   where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^p$ and $G \in \mathbb{R}^{p \times n}$.

   Hint: you should find that the dual problem is again a linear program, with equality constraints.

2. Quadratic program

   $$\inf_{x \in \mathbb{R}^n} \frac{1}{2} \langle x, Cx \rangle$$
   $$\text{under constraint } Gx \preceq b$$

   where C is symmetric, positive, definite.

   Hint : you should obtain an unconstrained quadratic problem.

- Assume in addition that the constraints are linearly independent, *i.e.* $\operatorname{rank}(G) = p$, *i.e.* $G = \begin{pmatrix} w_1^\top \\ \vdots \\ w_p^\top \end{pmatrix}$, where $(w_1, \ldots, w_p)$ are linearly independent. Compute then the dual value.

Dictionnaire français-anglais pour l'optimisation

FRANÇAIS	ENGLISH
optimisation	optimization
convexe	convex
propre	proper
semi-continue inférieurement (s.c.i)	lower semi-continuous (l.s.c)
gradient	gradient
sous-gradient	sub-gradient
positif	nonnegative
négatif	nonpositive
contrainte d'égalité	equality constraint
recherche linéaire	line search
opérateur proximal	proximal operator
pas (de gradient)	step size
lagrangien augmenté	augmented Lagrangian
théorème de dérivation des fonctions composées	chain rule
méthode d'éclatement	splitting method
dualité forte	strong duality
saut de dualité	duality gap
point selle	saddle point
programmation linéaire	linear programming

List of symbols

$\liminf$	Limit inferior, page 15
f^*	Fenchel-Legendre conjugate of function f, page 39
l.s.c.	Lower semicontinuous, page 15

Bibliography

Bauschke, H. H. and Combettes, P. L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer. 17

Borwein, J. and Lewis, A. (2006). *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS Books in Mathematics. Springer. 44

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. *Advances in neural information processing systems*, 20. 34

Boyd, S. and Vandenberghe, L. (2009). *Convex optimization*. Cambridge university press. 9

Brezis, H. (1987). Analyse fonctionnelle, 2e tirage.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. 35

Macchi, O. and Eweda, E. (1983). Second-order convergence analysis of stochastic adaptive linear filtering. *IEEE Transactions on Automatic Control*, 28(1):76–85. 32

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer. 9, 30

Rockafellar, R. T., Wets, R. J.-B., and Wets, M. (1998). *Variational analysis*, volume 317. Springer.