Basics of Applied Mathematics - Part III: Optimization

Preliminary Draft Version, from February 7, 2025

Léo Simpson and Moritz Diehl
Department of Microsystems Engineering and Department of Mathematics,
University of Freiburg, Germany
moritz.diehl@imtek.uni-freiburg.de

Preface

This script is designed for the course Basics of Applied Mathematics (BAM) - Optimization, for students pursuing a master's of mathematics that focuses on data and technologies.

The script is quite self-contained. Nevertheless, to follow this course, one must already have good mathematics fundamentals.

An important note is that this script is largely based on the book [1].

Contents

Chapter 1

Optimization problems: Definitions and Applications

1.1 Optimization in the real world

There are two contexts where optimization problems arise: Decision making, and model learning:

- In *decision making*, we are faced with a set of possible decisions, and we want to find the decision that minimizes some cost. This decision can be made based on the solution to some optimization problem.
- In *model learning*, a set of data is available, and we want to find a model that fits the data. This model can be found by solving an optimization problem.

In this course, we will mainly focus on the applications arising in model learning. In the rest of the section, we will introduce the basic idea of formulating a model learning problem as an optimization problem.

The typical optimization problem in data analysis is to find a model that agrees with some collected data but adheres to some structural constraints that reflect our beliefs about what a good model should be. The data set is typically a collection of inputs and outputs corresponding to different samples:

$$\mathcal{D} := \{(a_1, y_1), \dots, (a_m, y_m)\}, \tag{1.1}$$

where a_i is the input (also sometimes called *features*) and y_i is the output (also sometimes called *measurements*).

The goal is to find a model that predicts the output y given the input a. This typically takes the form of a function φ that maps inputs to outputs:

$$y \approx \varphi(a). \tag{1.2}$$

To define the problem mathematically, we need to parameterize the possible model functions φ with some unknown parameters $x \in \mathbb{R}^n$. The fitting problem can then be formulated as finding some $x \in \mathbb{R}^n$ such that for any input a, the input can be predicted by the model:

$$y \approx \varphi(a; x). \tag{1.3}$$

To formulate this as an optimization problem, we define a loss function $\mathcal{L}_{\mathcal{D}}(x)$ that measures how well the model fits the data:

$$\mathcal{L}_{\mathcal{D}}(x) \coloneqq \frac{1}{m} \sum_{i=1}^{m} l\left(y_i, \varphi(a_i; x)\right), \tag{1.4}$$

where $l(y, \bar{y})$ represents some distance between the true output y and the predicted output $\bar{y}$.

The goal is to find the parameter x that best fits the data while meeting some prior assumptions on the model. This is typically done by solving the optimization problem:

$$\underset{x \in \mathbb{R}^n}{\operatorname{minimize}} \quad \mathcal{L}_{\mathcal{D}}(x) + \lambda \ \operatorname{pen}(x) \tag{1.5}$$

where $\operatorname{pen}(x)$ is a penalty function that measures how well the model meets the constraints. The parameter $\lambda \geq 0$ is the *regularization parameter*, a hyperparameter that controls the trade-off between fitting the data and meeting the constraints.

Depending on the nature of the labels y_j, the model-fitting task takes different names:

- When y_j are *real numbers*, or *vectors of real numbers*, the task is called a *regression problem.*
- When y_j are *labels*, i.e. integers in a set $\{1, \ldots, q\}$, the task is called a *classification problem.*

Later, we will look at several examples of regression and classification problems.

1.2 General definitions and notations

1.2.1 Formulation of optimization problems

Definition 1.1: Optimization Problems

An optimization problem is mathematically formulated as follows:

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad f(x) \tag{1.6}$$

In (1.6), three "ingredients" are present:

- The *decision variable* $x \in \mathbb{R}^n$ that can be chosen, and that may contain several components.
- The *feasible set* $\mathcal{X}$ in which the decision variable x is imposed to be. Often, we will choose $\mathcal{X} = \mathbb{R}^n$. In this case, the optimization is qualified as an *unconstrained* problem.
- An *objective function*, $f(x) : \mathcal{X} \to \mathbb{R}$, that shall be minimized. Note that when a function $\tilde{f}(x)$ shall be maximized, one can minimize the function $f(x) \equiv -\tilde{f}(x)$.

Remark. Sometimes, an alternative formulation might be found, that accounts more explicitly for the constraint $x \in \mathcal{X}$:

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \\ &\text{subject to} \quad x \in \mathcal{X}. \end{aligned} \tag{1.7}$$

1.2.2 Minimizers

Definition 1.2: Global optimality

The point $x^\star \in \mathbb{R}^n$ is called a "global minimizer" (often also called a "global minimum") when $x^\star \in \mathcal{X}$ and $\forall x \in \mathcal{X} : \ f(x) \geq f(x^\star)$.

Definition 1.3: Strict optimality

The point $x^\star \in \mathbb{R}^n$ is called a "*strict* global minimizer" when $x^\star \in \mathcal{X}$ and $\forall x \in \mathcal{X} \setminus \{x^\star\} : \ f(x) > f(x^\star)$.

1.2.3 Local optimality

Definition 1.4: Local optimality

The point $x^\star \in \mathbb{R}^n$ is called a "local minimizer" when $x^\star \in \mathcal{X}$ and there exists a neighborhood of $\mathcal{N}$ of $x^\star$ such that $\forall x \in \mathcal{X} \cap \mathcal{N} : \ f(x) \geq f(x^\star)$.

Note that this neighborhood can be chosen to be in the form of an open ball: $\mathcal{N} := \{x \mid \|x - x^\star\| < \varepsilon\}$ for some $\varepsilon > 0$.

Definition 1.5: Strict local optimality

The point $x^\star \in \mathbb{R}^n$ is called a "*strict* local minimizer" when $x^\star \in \mathcal{X}$ and there exists a neighborhood $\mathcal{N}$ of $x^\star$ so that $\forall x \in (\mathcal{X} \cap \mathcal{N}) \setminus \{x^\star\} : \ f(x) > f(x^\star)$.

Remark. Note that a global minimizer is also a local minimizer, but the converse is not necessarily true.

Example 1.1: (Unconstrained) Quadratic Program

A Quadratic Program (QP) is an optimization problem where the objective function is quadratic. In the unconstrained case (i.e. $\mathcal{X} = \mathbb{R}^n$), the problem is formulated as follows:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} x^T Q x - c^T x + r, \tag{1.8}$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $c \in \mathbb{R}^n$ is a vector, and $r \in \mathbb{R}$ is a scalar.

Remark. The Hessian of the objective function of (1.8) is constant, and equal to Q:

$$\forall x \in \mathbb{R}^n \quad \nabla^2 f(x) = Q. \tag{1.9}$$

1.3 Examples of optimization problems in data analysis

1.3.1 Regression problems

The most common optimization problem in data analysis is the least squares problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m} \sum_{j=1}^{m} \|y_j - \varphi(a_j; x)\|^2, \tag{1.10}$$

where the outputs $y_j \in \mathbb{R}^p$ are vectors, and the inputs a_j can be some matrices or vectors.

For a general function φ, the optimization problem (1.10) is often called “non-linear least squares”. Since it is quite general, it is quite difficult to analyze: it might have multiple local minima, it might have no global minimum at all, etc.

There is, however, a special case where the analysis is way easier: the *linear least squares problem*. This corresponds to the case where the model φ is affine in the parameters x.

Example 1.2: The linear least squares problem

When the model takes the form: $\varphi(a_j; x) = A(a_j)x + b(a_j)$, then the optimization problem (1.10) takes the following form:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m} \|\tilde{y}_j - A_j x\|^2 =: f(x), \tag{1.11}$$

where $\tilde{y}_j := y_j - b(a_j)$ and $A_j := A(a_j)$. As a small abuse of notation, we might write y_j instead of $\tilde{y}_j$ in the next parts.

Proposition 1.1: Linear least squares is a QP

The linear least squares problem (1.11) is a quadratic optimization problem in the form (1.8), with:

$$Q := \frac{1}{m}\sum_{j=1}^{m} A_j^\top A_j,$$

$$c := \frac{1}{m}\sum_{j=1}^{m} A_j^\top \tilde{y}_j,$$

$$r := \frac{1}{2m}\sum_{j=1}^{m} \|\tilde{y}_j\|^2 .$$

In the next sections, we will create new mathematical tools for the analysis of optimization problems. In particular, these tools will help us to study the solutions of the optimization problem (1.11).

1.3.2 Ridge regression

As we saw earlier, some regularization is often used in practice. There can be different reasons for that; one of them is called *overfitting*. The idea is that if there is a lot of degrees of freedom in the model, and yet not enough data points, the procedure might fits ”’too well”’ the data, and might not generalize well to new data points. An extreme case is when there are more parameters than data points, i.e., $n > m$.

To prevent this effect, one can add a regularization term to the optimization problem. This will force the model to stay "simple" while still fitting the data well.

When the regularization term is a penalty on the squared norm of the parameters, the optimization problem is called *ridge regression*. The optimization problem (1.11) becomes the following:

Example 1.3: Ridge regression

The ridge regression problem is the following optimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m}\|\tilde{y}_j - A_j x\|^2 + \frac{\lambda}{2}\;\|x\|^2 =: f(x), \tag{1.12}$$

where $\lambda > 0$ is a hyperparameter.

Later, we will see that the ridge regression problem has the nice property that there always exists a unique local minimizer, which is also the unique global minimizer.

Proposition 1.2: Ridge regression is a QP

The ridge regression problem (1.12) is a quadratic optimization problem in the form (1.8), with:

$$\begin{aligned} Q &:= \lambda I_n + \frac{1}{m}\sum_{j=1}^{m} A_j^\top A_j, \\ c &:= \frac{1}{m}\sum_{j=1}^{m} A_j^\top \tilde{y}_j, \\ r &:= \frac{1}{2m}\sum_{j=1}^{m}\|\tilde{y}_j\|^2 . \end{aligned} \tag{1.13}$$

where I_n is the identity matrix of size n.

Remark. When $\lambda > 0$ we have:

$$Q \;\succcurlyeq\; \lambda I_n \;\succ\; 0 \tag{1.14}$$

which implies that Q is non-singular.

1.3.3 LASSO regression

In the case one looks for a sparse solution to the linear least squares problem, one would need to penalize the number of non-zero entries in x. Since this would imply a non-continuous (hence non-convex) penalization, the resulting problem would be extremely difficult to solve. Instead, one can use the LASSO regression, which penalizes the l_1 norm of the parameters:

Example 1.4: LASSO regression

The following optimization problem is called the LASSO regression:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m} \|y_j - A_j x\|^2 + \lambda \|x\|_1 \,, \tag{1.15}$$

Remark. In (1.15), we used the l_1 norm of a vector, which is defined as the sum of the absolute values of its components:

$$\|x\|_1 \coloneqq \sum_{i=1}^{n} |x_i| \,. \tag{1.16}$$

Note that the optimization problem (1.15) is not a QP, but it still has a rather simple structure. With the tools from the next chapter (convexity), we will be able to characterize the solution-set of this problem.

1.3.4 Cross-entropy for classification tasks

Now that we have seen several examples of regression tasks, let us mention another common type of optimization problem in data analysis: classification tasks.
Here, the outputs y_i are discrete labels: integers in a set $\{1,\dots,q\}$. Typically, the labels represent different classes to which the input a_i can belong. In this case, it is usefull to define $p^{y_j} \in \{0,1\}^q$ as follows:

$$\text{for } j = 1,\dots,m \text{ and } l = 1,\dots,q, \quad (p^{y_j})_l = \begin{cases} 1 & \text{if } y_j = l, \\ 0 & \text{otherwise.} \end{cases} \tag{1.17}$$

Here, the vector $p^{y_j} \in \{0,1\}^q$ represents the membership of the label y_j to each of the q classes. We will learn a model $\varphi(a_j;x) \in [0,1]^q$ that approximates the vectors p^{y_j}. One could interpret $\varphi_l(a_j;x)$ as the probability that the label y_j belongs to class l. In the previous section, we have seen that the squared norm error is a common loss function for regression tasks. For classification problems, instead of the least squares loss, one often uses the cross-entropy loss between the probability distribution associated with p^{y_j} and the one associated with $\varphi(a_j;x)$:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m} L\left(p^{y_j}, \varphi(a_j;x)\right). \tag{1.18}$$

where the function $L(p,\bar{p})$ is the cross-entropy between two distributions, which is defined as follows:

$$L(p,\bar{p}) \coloneqq -\sum_{l=1}^{q} p_l \log(\bar{p}_l). \tag{1.19}$$

While the simplest structure for φ was a linear one in the regression case, in the present case, we need $\varphi(a_j; x) \in [0,1]^q$. Therefore, the standard choice uses the softmax function, as in the following example:

Example 1.5: Logistic regression

The logistic regression is the optimization problem (1.18) with the following model for $\varphi(a; x)$:

$$\varphi(a; x) := \operatorname{softmax}\big(A(a)x\big) \tag{1.20}$$

for some matrix $A(a) \in \mathbb{R}^{q\times n}$ of $\mathbb{R}^n$ that depend on the input a, and the function softmax is defined as follows:

$$\operatorname{softmax}(z)_l := \frac{e^{z_l}}{\sum_{k=1}^{q} e^{z_k}}. \tag{1.21}$$

1.4 Basic properties of optimization problems

In this section, we will discuss some properties of optimization problems of the form:

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad f(x) \tag{1.22}$$

1.4.1 Existence of solutions

Since we study the solutions of optimization problems, a natural question is whether such solutions exist.

First, let us keep in mind the following two examples where no solution exists.

Example 1.6: Unbounded optimization problem

The following problem does not have any solution (because the function is not bounded from bellow):

$$\underset{x\in\mathbb{R}}{\text{minimize}} \quad x \tag{1.23}$$

Example 1.7: Bounded optimization problem but with no solution

The following opitmization problem is buonded from bellow, but still, it does not have any solution:

$$\underset{x\in\mathbb{R}}{\text{minimize}} \quad e^{-x} \tag{1.24}$$

In the following theorems, we will see some conditions that are easy to check that guarantee the existence of a solution.

Theorem 1.1: Existence of a minimizer for a compact feasible set

If the feasible set $\mathcal{X} \subset \mathbb{R}^n$ is non-empty and compact (i.e., bounded and closed) and $f : \mathcal{X} \to \mathbb{R}$ is continuous, then there exists a global minimizer to the optimization problem (1.22).

Proof. Let us write $f^\star := \inf\limits_{x\in\mathcal{X}} f(x) \in \mathbb{R}^n \cup \{-\infty\}$. Almost by definition of the infimum, there exists a sequence $(x_k)_{k\in\mathbb{N}}$ in $\mathcal{X}$ such that:

$$f(x_k) \xrightarrow[k\to+\infty]{} f^\star \tag{1.25}$$

Since $\mathcal{X}$ is compact, the sequence x_k has at least one accumulation point $\bar{x} \in \mathcal{X}$, i.e. for some increasing sequence of integers $(k_j)_{j\in\mathbb{N}}$, we have $x_{k_j} \xrightarrow[j\to+\infty]{} \bar{x}$.
By continuity of f, we have:

$$f(x_{k_j}) \xrightarrow[k_j\to+\infty]{} f(\bar{x}) \tag{1.26}$$

By combining (1.25) and (1.26), we obtain $f(\bar{x}) = f^\star = \inf\limits_{x\in\mathcal{X}} f(x)$. This proves that $\bar{x}$ is a minimizer of f over $\mathcal{X}$.

□

Definition 1.6: Coercive functions

A function $f : \mathcal{X} \to \mathbb{R}$ is called *coercive* if there exists a function $\kappa : \mathbb{R} \to \mathbb{R}$ such that $\kappa(t) \xrightarrow[t\to+\infty]{} +\infty$ and such that $\forall x \in \mathcal{X}, f(x) \geq \kappa(\|x\|)$.

Theorem 1.2: Existence of a minimizer for a coercive function

Let f be a continuous and coercive function and $\mathcal{X}$ a closed and non-empty set. Then, there exists a global minimizer of the optimization problem (1.22).

Proof. Since $\mathcal{X}$ is non-empty, there exists some $x_0 \in \mathcal{X}$.
Using the fact that $\kappa(t) \xrightarrow[t\to+\infty]{} +\infty$, there exists some $c \in \mathbb{R}$ such that if $t > c$, then $\kappa(t) > f(x_0) + 1$. The set $\tilde{\mathcal{X}}_c \coloneqq \{x \in \mathcal{X} \mid \quad \|x\| \leq c\}$ is closed and bounded (hence it is compact). Using the previous theorem, there exists a vector $x^\star$ that minimizes f on $\tilde{\mathcal{X}}_c$. The following also holds:

$$\forall x \in \mathcal{X} \setminus \tilde{\mathcal{X}}_c \qquad f(x) > f(x_0) + 1 \geq f(x^\star) + 1 > f(x^\star)$$

Hence, $x^\star$ also minmizes f on $\mathcal{X} \setminus \tilde{\mathcal{X}}_c$. We can conclude that $x^\star$ is a minimizer of f over the whole set $\mathcal{X}$. □

1.4.2 First order optimality conditions for smooth functions

In this part, we assume the function $f : \mathcal{X} \to \mathbb{R}$ to be continuously differentiable.

Definition 1.7: Stationary points

Let $\bar{x}$ be a point in the interior of $\mathcal{X}$. We say that $\bar{x}$ is a *stationary point* of the optimization problem (1.22) if it satisfies:

$$\nabla f\left(\bar{x}\right) = 0 \tag{1.27}$$

Remark. A more general definition exists for points that are not in the interior of the feasible set. In this course, we do not focus on the cases concerning the points at the border of the feasible set.

Theorem 1.3: First-order condition

Let $\bar{x}$ be in the interior of the feasible set $\mathcal{X}$. If $\bar{x}$ is a local minimizer of the optimization problem (1.22), then it is a stationary point:

$$\nabla f\left(\bar{x}\right) = 0 \tag{1.28}$$

Proof. Let p be a vector of $\mathbb{R}^n$.
Since $\bar{x}$ is in the interior of $\mathcal{X}$, $\bar{x} + \alpha p \in \mathcal{X}$ for α small enough.
Furthermore, since $\bar{x}$ is a local minimizer, $f(\bar{x} + \alpha p) \geq f(\bar{x})$ for for α small enough.
This implies that for α small enough:

$$0 \leq \frac{f(\bar{x} + \alpha p) - f(\bar{x})}{\alpha} \xrightarrow[\alpha\to 0]{} \nabla f\left(\bar{x}\right)^\top p \tag{1.29}$$

Hence, we have $\nabla f\left(\bar{x}\right)^\top p \geq 0$ for all $p \in \mathbb{R}^n$.
By choosing $p = -\nabla f\left(\bar{x}\right)$, we obtain $-\left\|\nabla f\left(\bar{x}\right)\right\|^2 \geq 0$, which implies $\nabla f\left(\bar{x}\right) = 0$. □

1.4.3 Second order optimality conditions for smooth functions

In this part, we now assume the function $f : \mathcal{X} \to \mathbb{R}$ to be *twice* continuously differentiable.

Theorem 1.4: Second-order necessary conditions for unconstrained optimization

Let $\bar{x}$ be in the interior of the feasible set $\mathcal{X}$. If $\bar{x}$ is a local minimizer of the optimization problem (1.22), then not only it satisfies $\nabla f(\bar{x}) = 0$ but also:

$$\nabla^2 f(\bar{x}) \succcurlyeq 0 \tag{1.30}$$

Proof. Let $\bar{x}$ be a local minimizer.
First, $\bar{x}$ is a stationary point from the previous theorem. Let p be an element of $\mathbb{R}^n$. Using the same argument as in the proof above, $f(\bar{x} + \alpha p) \geq f(\bar{x})$ for α small enough. Let us now use the first-order Taylor expansion of f at $\bar{x}$:

$$\begin{aligned} f(\bar{x} + \alpha p) &= f(\bar{x}) + \nabla f(\bar{x})^\top (\alpha p) + \int_0^1 s(\alpha p)^\top \nabla^2 f(\bar{x} + s\alpha p)(\alpha p) \mathrm{d}s \\ &= f(\bar{x}) + \alpha^2 \int_0^1 s p^\top \nabla^2 f(\bar{x} + s\alpha p) p \mathrm{d}s \end{aligned} \tag{1.31}$$

After a rearrangement, we obtain:

$$0 \leq \frac{f(\bar{x} + \alpha p) - f(\bar{x})}{\alpha^2} = \int_0^1 s p^\top \nabla^2 f(\bar{x} + s\alpha p) p \mathrm{d}s \xrightarrow[\alpha \to 0]{} \int_0^1 s p^\top \nabla^2 f(\bar{x}) p \mathrm{d}s = \frac{1}{2} p^\top \nabla^2 f(\bar{x}) p \tag{1.32}$$

This proves $p^\top \nabla^2 f(\bar{x}) p \geq 0$. Since this is true for any vector $p \in \mathbb{R}^n$, we have $\nabla^2 f(\bar{x}) \succcurlyeq 0$. □

Theorem 1.5: Second-order sufficient conditions for unconstrained optimization

Let $\bar{x}$ be in the interior of the feasible set $\mathcal{X}$. If $\bar{x}$ is a stationary point of the optimization problem (1.22), and in addition, $\bar{x}$ satisfies the following condition:

$$\nabla^2 f(\bar{x}) \succ 0, \tag{1.33}$$

then x is a strict local minimizer of the optimization problem (1.22).

Proof. Let $\bar{x}$ be a stationary point that satisfies the condition (1.33). Using (1.33), there exists $\lambda > 0$ such that $\nabla^2 f(\bar{x}) \succcurlyeq \lambda I_n$.
The function $\nabla^2 f(x)$ is continuous. This implies that for any $\delta > 0$, there exists a neighborhood $\mathcal{N} \subset \mathcal{X}$ of $\bar{x}$ such that:

$$\forall x \in \mathcal{N}, \quad -\delta I_n \preccurlyeq \nabla^2 f(x) - \nabla^2 f(\bar{x}) \preccurlyeq \delta I_n \tag{1.34}$$

In particular, choosing $\delta < \lambda$, we have:

$$\forall x \in \mathcal{N}, \quad \nabla^2 f(x) \succ 0 \tag{1.35}$$

Let x' be an element of $\mathcal{N} \setminus \{\bar{x}\}$. Let us define $p := x - \bar{x} \neq 0$. Now let us use the first-order Taylor expansion of f at $\bar{x}$:

$$\begin{aligned} f(x') = f(\bar{x}+p) &= f(\bar{x}) + \nabla f(\bar{x})^\top p + \int_0^1 sp^\top \nabla^2 f(\bar{x}+sp)\, p \mathrm{d}s \\ &= f(\bar{x}) + \int_0^1 sp^\top \nabla^2 f(\bar{x}+sp)\, p \mathrm{d}s \end{aligned} \tag{1.36}$$

Furthermore, for all $s \in [0,1]$, $\bar{x}+sp \in \mathcal{N}$ (assuming that $\mathcal{N}$ is convex, which is the case if $\mathcal{N}$ is a ball for example). From (1.35), this implies $\nabla^2 f(\bar{x}+sp) \succ 0$, and therefore: $p^\top \nabla^2 f(\bar{x}+sp)\, p > 0$. Injecting this into (1.36), we obtain:

$$f(x') > f(\bar{x}) \tag{1.37}$$

This is true for any $x' \in \mathcal{N} \setminus \{\bar{x}\}$, where $\mathcal{N}$ is a neighborhood of $\bar{x}$ in $\mathcal{X}$. Hence, $\bar{x}$ is a strict local minimum of the optimization problem (1.22).

□

Example 1.8: Illustrative example

For $p = 1, 2, 3, 4$, consider the following optimization problem:

$$\underset{x \in \mathbb{R}}{\text{minimize}} \quad x^p \tag{1.38}$$

Let us detail the status of the point $x = 0$ for each value of p:

- If $p = 1$, the condition (1.28) is not satisfied, hence x can not be a (local) minimizer.
- If $p = 2$, the conditions (1.28) and (1.33) are satisfied, hence x is a local minimizer. In this specific case, it is also the unique global minimizer.
- If $p = 3$, the conditions (1.28) and (1.30) are satisfied, but not the conditions (1.33), hence, the second order conditions do not allow us to conclude. In this specific case, x is not a (local) minimizer.
- If $p = 4$, the conditions (1.28) and (1.30) are satisfied, but not the conditions (1.33), hence, the second order conditions do not allow us to conclude. In this specific case, x is the unique local, and even global minimizer.

1.5 Application of the properties to the regression examples

In this section, we will apply the properties that were proven in Section 1.4 to the regression problems that were introduced in Section 1.3.

1.5.1 Quadratic programs with $Q \succ 0$

In this part, we consider the loss of the quadratic program (1.8):

$$f(x) = \frac{1}{2} x^\top Q x - c^\top x + r. \tag{1.39}$$

In particular, we focus on the case where Q is positive definite: $Q \succ 0$.

Remark. As noted before, the loss of the Ridge regression problem (1.12) with $\lambda > 0$ falls into this category.

Remark. Some of the linear least square problems (1.11) also fall into this category, under the assumption that $\frac{1}{m}\sum_{j=1}^{m} A_j^\top A_j \succ 0$.

Proposition 1.3: Existence of a global minimizer

The function $f(x)$ is coercive (see Definition 1.6). Hence, using Theorem 1.2, it has at least one global minimizer in $\mathbb{R}^n$.

Proof. Define $\lambda_{\min}$ as the lowest eigenvalue of Q. Then we have: $Q - \lambda_{\min} I_n \succcurlyeq 0$ (where I_n is the identity matrix of size n). This implies that for all $x \in \mathbb{R}^n$: $x^\top Q x \geq \lambda_{\min} \|x\|^2$. This implies that:

$$\begin{aligned} f(x) &= \frac{1}{2} x^\top Q x - c^\top x + r \\ &\geq \frac{\lambda_{\min}}{2} \|x\|^2 - \|c\| \, \|x\| + r \\ &=: \kappa(\|x\|) \end{aligned}$$

where $\kappa(t) := \frac{\lambda_{\min}}{2} t^2 - \|c\| \, t + r \xrightarrow[t \to +\infty]{} +\infty$.

By definition, this implies that $f(x)$ is coercive. □

Proposition 1.4: Unique stationary point

The function $f(x)$ has a unique stationary point, given by the following formula:

$$x^\star = Q^{-1} c \tag{1.40}$$

Proof. Since the problem is unconstrained, i.e. $\mathcal{X} = \mathbb{R}^n$, the stationary points all verify the formula

$$\nabla f(x) = Qx - c = 0 \tag{1.41}$$

Since Q is positive definite, it is also invertible. Hence, the equation (1.41) has a unique solution, given by (1.40). □

Proposition 1.5: Unique minimizer

The stationary point $x^\star$ is the unique global minimizer of the function $f(x)$. It is also the unique local minimizer.

Proof. All global (resp. local) minimizers are stationary points (cf. Theorem 1.3). Using Proposition 1.4, there exists not more than one stationary point, given by $x^\star$. This implies that there exists no more than one global (resp.) minimizer.
On the other hand, using Proposition 1.3, there exists at least one global minimizer (which is also a local minimizer).
Combining these two facts, $x^\star$ is the unique global minimizer, and the unique local minimizer. □

1.5.2 Quadratic Programs with $Q \succcurlyeq 0$

In the case where Q is positive semi-definite, the function $f(x)$ is no longer coercive. It could even be that no global minimizer exists. For example, this is the case if $Q = 0$ and $c \neq 0$.

However, we still have some interesting results. Unfortunately, we cannot prove them using only the properties from Section 1.4. We still derive them for completeness.

Proposition 1.6: Stationary points are global minimizers for QP

Assume that there exists some stationary point $x^\star$ of $f(x) = \frac{1}{2}x^\top Qx - c^\top x + r$ (with $Q \succ 0$) over $\mathbb{R}^n$. Then $x^\star$ is a global minimizer of the function $f(x)$.

Proof. Since $\mathbb{R}^n$ is an open-set, the stationary point $x^\star$ verifies $\nabla f(x^\star) = Qx^\star - c = 0$. Let $x \in \mathbb{R}^n$.

We want to prove that $f(x) \geq f(x^\star)$. Let us define $z \coloneqq x - x^\star$. Then the following holds:

$$\begin{aligned}
f(x) &= f(z + x^\star) \\
&= \frac{1}{2}(x^\star + z)^\top Q(x^\star + z) - c^\top(x^\star + z) + r \\
&= \frac{1}{2}z^\top Qz + \frac{1}{2}\big(z^\top Qx^\star + x^\star Qz\big) - c^\top z + \big(\frac{1}{2}x^\star Qx^\star - c^\top x^\star + r\big) \\
&= \frac{1}{2}z^\top Qz + x^\star Qz - c^\top z + f(x^\star) \\
&= \frac{1}{2}z^\top Qz + \big(Qx^\star - c\big)^\top z + f(x^\star) \\
&= \frac{1}{2}z^\top Qz + f(x^\star) \\
&\geq f(x^\star)
\end{aligned}$$

□

Remark. In the next chapter, we will generalize the result from Proposition 1.6 to a broader class of optimization problems: *the convex optimization problems.*

Proposition 1.7: Lower-bounded quadratic programs

Assume that $f(x) = \frac{1}{2}x^\top Qx - c^\top x + r$ is lower-bounded over $\mathbb{R}^n$, i.e. $\min\limits_{x \in \mathbb{R}^n} f(x) > -\infty$. Then the function $f(x)$ has at least one stationary point over $\mathbb{R}^n$.

Proof. Let $z \in \mathrm{Ker}(Q)$, i.e. $Qz = 0$. Then we have $f(tz) = r - t(c^\top z)$. Since f is lower bounded over $\mathbb{R}^n$, the function $t \mapsto f(tz)$ is also lower bounded. This implies that $c^\top z = 0$.
Since this is true for any $z \in \mathrm{Ker}(Q)$, $\mathrm{Ker}(Q) \subset \mathrm{Ker}(c^\top)$.
Now, let us use some linear algebra results (which can be found in Appendix A.1):

$$c \in \mathrm{Im}(c) = \mathrm{Ker}(c^\top)^\perp \subset \mathrm{Ker}(Q)^\perp = \mathrm{Im}(Q^\top) = \mathrm{Im}(Q).$$

This implies that $c \in \mathrm{Im}(Q)$, i.e. there exists $x^\star \in \mathbb{R}^n$ such that $Qx^\star = c$. Hence, $\nabla f(x^\star) = Qx^\star - c = 0$, i.e. $x^\star$ is a stationary point. □

Corollary 1.1

Lower-bounded QPs have at least one global minimizer. Furthermore, these are all the points that satisfy the equation $Qx^\star = c$.

Proof. This is a direct application of the two previous theorems. □

Remark. The linear least squares problem (1.11) falls into that category.

1.5.3 LASSO regression: existence of minimizers

We recall that the LASSO regression problem is associated with the following loss function:

$$f(x) := \frac{1}{2m} \sum_{j=1}^{m} \|\tilde{y}_j - A_j x\|^2 + \lambda \|x\|_1 .$$

Proposition 1.8: Existence of minimizers for LASSO regression

Assume that $\lambda > 0$. Then, the loss function of the LASSO regression problem is coercive (see Definition 1.6). Hence, using Theorem 1.2, the optimization problem (1.15) has at least one solution

Proof. We have:

$$f(x) \geq \lambda \|x\|_1 \geq \underbrace{\lambda \|x\|_2}_{=:\kappa(\|x\|_2)} \xrightarrow[\|x\|_2 \to +\infty]{} +\infty$$

□

Remark. $\|x\|_1 \geq \|x\|_2$ because: $\|x\|_1^2 = \left(\sum_{i=1}^{n} |x_i|\right)^2 = \underbrace{\sum_{i=1}^{n} |x_i|^2}_{=\|x\|_2^2} + \underbrace{\sum_{i \neq j} |x_i| |x_j|}_{\geq 0} \geq \|x\|_2^2$

Chapter 2

Convexity

2.1 Convex sets

In this section, we define what convexity means for sets, and discuss the properties of convex sets.

Definition 2.1: Convex Set

A set $\mathcal{X} \subset \mathbb{R}^n$ is convex if

$$\forall x, y \in \mathcal{X}, \alpha \in [0, 1] : \ (1 - \alpha)x + \alpha y \in \mathcal{X}. \tag{2.1}$$

Remark. Intuitively, one could translate this definition into "all connecting lines lie inside the set."

Proposition 2.1: Intersection of convex sets

If S is a set of convex sets, then their intersection $\bigcap\limits_{\mathcal{X} \in S} \mathcal{X}$ is also convex.

Proof. Let $x, y \in \bigcap\limits_{\mathcal{X} \in S} \mathcal{X}$ and $\alpha \in [0, 1]$.
For all $\mathcal{X} \in S$, we have $x, y \in \mathcal{X}$. Since $\mathcal{X}$ is convex, $(1 - \alpha)x + \alpha y \in \mathcal{X}$.
This holds for any $\mathcal{X} \in S$, hence:

$$(1 - \alpha)x + \alpha y \in \bigcap_{\mathcal{X} \in S} \mathcal{X}. \tag{2.2}$$

This holds for all $x, y \in \bigcap\limits_{\mathcal{X} \in S} \mathcal{X}$ and $\alpha \in [0, 1]$. Therefore, $\bigcap\limits_{\mathcal{X} \in S} \mathcal{X}$ is convex. □

Remark. In particular, if $\mathcal{X}_1$ and $\mathcal{X}_2$ are convex sets, then their intersection $\mathcal{X}_1 \cap \mathcal{X}_2$ is also convex.

Remark. The union of convex sets is not necessarily convex.

Proposition 2.2: Cartesian product of convex sets

Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be convex sets. Then their Cartesian product $\mathcal{X}_1 \times \mathcal{X}_2 := \{(x_1, x_2) \quad \text{such that } x_1 \in \mathcal{X}_1 \quad \text{and } x_2 \in \mathcal{X}_2\}$ is also convex.

Proof. Let $x, y \in \mathcal{X}_1 \times \mathcal{X}_2$, and $\alpha \in [0,1]$. Then $x = (x_1, x_2)$ and $y = (y_1, y_2)$, where $x_1, y_1 \in \mathcal{X}_1$ and $x_2, y_2 \in \mathcal{X}_2$.
This implies that:

$$(1-\alpha)x + \alpha y = \Bigg(\underbrace{(1-\alpha)x_1 + \alpha y_1}_{\in \mathcal{X}_1}, \quad \underbrace{(1-\alpha)x_2 + \alpha y_2}_{\in \mathcal{X}_2} \Bigg) \in \mathcal{X}_1 \times \mathcal{X}_2. \tag{2.3}$$

□

Proposition 2.3: Affine transformation on convex sets

Let $\mathcal{X} \in \mathbb{R}^n$ be a convex set. Let $A \in \mathbb{R}^{m \times n}$ be a matrix, and $b \in \mathbb{R}^m$ be a vector. Then the set $A\mathcal{X} + b = \{Ax + b \quad \text{for } x \in \mathcal{X}\}$ is convex.

Proof. Left as an exercise. □

2.2 Convex functions

2.2.1 General case

Definition 2.2: Convex Function

A function $f : \mathcal{X} \to \mathbb{R}$ is convex, if $\mathcal{X}$ is convex and if

$$\forall x, y \in \mathcal{X}, \alpha \in [0,1] : \; f\left((1-\alpha)x + \alpha y\right) \leq (1-\alpha) f(x) + \alpha f(y) \tag{2.4}$$

Remark. In words, a function is convex when all secants are above the graph.

Proposition 2.4: Convex over affine is convex

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex. Then for any $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^m$, the function $g : \mathbb{R}^m \to \mathbb{R}$ defined by $g(x) = f(Ax + b)$ is also convex.

Proof. Left as an exercise. □

Proposition 2.5: Increasing affine function over convex is convex

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex. Let $a > 0$ and c be real numbers. Then, the function $g : \mathbb{R} \to \mathbb{R}$ defined by $g(x) = af(x) + c$ is also convex.

Proof. Left as an exercise. □

Proposition 2.6: Sum of convex functions is convex

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are convex. Then the function $h(x) = f(x)+g(x)$ is also convex.

Proof. Left as an exercise. □

Proposition 2.7: Characterization of convex functions with epigraph

A function $f : \mathcal{X} \to \mathbb{R}$ is convex if and only if its epigraph, i.e., the set $\{(x, s) \in \mathcal{X} \times \mathbb{R} | x \in \mathcal{X},\ s \geq f(x)\}$, is a convex set.

Proof. Left as an exercise. □

Definition 2.3: Sublevel sets

The set $\{x \in \mathbb{R}^n | f(x) \leq c\}$ is the "sublevel set" of f for the value c.

Proposition 2.8: Convexity of sublevel Sets

The sublevel sets of a convex function $f : \mathcal{X} \to \mathbb{R}$ are convex.

Proof. If $f(x) \leq c$ and $f(y) \leq c$ then for any $\alpha \in [0,1]$ it holds also

$$f((1-\alpha)x + \alpha y) \leq (1-\alpha)f(x) + \alpha f(y) \leq (1-\alpha)c + \alpha c = c.$$

□

2.2.2 Convexity of smooth functions

Proposition 2.9: Convexity for $\mathcal{C}^1$ Functions

Assume that $f : \mathcal{X} \to \mathbb{R}$ is continuously differentiable and $\mathcal{X}$ is convex. Then the following properties are equivalent

$$f \text{ is convex} \tag{2.5a}$$

$$\forall x, y \in \mathcal{X}, \quad f(x) + \nabla f(x)^\top (y-x) \leq f(y) \tag{2.5b}$$

$$\forall x, y \in \mathcal{X}, \quad \left(\nabla f(y) - \nabla f(x)\right)^\top (y-x) \geq 0 \tag{2.5c}$$

Remark. The property (2.5b) means that the graph of f is above its tangents.

Remark. The property (2.5c) is a multi-dimensional equivalent of saying "$\nabla f(x)$ is a non-decreasing function".

Proof. We recall that by defintion, (2.5a) is equivalent to:

$$\forall x, y \in \mathcal{X}, \alpha \in [0,1] : \quad f\left((1-\alpha)x + \alpha y\right) \leq (1-\alpha)f(x) + \alpha f(y) \tag{2.6}$$

In order to prove the equivalences (2.5a) $\iff$ (2.5b) $\iff$ (2.5c), we will show the following chain of implications:

$$\big((2.5a) \iff\big) \qquad (2.6) \implies (2.5b) \implies (2.5c) \implies (2.6) \qquad \big(\iff (2.5a)\big)$$

- (2.6) $\implies$ (2.5b):
  A rearranegment of (2.6) gives:

$$f(y) - f(x) \geq \frac{f\left(x + \alpha(y-x)\right) - f(x)}{\alpha} \xrightarrow[\alpha \to 0]{} \nabla f(x)^\top (y-x) \tag{2.7}$$

  which proves (2.5b).

- (2.5b) $\implies$ (2.5c):
  Let x, y be in $\mathcal{X}$. Consider both equation (2.5b), and the one where we swap x and y:

$$\begin{aligned} f(x) + \nabla f(x)^\top (y - x) \leq f(y) \\ f(y) + \nabla f(y)^\top (x - y) \leq f(x) \end{aligned}$$

  Let us add these two inequalities:

$$f(y) + f(x) + (\nabla f(x) - \nabla f(y))^\top (y - x) \leq f(y) + f(x)$$

  Now, subsract $f(x) + f(y)$ from both sides:

$$(\nabla f(x) - \nabla f(y))^\top (y - x) \leq 0$$

  which proves (2.5c) after multiplication by -1.

- (2.5c) $\implies$ (2.6):
  Let x, y be in $\mathcal{X}$ and $\alpha \in [0, 1]$. The property (2.4) being trivial for $\alpha = 0, 1$, we will assume $\alpha \in (0, 1)$. Let us define the function $g(t) \coloneqq f(x + t(y - x))$.
  We have $g(0) = f(x)$ and $g(1) = f(y)$.
  Furthermore, from (2.5c), we have that if $t_1 > t_2$, then $g'(t_1) \geq g'(t_2)$:

$$\begin{aligned} g'(t_1) - g'(t_2) &= \nabla f(x + t_1(y - x))^\top (y - x) - \nabla f(x + t_2(y - x))^\top (y - x) \\ &= \frac{1}{t_1 - t_2} (\nabla f(x_1) - \nabla f(x_2))^\top (x_1 - x_2) \\ &\geq 0 \end{aligned}$$

  with $x_1 = x + t_1(y - x)$ and $x_2 = x + t_2(y - x)$.

  Now let us prove (2.4):

$$\begin{aligned} g(\alpha) &= g(0) + \int_0^\alpha g'(t)\, \mathrm{d}t \\ &= g(0) + \alpha \int_0^1 g'(s\alpha)\, \mathrm{d}s \\ &= g(0) + \alpha \left(g(1) - g(0) - \int_0^1 g'(s)\, \mathrm{d}s \right) + \alpha \int_0^1 g'(s\alpha)\, \mathrm{d}s \\ &= (1 - \alpha) g(0) + \alpha g(1) - \alpha \int_0^1 \underbrace{\left(g'(\alpha) - g'(s\alpha)\right)}_{\geq 0} \mathrm{d}s \\ &\leq (1 - \alpha) g(0) + \alpha g(1) \end{aligned}$$

  Finally, remarking that $g(\alpha) = f((1 - \alpha)x + \alpha y)$, $g(0) = f(x)$ and $g(1) = f(y)$, we have proved (2.6). This concludes the proof that f is convex.

□

Theorem 2.1: Convexity for C^2 Functions

Assume that $f : \mathcal{X} \to \mathbb{R}$ is twice continuously differentiable and $\mathcal{X}$ convex. Then f is convex if and only if the following holds:

$$\forall x, y \in \mathcal{X} : \quad (x-y)^\top \nabla^2 f(x)(x-y) \geq 0. \tag{2.8}$$

Proof. Let us use the following formula:

$$\nabla f(y) - \nabla f(x) = \int_0^1 \nabla^2 f\left(x + s(y-x)\right)(y-x)\,\mathrm{d}s \tag{2.9}$$

If we combine (2.9) with the equivalence (2.5a) $\iff$ (2.5c) from Proposition 2.9, we obtain that f is convex if and only if:

$$\forall x, y \in \mathcal{X}, \quad \int_0^1 (y-x)^\top \nabla^2 f\left(x + s(y-x)\right)(y-x)\,\mathrm{d}s \geq 0 \tag{2.10}$$

Now let us show that (2.10) is equivalent to (2.8).

- (2.10) $\implies$ (2.8):
  Assume that (2.10) holds. Let $x, y \in \mathcal{X}$. Apply (2.10) to $x' = x$ and $y' = x + \alpha(y-x)$:

  $$\begin{aligned} 0 &\leq \int_0^1 \left(y' - x'\right)^\top \nabla^2 f\left(x' + s(y'-x')\right)\left(y'-x'\right)\,\mathrm{d}s \\ &= \int_0^1 \left(\alpha(y-x)\right)^\top \nabla^2 f\left(x + s\alpha(y-x)\right)\left(\alpha(y-x)\right)\,\mathrm{d}s \end{aligned}$$

  Then divide by α^2 and take the limit $\alpha \to 0$:

  $$0 \leq \int_0^1 \left(\alpha(y-x)\right)^\top \nabla^2 f\left(x + s\alpha(y-x)\right)\left(\alpha(y-x)\right)\,\mathrm{d}s \xrightarrow[\alpha\to 0]{} (y-x)^\top \nabla^2 f(x)(y-x)$$

  This proves (2.8).

- (2.8) $\implies$ (2.10):
  Assume that (2.8) holds. Let $x, y \in \mathcal{X}$. Apply (2.8) to $x' = x + s(y-x)$ and $y' = y + s(x-y)$:

  $$0 \leq (y'-x')^\top \nabla^2 f\left(x'\right)\left(y'-x'\right) = (1-s)^2 (y-x)^\top \nabla^2 f\left(x + s(y-x)\right)(y-x)$$

  By dividing by $(1-s)^2$ and integrating with respect to s from 0 to 1, we find implies (2.10).

□

Remark. For the points x in the interior of $\mathcal{X}$, equation (2.8) is equivalent to:

$$\nabla^2 f(x) \succcurlyeq 0 \tag{2.11}$$

Example 2.1: Quadratic Function

The function $f(x) = \frac{1}{2}x^\top Qx - c^\top x + r$ is convex if and only if $Q \succcurlyeq 0$, because $\forall x \in \mathbb{R}^n : \nabla^2 f(x) = Q$.

2.2.3 Strictly convex Functions

Definition 2.4: Strict Convexity

A function $f : \mathcal{X} \to \mathbb{R}$ is said to be *strictly convex* if:

$$\forall x, y \in \mathcal{X} \text{ such that } x \neq y, \forall \alpha \in (0,1) : \quad f\left((1-\alpha)x + \alpha y\right) < (1-\alpha)f(x) + \alpha f(y). \tag{2.12}$$

Proposition 2.10: Strict convexity for $\mathcal{C}^1$ Functions

Assume that $f : \mathcal{X} \to \mathbb{R}$ is continuously differentiable and $\mathcal{X}$ is convex. Then the following properties are equivalent

$$f \text{ is strictly convex} \tag{2.13a}$$

$$\forall x, y \in \mathcal{X}, \quad \text{such that } x \neq y, \quad f(x) + \nabla f(x)^\top (y - x) > f(y) \tag{2.13b}$$

$$\forall x, y \in \mathcal{X}, \quad \text{such that } x \neq y, \quad \left(\nabla f(y) - \nabla f(x)\right)^\top (y - x) > 0 \tag{2.13c}$$

Proof. Simply take the proof of Proposition 2.9 and replace the inequalities by strict inequalities. □

Theorem 2.2: Strict convexity of smooth functions

Let f be a twice continuously differentiable function on a convex set $\mathcal{X}$. Assume that the following holds:

$$\forall x \in \mathcal{X} : \quad \nabla^2 f(x) \succ 0. \tag{2.14}$$

Then f is strictly convex.

Proof. Assume that (2.14) holds. Let $x, y \in \mathcal{X}$, such that $x \neq y$. Using (2.9) again, we have:

$$\left(\nabla f(y) - \nabla f(x)\right)^\top (y - x) = \int_0^1 (y - x)\nabla^2 f\left(x + s(y - x)\right)(y - x)\,\mathrm{d}s > 0 \tag{2.15}$$

This allows us to conclude that f is strictly convex using the equivalence (2.13a) $\iff$ (2.13c) from Proposition 2.10. □

Remark. The converse is not necessarily true. For example, the function $f(x) = x^4$ is strictly convex, but $\nabla^2 f(x) = 13x^2$ is zero for $x = 0$.

Example 2.2: Strongly Convex Quadratic

The quadratic function $f(x) = \frac{1}{2}x^\top Qx - c^\top x + r$ is strictly convex if and only if $Q \succ 0$.

2.2.4 Strong convexity

Definition 2.5: Strongly convex function

Let $\mu > 0$ be a positive scalar. We say that f is μ-strongly convex when the function $f(x) - \frac{\mu}{2}\|x\|^2$ is convex.

Proposition 2.11: Characterization of μ-strongly convex functions

A function f being μ-strongly convex is equivalent to each of the following properties (when f is sufficiently differentiable):

$$\forall x, y \in \mathbb{R}^n, \alpha \in [0,1], \quad f\left((1-\alpha)x + \alpha y\right) \leq (1-\alpha)g(x) + \alpha g(y) - \frac{\mu}{2}\alpha(1-\alpha)\|y - x\|^2 \tag{2.16a}$$

$$\forall x, y \in \mathcal{X}, \quad f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2 \tag{2.16b}$$

$$\forall x, y \in \mathcal{X}, \quad \left(\nabla f(x) - \nabla f(y)\right)^\top (x - y) \geq \mu \|x - y\|^2 \tag{2.16c}$$

$$\forall x, y \in \mathcal{X}, \quad (x - y)^\top \nabla^2 f(x)(x - y) \geq \mu \|x - y\|^2 \tag{2.16d}$$

Proof. Using the characterizations (2.4), (2.5b), (2.5c), and (2.8) to the function $h(x) := f(x) - \frac{\mu}{2}\|x\|^2$, we find that f being μ-strongly convex is equivalent to each of the following properties (when f is sufficiently differentiable):

$$\forall x, y \in \mathbb{R}^n, \alpha \in [0,1], \quad h\left((1-\alpha)x + \alpha y\right) \leq (1-\alpha)h(x) + \alpha h(y) \tag{2.17a}$$

$$\forall x, y \in \mathcal{X}, \quad h(y) \geq h(x) + \nabla h(x)^\top (y - x) \tag{2.17b}$$

$$\forall x, y \in \mathcal{X}, \quad \left(\nabla h(x) - \nabla h(y)\right)^\top (x - y) \geq 0 \tag{2.17c}$$

$$\forall x, y \in \mathcal{X}, \quad \forall x, y \in \mathcal{X}, \quad (x - y)^\top \nabla^2 h(x)(x - y) \geq 0 \tag{2.17d}$$

Replacing $h(x)$ with $f(x) - \frac{\mu}{2}\|x\|^2$, we find that (2.17a), (2.17b), (2.17c) and (2.17d) are respectively equivalent to the following:

$$\forall x, y \in \mathbb{R}^n, \alpha \in [0,1], \tag{2.18a}$$
$$f((1-\alpha)x + \alpha y) - \frac{\mu}{2}\|(1-\alpha)x+\alpha y\|^2 \leq (1-\alpha)f(x) + \alpha f(y) - \frac{\mu}{2}\left((1-\alpha)\|x\|^2 + \alpha\|y\|^2\right)$$

$$\forall x, y \in \mathcal{X}, \quad f(y) - \frac{\mu}{2}\|y\|^2 \geq f(x) + \nabla f(x)^\top (y-x) - \frac{\mu}{2}\left(\|x\|^2 + \left(\nabla\|x\|^2\right)^\top (y-x)\right) \tag{2.18b}$$

$$\forall x, y \in \mathcal{X}, \quad (\nabla f(x) - \nabla f(y))^\top (x-y) - \frac{\mu}{2}\left(\nabla\|x\|^2 - \nabla\|y\|^2\right)^\top (x-y) \geq 0 \tag{2.18c}$$

$$\forall x, y \in \mathcal{X}, \quad \forall x, y \in \mathcal{X}, \quad (x-y)^\top \nabla^2 f(x)(x-y) - \frac{\mu}{2}(x-y)^\top \left(\nabla^2\|x\|^2\right)(x-y) \geq 0 \tag{2.18d}$$

After some rearrangements, these properties can be rewritten as:

$$\forall x, y \in \mathbb{R}^n, \alpha \in [0,1], \tag{2.19a}$$
$$f((1-\alpha)x + \alpha y) \leq (1-\alpha)g(x) + \alpha g(y) - \frac{\mu}{2}\Big(\underbrace{\|(1-\alpha)x+\alpha y\|^2 - (1-\alpha)\|x\|^2 - \alpha\|y\|^2}_{=\alpha(1-\alpha)\|y-x\|^2}\Big)$$

$$\forall x, y \in \mathcal{X}, \quad f(y) \geq f(x) + \nabla f(x)^\top (y-x) + \frac{\mu}{2}\Big(\underbrace{\|y\|^2 - \left(\|x\|^2 + \left(\nabla\|x\|^2\right)^\top (y-x)\right)}_{=\|x-y\|^2}\Big) \tag{2.19b}$$

$$\forall x, y \in \mathcal{X}, \quad (\nabla f(x) - \nabla f(y))^\top (x-y) \geq \frac{\mu}{2}\underbrace{\left(\nabla\|x\|^2 - \nabla\|y\|^2\right)^\top (x-y)}_{=2\|x-y\|^2} \tag{2.19c}$$

$$\forall x, y \in \mathcal{X}, \quad \forall x, y \in \mathcal{X}, \quad (x-y)^\top \nabla^2 f(x)(x-y) \geq \frac{\mu}{2}\underbrace{(x-y)^\top \left(\nabla^2\|x\|^2\right)(x-y)}_{=2\|x-y\|^2} \tag{2.19d}$$

The properties (2.19a), (2.19b), (2.19c), and (2.19d) are respectively equivalent to (2.16a), (2.16b), (2.16c), and (2.16d).
This concludes the proof that f being μ-strongly convex is equivalent to each of the properties (2.16a), (2.16b), (2.16c), and (2.16d). □

Remark. For the points x in the interior of $\mathcal{X}$, equation (2.16d) is equivalent to:

$$\nabla^2 f(x) \succcurlyeq \mu I_n \tag{2.20}$$

Proposition 2.12: Strong convexity implies strict convexity

If f is μ-strongly convex, then it is also strictly convex.

Proof. Direct consequence from the characterization (2.16a) from Proposition 2.11. □

Example 2.3: Strongly Convex Quadratic

The quadratic function $f(x) = \frac{1}{2}x^\top Qx - c^\top x + r$ is μ-strongly convex if and only if $Q \succcurlyeq \mu I_n$.

2.3 Convex optimization problems

An important class of optimization problems is the convex optimization problem.

"The great watershed in optimization is not between linearity and nonlinearity, but convexity and nonconvexity" *R. Tyrrell Rockafellar*

Definition 2.6: Convex Optimization Problem

If $\mathcal{X}$ is a convex set and $f : \mathcal{X} \to \mathbb{R}$ is a convex function, then the optimization problem (1.6) is called a "convex optimization problem".

Theorem 2.3: Local Implies Global Optimality for Convex Problems

For a convex optimization problem, every local minimum is also a global one.

Proof. Regard a local minimum $x^\star$ of the convex optimization problem (1.6). This means that there exists $\varepsilon > 0$ such that for all $x \in \mathcal{N}_\varepsilon := \{x| \quad \|x - x^\star\| \le \varepsilon\}$ we have $f(x) \ge f(x^\star)$.
Now let $y \in \mathcal{X} \setminus \{x^\star\}$. Let us define $\alpha = \frac{\varepsilon}{\|x^\star - y\|}$. Then $(1-\alpha)x^\star + \alpha y \in \mathcal{N}_\varepsilon$. Hence, using convexity of f, we have that

$$f(x^\star) \le f\left((1-\alpha)x^\star + \alpha y\right) \le (1-\alpha)f(x^\star) + \alpha f(y)$$

This implies that $f(y) \ge f(x^\star)$. Since this is true for any $y \in \mathcal{X}$, we can conclude that $x^\star$ is a global minimum. □

Theorem 2.4: Solution set

For a convex optimization problem, the set of minimizers is convex.

Proof. The set of minimizers is the set $\{x \in \mathcal{X} \mid f(x) = \min_x f(x)\} = \{x \in \mathcal{X} \mid f(x) \le \min_x f(x)\}$. As it is a sublevel set of a convex function, it is convex. □

Remark. The function f is constant on the set of global minimizers.

Theorem 2.5: First order optimality condition for convex problems

Let f be a convex and continuously differentiable function on a convex set $\mathcal{X}$. Let $\bar{x}$ be in the interior of $\mathcal{X}$. Then $\bar{x}$ is a global optimizer of (1.6) if and only if it is a stationary point.

Proof.

- $\Rightarrow$: Global optimality implies stationarity from Theorem 1.3.
- $\Leftarrow$: Let $\bar{x}$ be a stationary point, i.e. $\nabla f(\bar{x}) = 0$ Let us use the characterization (2.5b) of convexity of smooth functions, from Proposition 2.9 with $x = \bar{x}$:

$$\forall y \in \mathcal{X}, \quad f(y) \geq f(\bar{x}) + \nabla f(\bar{x})^\top (y - \bar{x}) = f(y)$$

  This implies that $\bar{x}$ is a global optimizer.

□

Theorem 2.6: Unicity of minimizer for strictly convex functions

Let f be a strictly convex function on a convex set $\mathcal{X}$. Then f has at most one minimizer in $\mathcal{X}$ (which is the unique stationary point in case of existence).

Proof. Let us prove this by contradiction. Assume that there exists two distinct minimizers x and y. Then apply the strict convexity property (2.12) with $\alpha = \frac{1}{2}$:

$$f\left(\frac{x+y}{2}\right) < \frac{f(x)+f(y)}{2}f(x) + \frac{1}{2}f(y) = \min_{x \in \mathcal{X}} f(x),$$

meaning that f evaluated at $\frac{x+y}{2}$ has a value lower than its minimum, and yet $\frac{x+y}{2}$ is in $\mathcal{X}$, since $\mathcal{X}$ is convex. This is a contradiction. □

Remark. In the case where $\mathcal{X}$ is open and f is also continuously differentiable, this result, combined with Theorem 2.5, implies that if the minimizer exists, it is also the unique stationary point. Furthermore, the minimizer exists if and only if a stationary point exists.

Lemma 2.1: Strong convexity implies coercive

Let f be a μ-strongly convex function. Then it is coercive.

Proof. Let us define the set $B = \{x \in \mathcal{X} \mid \|x\| \leq 1\}$. Since B is compact, $\bar{f} := \inf_{x \in B} f(x)$ is finite.

Let y be such that $\|y\| \geq 1$. Let us use the property (2.16a) for $x = 0$:

$$f(\alpha y) \leq (1-\alpha) f(0) + \alpha f(y) - \frac{\mu}{2}\alpha (1-\alpha) \|y\|^2 \tag{2.21}$$

Now, apply (2.21) to $\alpha = \frac{1}{\|y\|}$:

$$\bar{f} \leq f\left(\frac{y}{\|y\|}\right) \leq \left(1 - \frac{1}{\|y\|}\right) f(0) + \frac{f(y)}{\|y\|} - \frac{\mu}{2}\left(1 - \frac{1}{\|y\|}\right)\|y\| \tag{2.22}$$

After rearranging the terms, we get:

$$f(y) \geq \underbrace{\frac{\mu}{2}\|y\|^2 + \left(\bar{f} - \frac{\mu}{2}\right)\|y\| - f(0) + \frac{f(0)}{\|y\|}}_{=:\kappa(\|y\|)} \tag{2.23}$$

Clearly, $\kappa(\|y\|) \xrightarrow[\|y\| \to +\infty]{} +\infty$. This implies that f is coercive. □

Theorem 2.7: Existence and unicity theorem for strongly convex functions

Let f be a μ-strongly convex function on a convex set $\mathcal{X}$. Also, assume that $\mathcal{X}$ is closed and non-empty. Then f has a unique global minimizer in $\mathcal{X}$ (which is also the unique stationary point).

Proof. Using Lemma 2.1 and Proposition 2.12, f is coercive and strictly convex.
Using Theorem 1.2, f has at least one global minimizer.
Using Theorem 2.6, it has at most one minimizer.
This implies that f has a unique global minimizer.

□

Remark. In the case where $\mathcal{X}$ is open, and f is also continuously differentiable, this result combined with Theorem 2.5 implies that the unique global minimizer is also the unique stationary point.

2.4 Examples of convex optimization problems in data analysis

Example 2.4: Quadratic Programs

Consider the QP (1.8):

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}x^TQx - c^Tx + r,$$

Then the following holds:

- The problem is convex if and only if $Q \succcurlyeq 0$.
- The problem is strictly convex if and only if $Q \succ 0$.
- The problem is μ-strongly convex if and only $Q \succcurlyeq \mu I_n$

Example 2.5: Linear least squares problems

Consider the linear least squares optimization problem (1.11):

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m}\|y_j - A_jx\|^2$$

Then the following holds:

- The problem is convex.
- The problem is strictly convex if and only if $\frac{1}{m}\sum_{j=1}^{m} A_j^\top A_j \succ 0$.
- The problem is μ-strongly convex if and only if $\frac{1}{m}\sum_{j=1}^{m} A_j^\top A_j \succcurlyeq \mu I_n$.

Example 2.6: Ridge Regression

Consider the ridge regression optimization problem (1.12):

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m}\|y_j - A_jx\|^2 + \frac{\lambda}{2}\|x\|^2$$

The problem is always λ-strongly convex; hence, it has a unique solution, as we saw before.

Example 2.7: LASSO Regression

Consider the LASSO regression optimization problem (1.15):

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m} \|y_j - A_j x\|^2 + \lambda \|x\|_1$$

The problem is convex, and even strongly convex if the matrix $\frac{1}{m}\sum_{j=1}^{m} A_j^\top A_j$ is invertible.

Example 2.8: Robust regression problems

Another variant of linear least squares is the robust regression problem, where the goal is to be robust against potential outliers in the data set. More precisely, there might be a couple of data points that are highly corrupted by noise, and we do not want these data points to affect the solution too much. The robust regression problem can be formulated as follows:

$$\underset{x\in\mathbb{R}^n, O\in\mathbb{R}^{m\times p}}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m} \|y_j - (A_j x + o_j)\|^2 + 2\rho \|o_j\|_1 \tag{2.24}$$

where o_j are some additional errors, typically high only for a couple of samples, corresponding to the outliers. This is translated by the l_1 penalization on o_j in the objective function.

The optimization problem (2.24) is non-differentiable, but it can be transformed into another form, by explicitly optimizing over o_j:

Proposition 2.13: Equivalent form of the robust regression problem

The optimization problem (2.24) is equivalent to the following optimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m} h_\rho^{\text{vec}}(y_j - A_j x) \tag{2.25}$$

where $h_\rho^{\text{vec}}(e) := \sum_{k=1}^{p} h_\rho(e_k)$, and where h_ρ is the Huber function defined as:

$$h_\rho(e) := \begin{cases} \frac{1}{2}e^2 & \text{if } |e| \le \rho, \\ \rho|e| - \frac{1}{2}\rho^2 & \text{otherwise.} \end{cases} \tag{2.26}$$

Furthermore, the objective function of the optimization problem (2.25) is convex and continuously differentiable.

Proof. Let us define $e_{ij}(x) \coloneqq (y_j - A_j x)_i$. Then the following holds:

$$\begin{aligned} \frac{1}{2m}\sum_{j=1}^{m} \|y_j - (A_j x + o_j)\|^2 + 2\rho \|o_j\|_1 &= \frac{1}{2m}\sum_{j=1}^{m}\sum_{i=1}^{p}(e_{ij}(x) - o_{ij})^2 + 2\rho \sum_{i=1}^{p} |o_{ij}| && (2.27) \\ &= \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} f_{if}(o_{ij}; x), && (2.28) \end{aligned}$$

where we defined $f_{ij}(o; x) \coloneqq \frac{1}{2}(e_{ij}(x) - o)^2 + \rho |o|$.

By explicitly minimizing over o, we can rewrite the optimization problem (2.24) as follows:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & \left(\min_{O\in\mathbb{R}^{m\times p}} \frac{1}{2m}\sum_{j=1}^{m} \|y_j - (A_j x + o_j)\|^2 + 2\rho \|o_j\|_1 \right) && (2.29) \\ = & \left(\min_{O\in\mathbb{R}^{m\times p}} \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} f_{if}(o_{ij}; x) \right) && (2.30) \end{aligned}$$

Now, let us use the following identity:

$$\min_{O\in\mathbb{R}^{m\times p}} \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} f_{ij}(o_{ij}; x) = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} \min_{o\in\mathbb{R}} f_{ij}(o; x) \tag{2.31}$$

This implies that the optimization problem (2.32) is equivalent to the following optimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} \min_{o\in\mathbb{R}} f_{if}(o; x) \tag{2.32}$$

Now a simple analyse of the 1D function $f_{ij}(o; x)$ shows that the minimum is reached for:

$$o^{\star}_{ij} = \begin{cases} e_{ij}(x) - \rho & \text{if } e_{ij}(x) > \rho, \\ e_{ij}(x) + \rho & \text{if } e_{ij}(x) < -\rho, \\ 0 & \text{otherwise.} \end{cases} \tag{2.33}$$

This yields the following value for the minimum of $f_{ij}(o; x)$:

$$\min_{o\in\mathbb{R}} f_{ij}(o; x) = \begin{cases} \rho e_{ij}(x) - \frac{1}{2}\rho^2 & \text{if } |e_{ij}(x)| > \rho, \\ \frac{1}{2} e_{ij}(x)^2 & \text{otherwise.} \end{cases} \tag{2.34}$$

Note that this matches with the definition of the Huber function h_ρ in (2.26). Hence, we have shown that $\min_{o\in\mathbb{R}} f_{ij}(o; x) = h_\rho(e_{ij}(x))$. Plugging this equality into (2.32), we find that the optimization problem (2.24) is equivalent to the following:

$$
\begin{aligned}
&\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} h_\rho(e_{ij}(x))\\
\iff &\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{p} h_\rho\left((y_j - A_j x)_i\right)\\
\iff &\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m} h_\rho^{\text{vec}}\left(y_j - A_j x\right)
\end{aligned}
$$

as desired. □

Example 2.9: Nonlinear least squares problems

In the case of the nonlinear least squares problem (1.10):

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m} \|y_j - \varphi(a_j; x)\|^2\,,$$

the problem is, in general, non-convex.

Because of the non-convexity, the analysis of the solutions of (1.10) is quite difficult. However, there are some algorithms that can be used to find at least stationary points of the problem. The next chapter will be dedicated to optimization algorithms. To give an idea of how one could approach the problem, one very natural idea is to linearize the model around some initial guess $\bar{x}$, and solve the resulting linear least squares problem. Such a problem would take the following form:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m} \left\| y_j - \left(\varphi(a_j;\bar{x}) + \nabla\varphi(a_j;\bar{x})^\top (x-\bar{x})\right)\right\|^2 . \tag{2.35}$$

Iterating over the described procedure results in a method called the *Gauss-Newton method*, which is a popular method to solve nonlinear least squares problems. In this course, we will not study this method specifically but rather similar methods.

Example 2.10: Logistic regression

In the previous chapter, we introduced the logistic regression problem, in Example 1.5.
This optimization problem can be written more explicitly as follows:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m}\log\left(\sum_{l=1}^{q} e^{a_j^\top x_l}\right) - \frac{1}{m}\sum_{j=1}^{m}\sum_{l=1}^{q} p_l^{y_j}(a_j^\top x_l) \tag{2.36}$$

This optimization problem is convex as we will see in the next proposition.

To probe the convexity of the logistic loss, we first need to prove the following lemma.

Lemma 2.2: Convexity of log-sum-exp functions

The function $h(z) := \log\left(\sum\limits_{l=1}^{q} e^{z_l}\right)$ is convex.

Proof. First, let us explicitly write the derivatives of h:

$$(\nabla h(z))_i = \frac{e^{z_i}}{\sum\limits_{l=1}^{q} e^{z_l}}$$

Now, regarding the second derivatives:

$$\left(\nabla^2 h(z)\right)_{ij} = \frac{1}{\left(\sum\limits_{l=1}^{q} e^{z_l}\right)^2}\left(-e^{z_i}e^{z_j} + \begin{cases} e^{z_i}\left(\sum\limits_{l=1}^{q} e^{z_l}\right) & \text{if } i=j \\ 0 & \text{else}\end{cases}\right)$$

Let z and d be vectors of $\in \mathbb{R}^q$. The following holds:

$$\begin{aligned} d^\top \nabla^2 h(z) d &= \sum_{i=1}^{q}\sum_{j=1}^{q} d_i d_j \left(\nabla^2 h(z)\right)_{ij} \\ &= \frac{1}{\left(\sum\limits_{l=1}^{q} e^{z_l}\right)^2}\left(\left(\sum_{i=1}^{q} d_i^2 e^{z_i}\right)\left(\sum_{i=1}^{q} e^{z_l}\right) - \sum_{i=1}^{q}\sum_{j=1}^{q} d_i d_j e^{z_i} e^{z_j}\right) \\ &= \frac{1}{\left(\sum\limits_{l=1}^{q} e^{z_l}\right)^2}\left(\left(\sum_{i=1}^{q} d_i^2 e^{z_i}\right)\left(\sum_{i=1}^{q} e^{z_l}\right) - \left(\sum_{i=1}^{q} d_i e^{z_i}\right)^2\right) \end{aligned}$$

Furthermore, the following inequality holds, using the Cauchy-Schwarz inequality:

$$\begin{aligned}\left(\sum_{i=1}^{q} d_i e^{z_i}\right)^2 &= \left(\sum_{i=1}^{q} \left(d_i \sqrt{e^{z_i}}\right)\left(\sqrt{e^{z_i}}\right)\right)^2 \\ &\le \left(\sum_{i=1}^{q} \left(d_i \sqrt{e^{z_i}}\right)^2\right)\left(\sum_{i=1}^{q} \left(\sqrt{e^{z_i}}\right)^2\right) \\ &= \left(\sum_{i=1}^{q} d_i^2 e^{z_i}\right)\left(\sum_{i=1}^{q} e^{z_i}\right)\end{aligned}$$

This allows us to conclude that $d^\top \nabla^2 h(z) d \ge 0$. Since this holds for any point z and any direction d, we can conclude that h is convex. □

Proposition 2.14: Convexity of the logistic regression loss

The logistic regression optimization problem (2.36) is convex.

Proof. The objective function in (2.36) is the sum of functions that are either linear, either in the form of $h({a_j}^\top x)$.
Using Lemma 2.2, we know that h is convex. Since a convex-over-linear function is convex, the term $h({a_j}^\top x)$ is also convex.
Moreover, linear functions are convex.
Therefore, the objective function is a sum of convex functions. Hence, it is itself a convex function. □

Chapter 3

Descent Methods for Solving Optimization Problems

In this section, we will assume that $\mathcal{X} = \mathbb{R}^n$ and that f is a continuously differentiable function. This is usually referred to as *smooth and unconstrained optimization*. The optimization problem that we will treat in this section takes the following form:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \tag{3.1}$$

3.1 Generalities about Descent methods

If a closed-form of the solution of the optimization problem (3.1) is not available, one needs to approximate the solution with some algorithm. Such an algorithm typically takes the following iterative form:

```
Initialize $x_0$ to some initial guess
For $k = 0, \dots, T$ or until some convergence criterion is satisfied        (3.2)
    compute $x_{k+1}$ according to rule
```

Furthermore, the iterative rule for x_k will take the form:

$$x_{k+1} = x_k + \alpha_k d_k \tag{3.3}$$

where $\alpha_k \in (0, 1]$ is called the *step-length* and $d_k \in \mathbb{R}^n$ represents the direction in which we update the solution point x_k.

In most of the algorithms that are going to be seen in this chapter, the direction d_k is (only) a function of the current point x_k:

$$d_k = \phi(x_k) \tag{3.4}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is some function. In the next section, we will see that a very natural choice is $\phi(x) = -\nabla f(x)$.

Definition 3.1: Descent directions

A vector d is called a *descent direction* for f at the point x if:

$$\exists \varepsilon > 0 \text{ such that } \forall \alpha \in (0, \varepsilon],\ f(x + \alpha d) < f(x) \tag{3.5}$$

Proposition 3.1: Conditions for descent directions

Let f be an L-smooth function. Then, for d to be a descent direction at point x:

- it is necessary that $\nabla f(x)^\top d \leq 0$,
- it is sufficient that $\nabla f(x)^\top d < 0$

Remark. One could maybe relax the L-smooth assumption here, but it makes the proof easier.

Proof. Let us define the function $g(\alpha) := \frac{f(x+\alpha d) - f(x)}{\alpha}$.
Note that $\lim\limits_{\alpha \to 0} g(\alpha) = \nabla f(x)^\top d$.

Now we prove each of the points.

- $\nabla f(x)^\top d \leq 0$ *is necessary:*
  If d is a descent direction, then $g(\alpha) < 0$ for α small enough. This implies that $\nabla f(x)^\top d = \lim\limits_{\alpha \to 0} g(\alpha) \leq 0$
- $\nabla f(x)^\top d < 0$ *is sufficient:*
  If $\nabla f(x)^\top d < 0$, then $\lim\limits_{\alpha \to 0} g(\alpha) < 0$. This implies that $g(\alpha) < 0$ for α small enough, hence d is a descent direction.

□

Regularity of the function f

Definition 3.2: L-smooth function

Let $L > 0$ be a positive scalar. We say the the function f is L-smooth with its gradient is L-Lipschitz-continuous:

$$\forall x, y \in \mathbb{R}^n, \quad \|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \tag{3.6}$$

Proposition 3.2: Inequality for L-smooth functions

Assume that f is L-smooth. Then the following holds:

$$\forall x, y \in \mathbb{R}^n, \quad f(y) = f(x) + \nabla f(x)^\top (y - x) + r(x, y) \tag{3.7}$$

with $r(x, y)$ satisfying:

$$|r(x, y)| \leq \frac{L}{2} \|x - y\|^2 \tag{3.8}$$

Proof. Using the integral form of the first-order Taylor expansion, we find:

$$r(x, y) = \int_0^1 \Big(\nabla f\left(x + t(y - x)\right) - \nabla f(x) \Big)^\top (y - x)\, dt \tag{3.9}$$

Using the L-smoothness of f, we find:

$$\begin{aligned}
|r(x, y)| &= \left| \int_0^1 \Big(\nabla f\left(x + t(y - x)\right) - \nabla f(x) \Big)^\top (y - x)\, dt \right| \\
&\leq \int_0^1 \left| \Big(\nabla f\left(x + t(y - x)\right) - \nabla f(x) \Big)^\top (y - x) \right| dt \\
&\leq \int_0^1 \|\nabla f\left(x + t(y - x)\right) - \nabla f(x)\| \|x - y\| \, dt \qquad \text{using the Cauchy-Schwarz inequality} \\
&\leq \int_0^1 L \|x + t(y - x) - x\| \|x - y\| \, dt \qquad \text{using (3.6)} \\
&= L \int_0^1 t\, dt \, \|x - y\|^2 \\
&= \frac{L}{2} \|x - y\|^2
\end{aligned}$$

□

Proposition 3.3: Characterization of L-smooth function

Let f be a function twice-differentiable. Then f is L-smooth if and only if

$$\forall x \in \mathbb{R}^n, \quad -LI_n \preccurlyeq \nabla^2 f(x) \preccurlyeq LI_n \tag{3.10}$$

Proof. Now let us prove that f is L-smooth $\iff$ (3.10) holds:

$\Rightarrow$: Let $x \in \mathbb{R}^n$ and $d \in \mathbb{R}^n$. Using (3.11) and (3.6), we find:

$$\begin{aligned}
\left|d^\top \nabla^2 f(x) d\right| &= \left|\lim_{\alpha\to 0} \frac{1}{\alpha}\Big(\nabla f(x+\alpha d) - \nabla f(x)\Big)^\top d\right| \\
&= \lim_{\alpha\to 0} \frac{1}{\alpha}\left|\Big(\nabla f(x+\alpha d) - \nabla f(x)\Big)^\top d\right| \\
&\leq \lim_{\alpha\to 0} \frac{1}{\alpha}\left\|\Big(\nabla f(x+\alpha d) - \nabla f(x)\Big)^\top\right\| \|d\| \\
&\leq \lim_{\alpha\to 0} \frac{1}{\alpha} L \|\alpha d\| \|d\| \\
&\leq \lim_{\alpha\to 0} L \|d\|^2
\end{aligned}$$

which proves that $-L\|d\|^2 \leq d^\top \nabla^2 f(x) d \leq L\|d\|^2$. Since this holds for all $d \in \mathbb{R}^n$, we have that $-LI_n \preccurlyeq \nabla^2 f(x) \preccurlyeq LI_n$.

$\Leftarrow$: Using the fundamental theorem of calculus applied to $\nabla f(\cdot)$, we have that:

$$\nabla f(y) - \nabla f(x) = \int_0^1 \nabla^2 f(x + t(y-x))(y-x)\, dt \tag{3.11}$$

Since implies:

$$\begin{aligned}
\|\nabla f(y) - \nabla f(x)\| &= \left\|\int_0^1 \nabla^2 f(x+t(y-x))(y-x)\, dt\right\| \\
&= \int_0^1 \left\|\nabla^2 f(x+t(y-x))(y-x)\right\| \, dt \\
&= \int_0^1 \sqrt{(y-x)^\top \left(\nabla^2 f(x+t(y-x))\right)^2 (y-x)}\, dt \\
&\leq \int_0^1 \sqrt{L^2 \|x-y\|^2}\, dt \qquad\qquad = L\|x-y\|
\end{aligned}$$

which proves that f is L-smooth.

Note that we used the fact that (3.10) implies that $\left(\nabla^2 f(x)\right)^2 \preccurlyeq LI_n$.

□

A useful inequality for descent methods

Proposition 3.4

Let f be an L-smooth function. Let $x_0, \ldots, x_t, \ldots$ be updated according to the general rule (3.3). Then the following holds:

$$f(x_{k+1}) \leq f(x_k) + \alpha_k \nabla f(x_k)^\top d_k + \alpha_k^2 \frac{L}{2} \|d_k\|^2 \tag{3.12}$$

Proof. This comes directly from Proposition 3.2. □

Remark. From the inequality (3.12), we see that to minimize the right-hand side of (3.12), the choice $d_k = -\nabla f(x_k)$ and $\alpha_k = \frac{1}{L}$ is optimal. This choice results in a method called *the gradient descent method*. This will be studied more in-depth in the next section.

3.2 The gradient descent method

In this section, one very classical, widely used, algorithm: the *gradient descent algorithm*.

Definition 3.3: The gradient descent algorithm

The gradient descent algorithm is the following iterative method:

$$\begin{aligned}
&\text{Initialize } x_0 \text{ to some initial guess} \\
&\text{For } k = 0, \ldots, T \text{ or until some convergence criterion is satisfied} \\
&\qquad x_{k+1} = x_k - \alpha_k \nabla f(x_k)
\end{aligned} \tag{3.13}$$

Remark. As mentioned above, this corresponds to the choice $d_k = \phi(x_k) = -\nabla f(x_k)$

Remark. We have not yet defined how the step length is chosen. There are several ways to choose it, either via fixing a value $\alpha_k = \alpha$, or via a dedicated procedure, as we will see later.

Proposition 3.5

Assume that f is L-smooth. Let $\alpha_{\min}$ and $\alpha_{\max}$ be such that $0 < \alpha_{\min} \leq \alpha_{\max} < \frac{2}{L}$. Let $x_0, \ldots, x_t, \ldots$ be updated according to the gradient descent algorithm (3.13) with $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$. Then the following holds:

$$f(x_{k+1}) \leq f(x_k) - C \left\| \nabla f(x_k) \right\|^2 \tag{3.14}$$

with $C = \alpha_{\min}(1 - \alpha_{\max}\frac{L}{2}) > 0$.

Proof. Using equation (3.12) with $d_k = -\nabla f(x_k)$:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - \alpha_k \left\| \nabla f(x_k) \right\|^2 + \alpha_k^2 \frac{L}{2} \left\| \nabla f(x_k) \right\|^2 \\ &= f(x_k) - \alpha_k \left(1 - \alpha_k \frac{L}{2}\right) \left\| \nabla f(x_k) \right\|^2 \\ &\leq f(x_k) - \underbrace{\alpha_{\min} \left(1 - \alpha_{\max} \frac{L}{2}\right)}_{=C} \left\| \nabla f(x_k) \right\|^2 \end{aligned}$$

□

Remark. We see that to get the best bound on the decrease of the function, we should choose:

$$\alpha_{\min} = \alpha_{\max} = \frac{1}{L}. \tag{3.15}$$

This choice is called the *steepest descent method*, and it results in the following constant C:

$$C = \frac{1}{2L} \tag{3.16}$$

Remark that in practive, one might not know L and has to estimmate it from past iterates.

3.3 Convergence properties

3.3.1 The important assumption

In this section, we go back to the general case of descent methods. However, we make the assumption that a property similar to the property (3.14) is verified:

Assumption 3.1

Assume that there exists a constant C such that the following holds:

$$\forall k \in \mathbb{N} \quad f\left(x_{k+1}\right) \leq f\left(x_k\right) - C\left\|\nabla f\left(x_k\right)\right\|^2 \tag{3.17}$$

Remark. As we saw above, this property is verified for the gradient method under the assumption that the step lengths α_k are in a certain interval.

3.3.2 Convergence for a general smooth function

Proposition 3.6: Convergence of $\nabla f(x_k)$ to 0

Assume that f is L-smooth and bounded from below. Also, assume that $x_0, \ldots, x_t, \ldots$ is a sequence of points such that the Assumption 3.1 is fulfilled. Then we have

$$\nabla f\left(x_t\right) \xrightarrow[t \to +\infty]{} 0 \tag{3.18}$$

Proof. From the inequality (3.17), we have, for all k:

$$\left\|\nabla f\left(x_k\right)\right\|^2 \leq \frac{1}{C}\left(f\left(x_k\right) - f\left(x_{k+1}\right)\right) \tag{3.19}$$

Summing this inequality from $k = 0$ to $t-1$, we find:

$$\underbrace{\sum_{k=0}^{t-1}\left\|\nabla f\left(x_k\right)\right\|^2}_{=: u_t} \leq \frac{1}{C}\left(f\left(x_0\right) - f\left(x_t\right)\right) \leq \frac{1}{C}\left(f\left(x_0\right) - \inf_x f(x)\right) \tag{3.20}$$

The sequence u_t is non-decreasing and bounded; hence it converges. Therefore, the following holds:

$$\|\nabla f(x_t)\|^2 = u_{t+1} - u_t \xrightarrow[t \to +\infty]{} \lim_{t \to +\infty} u_{t+1} - \lim_{t \to +\infty} u_t = 0,$$

which proves (3.18). □

Remark. This proposition is important, but it does not necessarily mean that x_k will approach a stationary point of f. For example, if $f(x) = e^{-x}$, we still have f bounded from below, yet it does not admit any stationary point. In fact, such a point does not necessarily exist.

Theorem 3.1: Convergence theorem for a general smooth function

Assume that f is L-smooth (hence $\mathcal{C}^1$). Assume that the set $K := \{x \quad | \quad f(x) \leq f(x_0)\}$ is bounded. Also, assume that $x_0, \dots, x_t, \dots$ is a sequence of points such that the Assumption 3.1 is fulfilled. Then:

$$f\left(x_t\right) \xrightarrow[t \to +\infty]{} f(\bar{x}) \tag{3.21}$$

where $\bar{x}$ is a stationary point of f.

Remark. The set K is always bounded if f is coercive.

Proof. Because of Assumption 3.1, $f\left(x_k\right)$ is decreasing, which implies that $x_k \in K$ for all k. Using Bolzano-Weierstrass theorem, this implies that it has at least one accumulation point $\bar{x}$, i.e. there exists a sequence k_j that goes to $+\infty$ such that:

$$x_{k_j} \xrightarrow[j \to +\infty]{} \bar{x} \tag{3.22}$$

Moreover, the sequence $f_k := f\left(x_k\right)$ is non-increasing, and bouded (because $x_k \in K$) hence $f_k \xrightarrow[k \to +\infty]{} \bar{f}$ for some $\bar{f} \in \mathbb{R}$. Let us write $\bar{f}$ its limit. The following holds:

$$\bar{f} = \lim_{k \to +\infty} f\left(x_k\right) = \lim_{j \to +\infty} f\left(x_{k_j}\right) = f(\bar{x}) \tag{3.23}$$

where the last equality comes from the continuity of $f(\cdot)$.
This proves $f\left(x_k\right) \xrightarrow[k \to +\infty]{} f\left(\bar{x}\right)$.

Now, we only have to show that $\bar{x}$ is a stationary point of f, i.e. $\nabla f\left(\bar{x}\right) = 0$. Let us use (3.17) from Assumption 3.1:

$$\left\|\nabla f\left(x_k\right)\right\|^2 \leq \frac{1}{C}\left(f\left(x_k\right) - f\left(x_{k+1}\right)\right) \xrightarrow[k \to +\infty]{} \frac{1}{C}\left(\bar{f} - \bar{f}\right) = 0 \tag{3.24}$$

This implies that $\nabla f\left(x_k\right) \xrightarrow[k \to +\infty]{} 0$.

On the other hand, $\nabla f\left(\cdot\right)$ is continuous, which implies that $\nabla f\left(x_{k_j}\right) \xrightarrow[j \to +\infty]{} \nabla f\left(\bar{x}\right)$. Combining these two limits, we find $\nabla f\left(\bar{x}\right) = 0$. □

Remark. While Theorem 3.1 provides a good guarantee, it does not provide any information on the speed of convergence. In fact, the convergence might be very slow, as we see in the following example.

Example 3.1: A function where GD converges slowly

Let us define the function $f : \mathbb{R} \to \mathbb{R}$ as:

$$f(x) = \begin{cases} 1 - e^{-x} & \text{if } x \geq 0 \\ \frac{1}{2}(x+1)^2 - 1 & \text{if } x < 0 \end{cases} \tag{3.25}$$

This function is L-smooth with $L = 1$. It has a unique stationary point and minimum at $x = -1$. The steepest gradient method for f is:

$$x_{k+1} = x_k - \nabla f(x_k). \tag{3.26}$$

While the sequence x_k converges to -1, the convergence is very slow.
Indeed, if $x_0 > 0$, it will take more than $e^{x_0} - 1$ iterations to reach a negative point $x_t \leq 0$.

Proof. Let t be the first iteration such that $x_t < 0$. Then, for all $k \leq t-1$, we have:

$$x_{k+1} = x_k + d_k$$

with $d_k = -\nabla e^{-x_k}$.
Using the convexity of e^x, we have $e^{d_k} \geq 1 + d_k$. This implies:

$$e^{x_{k+1}} = e^{x_k} e^{d_k} \geq e^{x_k}(1 + d_k) = e^{x_k} - 1$$

By repeating the inequality found for $k = 0$ to $t-1$, we find:

$$e^{x_t} \geq e^{x_0} - t.$$

Finally, since $x_t \leq 0$, we have $e^{x_t} \leq 1$, which implies that $t \geq e^{x_0} - 1$. □

3.3.3 Convergence for a strongly convex function

Theorem 3.2: Convergence theorem for a strongly convex function

Assume that f is continuously differentiable (i.e. $f \in \mathcal{C}^1$) and μ-strongly convex. Let $x^\star$ be the solution of the optimization problem (3.1). Also, assume that $x_0, \ldots, x_t, \ldots$ is a sequence of points such that the Assumption 3.1 is fulfilled. Then the following holds for all $k \in \mathbb{N}$:

$$f(x_k) - f(x^\star) \leq (1 - 2\mu C)^k \left(f(x_0) - f(x^\star)\right) \tag{3.27}$$

Proof. Using (2.16b) from Proposition 2.11, we have:

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2} \|y - x\|^2$$

By minimizing both sides over y, we find:

$$f(x^\star) = \min_y f(y) \geq \min_y f(x) + \nabla f(x)^\top (y-x) + \frac{\mu}{2}\|y-x\|^2 = f(x) - \frac{1}{2\mu}\|\nabla f(x)\|^2$$

Now, apply this inequality for $x = x_k$:

$$\|\nabla f(x_k)\|^2 \geq 2\mu \left(f(x_k) - f(x^\star)\right)$$

Now, plugging this inequality into (3.17), we find:

$$\forall k \in \mathbb{N} \quad f(x_{k+1}) \leq f(x_k) - C\|\nabla f(x_k)\|^2 \leq f(x_k) - 2\mu C(f(x_k) - f(x^\star))$$

Hence, the sequence $u_k := f(x_k) - f(x^\star)$ verifies $u_{k+1} \leq (1-2\mu C)u_k$. Repeating this inequality iteratively leads to $u_k \leq (1-2\mu C)^k u_0$, which is the inequality (3.27). □

Remark. The speed of convergence here is very satisfying. For example, it ensures that the number of steps that are required is linear in the number of digits of precision that is desired.
For example, if a solution with $\left|f(x) - \min_x f(x)\right| \leq 10^{-M}$ is required, then one needs less that $k = \frac{M+\log_{10}(f(x_0)-f(x^\star))}{-\log_{10}(1-2\mu C)}$ number of steps.

Corollary 3.1: Convergence rate for the steepest descent method

For the steepest descent method $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$, the following holds:

$$f(x_k) - f(x^\star) \leq \left(1 - \frac{\mu}{L}\right)^k (f(x_0) - f(x^\star)) \tag{3.28}$$

Proof. From (3.16), we saw that $C = \frac{1}{2L}$ is achieved for the steepest descent method. Plugging this into (3.27), we find the desired result. □

Corollary 3.2: Bound on $x_k - x^\star$

If the assumptions of Theorem 3.2 are fulfilled, then the following also holds:

$$\|x_k - x^\star\|^2 \leq \frac{2}{\mu}(1-2\mu C)^k (f(x_0) - f(x^\star)) \tag{3.29}$$

Proof. This directly follows from (3.27) and from:

$$f(x_k) \geq f(x^\star) + \frac{\mu}{2}\|x_k - x^\star\|^2, \tag{3.30}$$

which is derived from the strong convex inequality (2.16b) for $y = x_k$ and $x = x^\star$. □

Corollary 3.3

If in addition to the assumption in Theorem 3.2, f is L-smooth, then the following holds

$$\|x_k - x^\star\|^2 \leq \frac{L}{\mu}(1-2\mu C)^k \|x_0 - x^\star\|^2 \tag{3.31}$$

Proof. This directly follows from (3.29) and from:

$$f(x_0) \leq f(x^\star) + \frac{L}{2}\|x_0 - x^\star\|^2 \tag{3.32}$$

which is derived from the L-smooth inequality (3.7) for $y = x_0$ and $x = x^\star$. □

3.3.4 Convergence for a weakly convex function

Regarding the convergence of gradient descent methods for convex functions, Theorem 3.2 is very encouraging. Indeed, it provides a strong convergence guarantee and ensures a fast convergence when the function is strongly convex.

Furthermore, a function that is convex is not so far from being strongly convex. For example, if f is convex, then $f + \frac{\epsilon}{2}\|x\|^2$ is strongly convex. That means that by perturbing the function f a little, the gradient method would converge rather quickly to the solution.

The following theorem provides convergence guarantee for weakly convex functions under reasonable assumptions. The convergence speed is however slower than for strongly convex functions.

Before we state the theorem, let us prove the following lemma:

Lemma 3.1: A small mathematical point

Let $u_0, \ldots, u_t$ be a sequence of non-negative real numbers such that for some $\lambda > 0$:

$$\forall k \in \mathbb{N}: \quad u_{k+1} \leq u_k - \lambda u_k^2 \tag{3.33}$$

Then $u_k \leq \frac{1}{\lambda k}$ for all $k \in \mathbb{N}$.

Proof. First, note that $u_{k+1} \leq u_k$ for all $k \in \mathbb{N}$. This permits us to rearrange (3.33) into:

$$\frac{1}{u_{k+1}} - \frac{1}{u_k} = \frac{u_k - u_{k+1}}{u_k u_{k+1}} \geq \frac{u_k - u_{k+1}}{u_k^2} \geq \lambda \tag{3.34}$$

where the last inequality comes from (3.33) directly.

Now, summing the inequality (3.34) from $k = 0$ to $t - 1$, we find:

$$\frac{1}{u_t} - \frac{1}{u_0} \geq \lambda t \tag{3.35}$$

which implies $u_t \leq \frac{1}{\lambda t}$. □

Theorem 3.3: Convergence theorem for a convex function

Assume that f is L-smooth and convex. Let $x^\star$ be a solution of the optimization problem (3.1). Also, assume that $x_0, \ldots, x_t, \ldots$ is a sequence of points such that the Assumption 3.1 is fulfilled. Moreover, like in Theorem 3.1, assume that the set $K := \{x \mid f(x) \leq f(x_0)\}$ is bounded. Then define:

$$R_0 := \max \left\{ \|x - x^\star\| \quad \middle| \quad x \in K \right\} = \max \left\{ \|x - x^\star\| \quad \middle| \quad f(x) \leq f(x_0) \right\} < \infty \tag{3.36}$$

Then the following holds for all k:

$$f(x_k) - f(x^\star) \leq \frac{R_0^2}{C} \frac{1}{k} \tag{3.37}$$

Proof. First, $f(x_k)$ is decreasing because of Assumption 3.1. This implies $x_k \in K$ for all k, which implies:

$$\forall k \in \mathbb{N}, \quad \|x_k - x^\star\| \leq R_0. \tag{3.38}$$

Furthermore, since f is convex, we have:

$$\begin{aligned} f(x_k) &\leq f(x^\star) + \nabla f(x^\star)^\top (x_k - x^\star) && \text{from (2.5b)} \\ &\leq f(x^\star) + \|\nabla f(x^\star)\| \, \|x_k - x^\star\| && \text{from the Cauchy-Schwarz inequality} \\ &\leq f(x^\star) + \|\nabla f(x^\star)\| \, R_0 && \text{from (3.38)} \end{aligned}$$

which can be rearranged as:

$$\|\nabla f(x^\star)\| \geq \frac{f(x_k) - f(x^\star)}{R_0} \tag{3.39}$$

Now, using equation (3.17) combined with (3.39), we have:

$$f(x_{k+1}) \leq f(x_k) - \frac{C}{R_0^2} (f(x_k) - f(x^\star))^2 \tag{3.40}$$

Now, define $\lambda := \frac{C}{R_0^2}$ and $u_k := f(x_{k+1}) - f(x^\star)$. Note that $u_k \leq 0$. Substracting $f(x^\star)$ in both sides of equation (3.40):

$$u_{k+1} \leq u_k - \lambda u_k^2 \tag{3.41}$$

Using Lemma 3.1, this implies that $u_k \leq \frac{1}{\lambda k}$ for all $k \in \mathbb{N}$, which proves (3.37). □

3.4 Globalization techniques

As we saw for the gradient descent method, to ensure convergence, we need to ensure some constraints on the step-lengths α_k. The critical part of the constraint that we saw was $\alpha_{\max} < \frac{2}{L}$. However, the constant L depends on global properties of the function f, while we can only access local properties of f.

We make the distinction between three types of method to choose the step-length α_k:

- *Fixed step-length*: $\alpha_k = \alpha \approx \frac{1}{L}$ for all k. This is the simplest method, but it requires a good guess on the value of L.
- *Exact line search*: α_k can be chosen according to the optimization problem:

  $$\alpha_k = \arg\min_{\alpha > 0} f\left(x_k + \alpha d_k\right)$$

  This method require to solve an additional 1D optimization problem at each iteration, so it is in general computationally expensive.
- *Backtracking line search*: one finds a value on the form $\alpha_k = \beta^i \bar{\alpha}$ for some $\bar{\alpha}, \beta \in (0,1)$ by iteratively decreasing i until the Armijo criterion (cf. definition below) is satisfied. This is summarized in the following algorithm:

  ```
  $\alpha_k \leftarrow \bar{\alpha}$
  while the Armijo Criterion is not satisfied for $\alpha_k$ :        (3.42)
      $\alpha_k \leftarrow \beta\alpha_k$
  ```

Definition 3.4: Armijo criterion

The Armijo criterion is a test on α regarding the veracity of the following inequality:

$$f\left(x_k + \alpha d_k\right) \leq f\left(x_k\right) + c_A \alpha \nabla f\left(x_k\right)^\top d_k \tag{3.43}$$

for some $c_A \in (0,1)$.

In the following theorem, we will prove a strong convergence result for the backtracking line search method. Before stating that theorem, however, we need to introduce the following assumption:

Assumption 3.2

Assume that the direction is chosen as $d_k = \phi(x_k)$, and assume that $\phi(x)$ is such that, for some $\varepsilon > 0$ and somme $\gamma > 0$, the two following conditions hold:

$$\forall x \in \mathbb{R}^n, \quad -\nabla f(x)^\top \phi(x) \geq \varepsilon \left\|\nabla f(x)\right\|^2 \tag{3.44a}$$

$$\forall x \in \mathbb{R}^n, \quad \left\|\phi(x)\right\| \leq \gamma \left\|\nabla f(x)\right\| \tag{3.44b}$$

Remark. The Assumption 3.2 is verified for the gradient descent method with $\gamma = \varepsilon = 1$.

Proposition 3.7: Termination of backtracking line-search

Assume that f is an L-smooth function. Under Assumption 3.2, the backtracking line-search method (3.42) with Armijo criterion (3.43) terminates in a finite number of iterations.

Proof. Using the L-smooth property of f and equation (3.7), we have for all α:

$$f(x_k + \alpha d_k) \leq f(x_k) + \alpha \nabla f(x_k)^\top d_k + \alpha^2 \frac{L}{2} \left\|d_k\right\|^2 \tag{3.45}$$

This implies:

$$\begin{aligned}
f(x_k + \alpha d_k) &\leq f(x_k) + \alpha \nabla f(x_k)^\top d_k + \alpha^2 \frac{L}{2} \left\|d_k\right\|^2 \\
&= f(x_k) + c_A \alpha \nabla f(x_k)^\top d_k + (1 - c_A)\alpha \nabla f(x_k)^\top d_k + \frac{L}{2}\alpha^2 \left\|d_k\right\|^2 \\
&\leq f(x_k) + c_A \alpha \nabla f(x_k)^\top d_k - (1 - c_A)\alpha\varepsilon \left\|\nabla f(x_k)\right\|^2 + \frac{L}{2}\alpha^2 \left\|d_k\right\|^2 \\
&\leq f(x_k) + c_A \alpha \nabla f(x_k)^\top d_k + \alpha \frac{L}{2} \left\|d_k\right\|^2 \left(\alpha - \frac{2(1 - c_A)\varepsilon \left\|\nabla f(x_k)\right\|^2}{L \left\|d_k\right\|^2}\right)
\end{aligned}$$

On the otherhand, since $\beta^i \bar{\alpha} \xrightarrow[i \to +\infty]{} 0$, there exists an iteration i' such that $\alpha' := \beta^{i'} \bar{\alpha} \leq \frac{2(1-c_A)\varepsilon\|\nabla f(x_k)\|^2}{L\|d_k\|^2}$. This implies, using the inequality derived above:

$$f\left(x_k + \alpha' d_k\right) \leq f(x_k) + c_A \alpha' \nabla f(x_k)^\top d_k$$

This proves that the Armijo criterion at iteration i', therefore, the backtracking line-search method terminates in a finite number of iterations (maximum i'). □

Theorem 3.4: Convergence result for backtracking line search

Let f be an L-smooth function. Assume that the chosen direction d_k follows Assumption 3.2. Furthermore, consider that the backtracking line-search method is used with the Armijo criterion (3.43).
Then, Assumption 3.1 holds for $C = \max\left(c_A\varepsilon\bar{\alpha},\ c_A(1-c_A)\frac{2\beta\varepsilon^2}{L\gamma^2}\right)$, i.e.:

$$f\left(x_{k+1}\right) \leq f\left(x_k\right) - C\left\|\nabla f\left(x_k\right)\right\|^2 \tag{3.46}$$

Proof. We have to distinguish between two cases: the case where backtracking is uncessary, and the case where at least one backtracking was performed.

- For the iterations where no backtracking is needed, the Armijo criterion is satisfied for $\alpha_k = \bar{\alpha}$. Using (3.43) from the Armijo condition and (3.44a) from Assumption 3.1:

$$\begin{aligned} f\left(x_{k+1}\right) &\leq f\left(x_k\right) + c_A\bar{\alpha}\nabla f\left(x_k\right)^\top d_k, \\ &\leq f\left(x_k\right) - c_A\bar{\alpha}\varepsilon\left\|\nabla f\left(x_k\right)\right\|^2, \\ &\leq f\left(x_k\right) - C\left\|\nabla f\left(x_k\right)\right\|^2 \end{aligned}$$

- For the iterations where backtracking is needed, the condition is satisfied for $\alpha = \alpha_k$, but not for $\alpha = \beta^{-1}\alpha_k$. This is summarized as follows:

$$f\left(x_k + \beta^{-1}\alpha_k d_k\right) > f\left(x_k\right) + c_A\beta^{-1}\alpha_k\nabla f\left(x_k\right)^\top d_k \tag{3.47a}$$

$$f\left(x_{k+1}\right) = f\left(x_k + \alpha_k d_k\right) \leq f\left(x_k\right) + c_A\alpha_k\nabla f\left(x_k\right)^\top d_k \tag{3.47b}$$

On the other hand, the inequality (3.7) for L-smooths functions gives:

$$f\left(x_k + \beta^{-1}\alpha_k d_k\right) \leq f\left(x_k\right) + \beta^{-1}\alpha_k\nabla f\left(x_k\right)^\top d_k + \frac{L}{2}\beta^{-2}\alpha_k^2\left\|d_k\right\|^2 \tag{3.48}$$

Combining (3.47a) and (3.48), we have:

$$f\left(x_k\right) + c_A\beta^{-1}\alpha_k\nabla f\left(x_k\right)^\top d_k \leq f\left(x_k\right) + \beta^{-1}\alpha_k\nabla f\left(x_k\right)^\top d_k + \frac{L}{2}\beta^{-2}\alpha_k^2\left\|d_k\right\|^2, \tag{3.49}$$

which implies:

$$\alpha \geq -(1-c_A)\frac{2\beta}{L}\frac{\nabla f\left(x_k\right)^\top d_k}{\left\|d_k\right\|^2} \tag{3.50}$$

Now, let us plug (3.50) into (3.47b):

$$\begin{aligned}
f\left(x_{k+1}\right) \leq f\left(x_k\right) - c_A &\left((1-c_A)\frac{2\beta}{L}\frac{\nabla f\left(x_k\right)^\top d_k}{\|d_k\|^2}\right)\nabla f\left(x_k\right)^\top d_k \\
&= f\left(x_k\right) - c_A(1-c_A)\frac{2\beta}{L}\left(\frac{\nabla f\left(x_k\right)^\top d_k}{\|d_k\|}\right)^2, \\
&\leq f\left(x_k\right) - c_A(1-c_A)\frac{2\beta\varepsilon^2}{L}\left(\frac{\|\nabla f\left(x_k\right)\|^2}{\|d_k\|}\right)^2 \quad \text{using (3.44a)}, \\
&\leq f\left(x_k\right) - c_A(1-c_A)\frac{2\beta\varepsilon^2}{L\gamma^2}\left(\frac{\|\nabla f\left(x_k\right)\|^2}{\|\nabla f\left(x_k\right)\|}\right)^2 \quad \text{using (3.44b)}, \\
&= f\left(x_k\right) - c_A(1-c_A)\frac{2\beta\varepsilon^2}{L\gamma^2}\|\nabla f\left(x_k\right)\|^2, \\
&\leq f\left(x_k\right) - C\|\nabla f\left(x_k\right)\|^2,
\end{aligned}$$

□

Corollary 3.4

Thanks to Theorem 3.4, the convergence theorems 3.1, 3.3, 3.2 and Corollary 3.3 also apply when using the backtracking line search method.

3.5 Other examples of descent methods

3.5.1 Quasi-Newton methods

Withouth going to much into the details of these methods, a popular class of methods is called second-order methods. There, the idea is to minimize a convex quadratic approximation of the function f at each iteration. Such quadratic approximation takes the following form:

$$f(x) \approx f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top H(x_k)(x - x_k) \tag{3.51}$$

for some matrix $H(x_k) \succ 0$.
When minimizing the right hand-side of (3.51), we obtain $x = x_k + d_k$ where d_k is defined as follows:

$$d_k = -H(x_k)^{-1}\nabla f(x_k) =: \phi(x_k) \tag{3.52}$$

When $H(x_k) = \nabla^2 f(x_k)$, the method is called the *exact Newton method.* Note that this method is well defined only when $H(x_k) \succ 0$.

If we define $M(x) := H(x)^{-1}$,we get the following class of methods.

Definition 3.5: Quasi-Newton method

The *quasi-Newton methods* are descent methods where the direction d_k is computed as follows:

$$d_k = -M(x_k)\nabla f(x_k) \quad (=: \phi(x_k)) \tag{3.53}$$

for some matrices $M(x_k) \succcurlyeq 0$.

Proposition 3.8

Assume that $M(x)$ is such that:

$$\forall x \in \mathbb{R}^n, \quad \varepsilon I_n \preccurlyeq M(x) \preccurlyeq \gamma I_n, \tag{3.54}$$

with some $\varepsilon > 0$ and $\gamma > 0$.
Then, the direction d_k defined by (3.53) fulfills Assumption 3.1 (with the same ε and γ).

Proof. Writting Assumption 3.1 for $d_k = -M(x_k)\nabla f(x_k)$ reads:

$$\forall x \in \mathbb{R}^n \quad g^\top M(x) g \geq \varepsilon \|g\|^2 \quad \text{with } g = \nabla f(x) \tag{3.55a}$$

$$\forall x \in \mathbb{R}^n \quad \|M(x)g\| \leq \gamma \|g\| \quad \text{with } g = \nabla f(x) \tag{3.55b}$$

Equation (3.55a) is directly derived from $M(x) \succcurlyeq \varepsilon I_n$ Regarding equation (3.55b), we can rearrange it at follows:

$$\forall x \in \mathbb{R}^n g^\top M(x)^2 g \leq \gamma^2 \|g\|^2 \quad \text{with } g = \nabla f(x), \tag{3.56}$$

which is verified by remarking that $0 \preccurlyeq M(x) \preccurlyeq \gamma I_n \implies M(x)^2 \preccurlyeq \gamma^2 I_n$. □

Corollary 3.5

The quasi-Newton method combined with a backtracking line search for the globalization strategy inherits from the convergence results from theorems 3.1, 3.3, 3.2 and the corollary 3.3.

Proof. Direct consequence of Proposition 3.8. □

The Gauss-Newton method For solving the non-linear least squares problem (1.10), we discussed about the Gauss-Newton method in Section 2.4, which consists in solving iteratively the optimization problem:

$$\underset{d\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m}\left\|y_j - \left(\varphi(a_j;x_k) + \nabla\varphi(a_j;x_k)^\top d\right)\right\|^2. \tag{3.57}$$

Note that this method is an example of quasi-Newton methods, with the choice:

$$M(x) = \left(\frac{1}{m}\sum_{j=1}^{m}\nabla\varphi(a_j;x)\nabla\varphi(a_j;x)^\top\right)^{-1}$$

3.5.2 Stochastic gradient methods

In Chapter 5, we will discuss in detail the stochastic gradient descent methods. Here, we can at least mention the setup: instead of having access to the true gradient $\nabla f(x_k)$, we only have access to a random variable g_k which is such that $\mathbb{E}[g_k] = \nabla f(x_k)$. In this case, the direction $d_k = -g_k$ does not exactly verify the descent conditions (3.44a) and (3.44b), but it will be verified in average.

3.5.3 Coordinate descent methods

In coordinate descent methods, at each iteration, only one coordinate of the solution x_k is updated. More precisely, let e_i be the vector with a 1 at the i-th position and 0 elsewhere.
The coordinate descent methods take the form:

$$x_{k+1} = x_k - \alpha_k\left(\nabla f(x)\right)_{i_k} e_{i_k} \tag{3.58}$$

for some index i_k and some step-size α_k.
A typical choice is to choose the index i_k randomly at each iteration, or to cycle through the indices.
For these choices, the descent conditions discussed in the previous sections are not exactly verified, but similar results can be obtained.
There exists, however, one variation where the Assumption 3.1 *is* verified: the Gauss-Southwell method. In this specific method, we choose the index i_k as follows:

$$i_k = \arg\max_i \left|\left(\nabla f(x_k)\right)_i\right| \tag{3.59}$$

Regarding the motivation behind this choice, one could be reluctant: why would we use only one coordinate of the gradient if we have to access to all of them anyway? The answer is that, in some specific cases, the gradient can be updated very efficiently when only one coordinate is updated. This is the case, for example, when the objective function is a sum of many functions, each depending only on a limited number of variables.

Chapter 4

Descent Methods with Momentum

In the previous methods that we saw, at each step, we only keep track of the current estimate of the solution. However, it could be that by using more information from the previous steps, the convergence is improved. In descent methods, the algorithm typically takes the following form:

```
Initialize $x_0, y_0$ to some initial guess
For $k = 0, \dots, T$ or until some convergence criterion is satisfied
    compute the gradient $g_k = \nabla f(x_k)$                                  (4.1)
    compute $d_k$ according to some rule $d_k = \Phi_k\big(g_1, \dots, g_k, x_1, \dots, x_k\big)$
    update $x_k$ $x_{k+1} = x_k + \alpha_k d_k$
```

In the sketch of algorithm (4.1), the rule is very general, and will see some more specificity later on.

4.1 Derivation of descent methods with momentum

4.1.1 Motivation from differential equations

To guide the choice of the update rule, one analogy is meaningful: seeing the gradient descent method as an approximation of the following differential equation:

$$\dot{x}(t) = -\nabla f\big(x\big) \tag{4.2}$$

The differential equation (4.2) is called the gradient flow method and has very strong properties, except that it is not numerically implementable. Let us note that if one would approximate (4.2) with an Euler scheme:

$$x(t + \Delta t) = x(t) - \Delta t \nabla f\big(x(t)\big), \tag{4.3}$$

one would recover the gradient descent method, with $\alpha = \Delta t$ and $x_k = x(k \cdot \Delta t)$.

Now, if one replaces the equation (4.2) by a physics-inspired equation, where $f(x)$ represents the energy associated with the position x, one would get the following second-order differential equation:

$$\ddot{x}(t) = -\nabla f\big(x\big) - \nu \dot{x}(t) \tag{4.4}$$

where ν is a friction coefficient. In (4.4), the vector $d = -\nabla f\big(x\big)$ does not directly influence the velocity of $x(t)$ anymore, but rather the acceleration.
One of the motivations behind (4.4) is that "small local minim" might be skipped thanks to the inertia of the particle $x(t)$. On the other hand, the friction term $-\nu\dot{x}(t)$ ensures that in the long term, the particle $x(t)$ will stabilize at a local minimum of its energy $f(x)$.

4.1.2 Derivation of the heavy-ball method

The heavy-ball method is derived from a discretization of the differential equation (4.4). Using finite differences on (4.4), we get the following discretization:

$$\frac{x(t+\Delta t) - 2x(t) + x(t-\Delta t)}{(\Delta t)^2} = -\nabla f\big(x(t)\big) - \nu \frac{x(t) - x(t-\Delta t)}{\Delta t} \tag{4.5}$$

Like for the gradient flow, we define $x_k = x(k \cdot \Delta t)$, and rearrange equation (4.5):

$$x_{k+1} - x_k = -(\Delta t)^2 \nabla f\big(x(t)\big) + (1 - \nu \Delta t)\left(x_k - x_{k-1}\right) \tag{4.6}$$

Finally, after defining $\alpha = (\Delta t)^2$ and $\beta = 1 - \nu\Delta t$, we can see that (4.6) is equivalent to the heavy-ball method defined as follows.

Definition 4.1: The heavy-ball method

The heavy-ball method is the following iterative algorithm:

$$x_{k+1} = x_k + \beta\left(x_k - x_{k-1}\right) - \alpha \nabla f\big(x_k\big) \tag{4.7}$$

where α and β are two parameters.

Proposition 4.1: Alternative formulation

The heavy-ball method (4.7) can also be formulated as follows:

$$\begin{aligned} x_{k+1} &= x_k + \alpha p_k, \\ d_{k+1} &= -\nabla f\big(x_{k+1}\big), \\ p_{k+1} &= \gamma p_k + d_{k+1} \end{aligned} \tag{4.8}$$

Proof. Define $p_k := \frac{x_{k+1} - x_k}{\alpha}$, and $\gamma = \beta$ permits us to rewrite the heavy-ball method (4.7) as in (4.8). □

Remark. By setting $r = 1 - \gamma$, $\tilde{\alpha} = \frac{\alpha}{r}$ and $\tilde{p}_k = rp_k$, we can rewrite (4.8) as follows:

$$\begin{aligned} x_{k+1} &= x_k + \tilde{\alpha}\tilde{p}_k, \\ d_{k+1} &= -\nabla f\big(x_{k+1}\big), \\ \tilde{p}_{k+1} &= (1-r)\tilde{p}_k + rd_{k+1} \end{aligned} \tag{4.9}$$

From this formulation, we can see that in the heavy-ball method, the descent direction $\tilde{p}_k$ is a weighted average of the previous direction and the current negative gradient. By rearranging the equations (4.9), we can see that that the descent direction is a weighted sum of all the previous negative gradients, with a forgetting factor r:

$$x_{k+1} = x_k + \tilde{\alpha}\frac{\sum\limits_{i=0}^{k}(1-r)^i d_{k-i}}{\sum\limits_{i=0}^{k}(1-r)^i} \tag{4.10}$$

From this perspective, this method has another advantage: when one has only access to a noisy estimate of the gradient, averaging the gradient over time reduces the noise for the descent direction.

4.1.3 Nesterov's accelerated gradient method

Now that we have seen the heavy-ball method, we will see a small variation of it.

Definition 4.2: Nesterov's accelerated gradient method

Nesterov's accelerated gradient method is the following iterative method:

$$x_{k+1} = x_k + \beta(x_k - x_{k-1}) - \alpha\nabla f\big(x_k + \beta(x_k - x_{k-1})\big) \tag{4.11}$$

where α and β are two parameters.

Remark. The present method is very similar to the heavy-ball method, except that the gradient is computed after adding the inertia term $\beta(x_k - x_{k-1})$ instead of before.

Remark. For a fine choice of the parameters α and β, the Nesterov's accelerated gradient method actually converges faster than the heavy-ball method. This is the reason why this method is also called *The Nesterov's Optimal Method.* In the next section, we will dive into the convergence rate of this method.

Note that one might prefer to use the following notation for Nesterov's method:

Definition 4.3: Nesterov's accelerated gradient method (alternative formulation)

Nesterov's accelerated gradient method can also be written as follows

$$\begin{aligned} y_k &= x_k + \beta(x_k - x_{k-1}), \\ x_{k+1} &= y_k - \alpha \nabla f\big(y_k\big), \end{aligned} \tag{4.12}$$

4.2 Convergence analysis of Nesterov's accelerated gradient method

In this section, we will analyze the convergence rate of Nesterov's method, for μ-strongly convex and L-smooth functions. We recall that, in the case of twice continuously differentiable functions, these are the functions that verifies:

$$\forall x \in \mathbb{R}^n, \quad \mu I_n \preccurlyeq \nabla^2 f(x) \preccurlyeq L I_n \tag{4.13}$$

We will also define the following quantity:

$$c := \sqrt{\frac{\mu}{L}} \in (0,1) \tag{4.14}$$

Remark. In the case where f is quadratic, $c = \sqrt{\operatorname{cond}\left(Q^{-1}\right)}$ where $\operatorname{cond}(P)$ is the condition number of a matrix P: the ration between its largest and smallest eigenvalues.

We also recall that for such functions, we found in Chapter 3 that the steepest gradient descent method (i.e. the gradient descent method with $\alpha = \frac{1}{L}$) converged exponentially fast, with the following rate (cf. (3.28) in Corollary 3.1):

$$f(x_k) - f(x^\star) \leq M(1 - c^2)^k \tag{4.15}$$

for some constant $M > 0$.

In the case of Nesterov's method, a better convergence rate can be obtained, given that both μ and L are known a priori, as we will see. The method is optimal for the following choice of parameters:

$$\alpha = \frac{1}{L}, \qquad \beta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} = \frac{1 - c}{1 + c} \tag{4.16}$$

Since the proof is quite long, we will divide it into many Lemma.

4.2.1 Proof of the convergence rate of Nesterov's optimal method

Lemma 4.1: Some convexity inequalities

The following inequalities hold:

$$f(x_{k+1}) \leq f(y_k) - \frac{L}{2}\left\|\alpha \nabla f(y_k)\right\|^2 \tag{4.17a}$$

$$f(y_k) \leq f(x_k) + \beta \nabla f(y_k)^\top (x_k - x_{k-1}) \tag{4.17b}$$

$$f(y_k) \leq f(x^\star) + \nabla f(y_k)^\top (y_k - x^\star) - \frac{\mu}{2}\left\|y_k - x^\star\right\|^2 \tag{4.17c}$$

Proof.

- Since f is L-smooth, we can use the inequality (3.7) from Proposition 3.2:

$$\begin{aligned} f(x_{k+1}) = f\big(y_k - \alpha \nabla f(y_k)\big) &\leq f(y_k) - \alpha \nabla f(y_k)^\top \nabla f(y_k) + \frac{L}{2}\left\|\alpha \nabla f(y_k)\right\|^2 \\ &= f(y_k) + \left(-\frac{1}{\alpha} + \frac{L}{2}\right)\left\|\alpha \nabla f(y_k)\right\|^2 \\ &= f(y_k) - \frac{L}{2}\left\|\alpha \nabla f(y_k)\right\|^2 \end{aligned}$$

- Then, since f is convex, we can use the inequality (2.5b) from Proposition 2.9:

$$f(y_k) + \nabla f(y_k)^\top \underbrace{(x_k - y_k)}_{=\beta(x_k - x_{k-1})} \leq f(x_k)$$

  which directly implies (4.17b).

- Finally, since f is μ-strongly convex, we can use the inequality (2.16b) from Proposition 2.11:

$$f(x^\star) \geq f(y_k) + \nabla f(y_k)^\top (x^\star - y_k) + \frac{\mu}{2}\left\|x^\star - y_k\right\|^2$$

  which implies (4.17c) after a small rearrangement.

□

Lemma 4.2: Bound on the function decrease

The following holds:

$$f(x_{k+1}) - f(x^\star) \leq (1-c)\left(f(x_k) - f(x^\star)\right) + r_k, \tag{4.18}$$

with r_k defined as:

$$r_k \coloneqq \frac{L}{2}\left[\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - c^3\|y_k - x^\star\|^2 - \|x_{k+1} - x_k + c(x_k - x^\star)\|^2\right] \tag{4.19}$$

Proof. Using the inequalities from Lemma 4.1, we can derive the following chain of inequalities:

$$\begin{aligned}
&f(x_{k+1}) - f(x^\star)\\
&\quad\leq f(y_k) - f(x^\star) - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2 && \text{from (4.17a)},\\
&\quad= (1-c)\left(f(y_k) - f(x^\star)\right) + c\left(f(y_k) - f(x^\star)\right) - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2,\\
&\quad\leq (1-c)\Big(f(x_k) + \beta\nabla f(y_k)^\top(x_k - x_{k-1}) - f(x^\star)\Big) + c\left(f(y_k) - f(x^\star)\right) - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2 && \text{from (4.17b)},\\
&\quad\leq (1-c)\left(f(x_k) - f(x^\star)\right) + \underbrace{\beta(1-c)\nabla f(y_k)^\top(x_k - x_{k-1}) + c\left(f(y_k) - f(x^\star)\right) - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2}_{=:\tilde{r}_k}
\end{aligned}$$

The quantity $\tilde{r}_k$ can be bounded using inequality (4.17c) and then simplified:

$$\begin{aligned}
\tilde{r}_k &\coloneqq \beta(1-c)\nabla f(y_k)^\top(x_k - x_{k-1}) + c\left(f(y_k) - f(x^\star)\right) - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2,\\
&\leq \beta(1-c)\nabla f(y_k)^\top(x_k - x_{k-1}) + c\left(\nabla f(y_k)^\top(y_k - x^\star) - \frac{\mu}{2}\|y_k - x^\star\|^2\right) - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2,\\
&= \nabla f(y_k)^\top\Big(\beta(1-c)(x_k - x_{k-1}) + c(y_k - x^\star)\Big) - c\frac{\mu}{2}\|y_k - x^\star\|^2 - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2,\\
&= \nabla f(y_k)^\top\Big(\beta(x_k - x_{k-1}) + c(x_k - x^\star)\Big) - c\frac{\mu}{2}\|y_k - x^\star\|^2 - \frac{L}{2}\|\alpha\nabla f(y_k)\|^2,\\
&= \frac{L}{2}\left[2\alpha\nabla f(y_k)^\top\Big(\beta(x_k - x_{k-1}) + c(x_k - x^\star)\Big) - \|\alpha\nabla f(y_k)\|^2\right] - c\frac{\mu}{2}\|y_k - x^\star\|^2,\\
&= \frac{L}{2}\left[\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - \|\beta(x_k - x_{k-1}) + c(x_k - x^\star) - \alpha\nabla f(y_k)\|^2\right] - c\frac{\mu}{2}\|y_k - x^\star\|^2,\\
&= \frac{L}{2}\left[\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - \|x_{k+1} - x_k + c(x_k - x^\star)\|^2\right] - c\frac{\mu}{2}\|y_k - x^\star\|^2,\\
&= \frac{L}{2}\left[\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - \|x_{k+1} - x_k + c(x_k - x^\star)\|^2 - c^3\|y_k - x^\star\|^2\right],\\
&= r_k.
\end{aligned}$$

which concludes the proof. □

Lemma 4.3: An inequality about norms

If $c \in [0,1]$ and $\beta = \frac{1-c}{1+c}$, then: The following inequality holds:

$$\forall u, v \in \mathbb{R}^n \quad \|\beta v + cu\|^2 - c^3 \|\beta v + u\|^2 \leq (1-c) \|(1-c)v + cu\|^2 \tag{4.20}$$

Proof. First, by expanding the right side, one can proves the following equality:

$$\|a\|^2 + (1-c)(1-c+c^2) \|b\|^2 = c \|a - (1-c)b\|^2 + (1-c) \|a + cb\|^2$$

Remarking that $1 - c + c^2 \geq 0$, we have $(1-c)(1-c+c^2) \|b\|^2 \geq 0$ for $c \in [0,1]$. This implies:

$$\|a\|^2 \leq c \|a - (1-c)b\|^2 + (1-c) \|a + cb\|^2$$

Then, applying it to $a = \beta v + cu$ and $b = \beta v$, we obtain:

$$\begin{aligned}\|\beta v + cu\|^2 &\leq c \|\beta v + cu - (1-c)\beta v\|^2 + (1-c) \|\beta v + cu + c\beta v\|^2 \\ &= c \|c\beta v + cu\|^2 + (1-c) \|(1+c)\beta v + cu\|^2 \\ &= c^3 \|\beta v + u\|^2 + (1-c) \|(1+c)\beta v + cu\|^2\end{aligned}$$

Finally, use the fact that $\beta = \frac{1-c}{1+c}$ to simplify $(1+c)\beta$ into $(1-c)$ and get the desired result. □

Lemma 4.4: A storage function

The quantity r_k defined in Lemma 4.2, equation (4.19), verifies:

$$r_k \leq (1-c) l(x_k, x_{k-1}) - l(x_{k+1}, x_k) \tag{4.21}$$

where the function $l(x_k, x_{k-1})$ is defined as follows:

$$l(x_k, x_{k-1}) := \frac{L}{2} \|(1-c)(x_k - x_{k-1}) + c\,(x_k - x^\star)\|^2 \tag{4.22}$$

Proof. Recall the definition of r_k:

$$r_k := \frac{L}{2}\Big[\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - c^3 \|\beta(x_k - x_{k-1}) + (x_k - x^\star)\|^2 \\ - \|x_{k+1} - x_k + c(x_k - x^\star)\|^2 \Big]$$

First note that we can make the following arrangement:

$$l(x_{k+1}, x_k) = \frac{L}{2} \|(1-c)(x_{k+1} - x_k) + c(x_{k+1} - x^\star)\|^2 = \frac{L}{2} \|x_{k+1} - x_k + c(x_k - x^\star)\|^2$$

This implies:

$$r_k \coloneqq \frac{L}{2}\left[\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - c^3 \|\beta(x_k - x_{k-1}) + (x_k - x^\star)\|^2 \right] - l(x_{k+1}, x_k) \tag{4.23}$$

Secondly, using the fact that $c \in [0,1]$ and that $\beta = \frac{1-c}{1+c}$, we apply the inequality (4.20) from Lemma 4.3 to $u = x_k - x^\star$ and $v = x_k - x_{k-1}$:

$$\begin{aligned} &\|\beta(x_k - x_{k-1}) + c(x_k - x^\star)\|^2 - c^3 \|\beta(x_k - x_{k-1}) + (x_k - x^\star)\|^2 \\ &\leq (1-c)\left[\|(1-c)(x_k - x_{k-1}) + c\,(x_k - x^\star)\|^2 \right] \end{aligned}$$

Combining this with (4.23), we can conclude:

$$\begin{aligned} r_k &\leq \frac{L}{2}(1-c)\left[\|(1-c)(x_k - x_{k-1}) + c\,(x_k - x^\star)\|^2 \right] - l(x_{k+1}, x_k) \\ &= (1-c)l(x_k, x_{k-1}) - l(x_{k+1}, x_k) \end{aligned}$$

which concludes the proof. □

Lemma 4.5: A Lyapunov function

We define the function $V(x_k, x_{k-1})$ as folllows:

$$V(x_k, x_{k-1}) \coloneqq f(x_k) - f(x^\star) + l(x_k, x_{k-1}) \tag{4.24}$$

Then, the following holds:

$$V(x_{k+1}, x_k) \leq (1-c)V(x_k, x_{k-1}) \tag{4.25}$$

Remark. It is classical in a lot of proof to find a function $V(\cdot)$ such that an equation like (4.25) is verified. This is called a Lyapunov function.

Remark. For a full expression of the function $V(x_k, x_{k-1})$:

$$V(x_k, x_{k-1}) \coloneqq f(x_k) - f(x^\star) + \frac{L}{2}\|(1+c)\beta(x_k - x_{k-1}) + c\,(x_k - x^\star)\|^2 \tag{4.26}$$

Proof. Combining 4.2 and 4.4, we find:

$$\begin{aligned} & f(x_{k+1}) - f(x^\star) \leq (1-c)\left(f(x_k) - f(x^\star)\right) + (1-c)l(x_k, x_{k-1}) - l(x_{k+1}, x_k) \\ \implies & \underbrace{f(x_{k+1}) - f(x^\star) + l(x_k, x_{k-1})}_{=V(x_{k+1}, x_k)} \leq (1-c)\big(\underbrace{f(x_k) - f(x^\star) + l(x_k, x_{k-1})}_{=V(x_k, x_{k-1})}\big) \end{aligned}$$

which concludes the proof. □

Theorem 4.1: Convergence of Nesterov's method for strongly convex functions

Assume that f is μ-strongly convex and L-smooth. Consider Nesterov's method (4.11) with the choice of parameters (4.16):

$$\alpha = \frac{1}{L}, \qquad \beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}} = \frac{1-c}{1+c}$$

Let $x^\star$ be the solution of the optimization problem (3.1). Then, the sequence x_k converges to $x^\star$, with the following rate:

$$f(x_k) - f(x^\star) \leq (1-c)^k\, M(x_0) \tag{4.27}$$

with $M(x_0) := f(x_0) - f(x^\star) + \frac{\mu}{2}\left\|x_0 - x^\star\right\|^2$

Remark. Since $1 - c \leq 1 - c^2$, we can see that the Nesterov's method converges faster than the steepest gradient descent method.

Proof. This is a direct consequence of Lemma 4.5. Indeed, applying the inequality (4.25) recursively implies:

$$f(x_k) - f(x^\star) \leq V(x_k, x_{k-1}) \leq (1-c)^k V(x_0, x_{-1})$$

Finally, a little rearrangement can be made to get to the final result (4.27):

$$\begin{aligned} V(x_0, x_{-1}) &= f(x_0) - f(x^\star) + \frac{L}{2}\left\|z_0\right\|^2 \\ &= f(x_0) - f(x^\star) + \frac{Lc^2}{2}\left\|e_0\right\|^2 \\ &= f(x_0) - f(x^\star) + \frac{\mu}{2}\left\|x_0 - x^\star\right\|^2 \end{aligned}$$

This directly gives the desired result (4.27):

$$f(x_k) - f(x^\star) \leq (1-c)^k \left(\underbrace{f(x_0) - f(x^\star) + \frac{\mu^2}{2L}\left\|x_0 - x^\star\right\|^2}_{=:M(x_0)} \right) \tag{4.28}$$

□

4.3 The conjugate gradient method

Now, we will see another gradient method with momentum called the *Conjugate Gradient* (CG) method. This method is specific to the case of unconstrained (strictly) convex quadratic programming, i.e., it is for solving the following kinds of problems (where $Q \succ 0$):

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}x^\top Q x - c^\top x + r =: f(x), \tag{4.29}$$

4.3.1 Motivation

As we saw in the previous chapters, the problem (4.29) has a closed-form solution given by $x^\star = Q^{-1}c$. A legitimate question is: *Why would we derive an iterative method to solve* (4.29) *if we already have a closed-form solution?*

The answer is that when $n \gg 1$, the computation of Q^{-1} might be costly. The CG method is a good alternative to solve the problem in this case. More precisely, the computation cost of computing $x^\star$ is on the order of $\frac{n^3}{3}$ (with the Cholesky factorization, which is out of the scope here).
Instead, as we will see in this section, the CG method is very fast (maximum of n iterations). On each iteration, the computational cost is majorated by the computation of matrix-vector product on the form "Qp" (where $Q \in \mathbb{R}^{n\times n}$ and $p \in \mathbb{R}^n$).

An example of how the CG method is useful is when only an approximate solution is needed. In that case, one could run less than n iteration. Therefore, the CG might be cheeper that $\frac{n^3}{3}$ operations (what is needed to compute Q^{-1}).

Another case where the CG method is useful is when the products Qp can be computed efficiently. More precisely, if Q is a sparse matrix, or a sum-product of a few sparse matrices, one can compute the product Qp with much less computational effort than n^2 operations. More precisely, one could have: complexity $(Qp) << n^2$. In that case, running n iterations of the CG method would yield the solution $x^\star$ with a computational cost of $\approx n\,\text{complexity}\,(Qp) << n^3$, which would be much more efficient than the closed-form expression, which requires $\approx \frac{n^3}{3}$ operations. In fact, in the case where Qp can be computed efficiently, the CG method can even be used as a linear solver for the system $Qx = c$.

Example 4.1: Ridge regression with $m << n$

Consider the Ridge regression example (1.12) with $\dim(y_j) = 1$ and $m << n$:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2m}\sum_{j=1}^{m}\left\|y_j - a_j^\top x\right\|^2 + \frac{\lambda}{2}\|x\|^2 . \tag{4.30}$$

Then, the optimization problem (4.30) can be written as a quadratic program (4.29) with $Q = \lambda I_n + \frac{1}{m}\sum_{j=1}^m a_j a_j^\top$.
In that case, the matrix-vector product Qp can be computed with $\approx 2nm << n^2$ operations:

$$Qp = \lambda p + \frac{1}{m}\sum_{j=1}^{m} a_j(a_j^\top p). \tag{4.31}$$

4.3.2 Construction of the CG method

Let us dive into the construction of the CG method. The idea is to take the heavy-ball formulation (4.8), but to let the parameters γ and α vary over time:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k p_k, \\ d_{k+1} &= -\nabla f\big(x_{k+1}\big), \\ p_{k+1} &= \gamma_k p_k + d_{k+1} \end{aligned} \tag{4.32}$$

Here is how the parameters γ_k and α_k are chosen:

- The parameters α_k are found via exact line search:

$$\begin{aligned} \alpha_k &= \arg\min_{\alpha} f\big(x_k + \alpha p_k\big) \\ &= \arg\min_{\alpha} \frac{1}{2}(\alpha p_k)^\top \nabla^2 f(x_k)(\alpha p_k) + \nabla f(x_k)^\top (\alpha p_k) + f(x_k) \end{aligned} \tag{4.33}$$

  Solving equation (4.33) yields:

$$\alpha_k = -\frac{\nabla f(x_k)^\top p_k}{{p_k}^\top \nabla^2 f(x_k) p_k} = \frac{{d_k}^\top p_k}{{p_k}^\top Q p_k} \tag{4.34}$$

- The parameters γ_k are chosen as follows:

$$\gamma_k = \frac{\|d_{k+1}\|^2}{\|d_k\|^2} \tag{4.35}$$

Remark. An important remark is that if $p_k = 0$ or $d_k = 0$, then α_k and γ_k respectively can not be computed. In that case, the algorithm terminates.
Later, we will see that these conditions are equivalent, so only one of them is needed.

Now, using the definition of α and γ_k ((4.34) and (4.35) resp.), we can rewrite the definition of the method (4.32) explicitly.

Definition 4.4: The Conjugate Gradient Method

The Conjugate Gradient (CG) method is the following iterative algorithm:

While $d_k \neq 0$:

$$\alpha_k = \arg\min_\alpha f\big(x_k + \alpha p_k\big) \qquad \left(= \frac{{d_k}^\top p_k}{{p_k}^\top Q p_k} \right) \tag{4.36a}$$

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.36b}$$

$$d_{k+1} = -\nabla f\big(x_{k+1}\big) \qquad \Big(= d_k - \alpha_k Q p_k \Big) \tag{4.36c}$$

$$\gamma_k = \frac{\|d_{k+1}\|^2}{\|d_k\|^2}, \tag{4.36d}$$

$$p_{k+1} = \gamma_k p_k + d_{k+1} \tag{4.36e}$$

The quantity x_0 is initialized with some initial guess, and $p_0 = -\nabla f(x_0) = c - Qx_0$.

Remark. The equations on the left are the most intuitive, while the expressions on the right are the most efficient to compute.

Remark. By storing the value of Qp_k at each iteration, it has to be computed only once per iteration. Hence, the complexity of each iteration is complexity(“Qp”) $+ \mathcal{O}(n)$

4.3.3 Nonlinear extensions

A very active field of research is to extend the CG method to the general non-quadratic case:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x), \tag{4.37}$$

In the litterature, one can find different variations of this. What they have in common is that they all become equivalent to the CG method when the function is quadratic. The most natural extension of CG (in the way we have presented it) is the Fletcher-Reeves method:

Definition 4.5: The Fletcher-Reeves method

The Fletcher-Reeves (FR) method is the following iterative algorithm for solving the problem (1.6):

$$\text{While } \left\|\nabla f\left(x_k\right)\right\| \geq \varepsilon :$$

$$\alpha_k = \arg\min_{\alpha} f\left(x_k + \alpha p_k\right) \tag{4.38a}$$

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.38b}$$

$$\gamma_k = \frac{\left\|\nabla f\left(x_{k+1}\right)\right\|^2}{\left\|\nabla f\left(x_k\right)\right\|^2}, \tag{4.38c}$$

$$p_{k+1} = \gamma_k p_k - \nabla f\left(x_{k+1}\right) \tag{4.38d}$$

Remark. A popular heuristic is to set $\gamma_k = 0$ sometimes.

This is equivalent to a "reset" of the direction p_{k+1}, i.e. $p_{k+1} = -\nabla f(x_{k+1})$.

4.3.4 Some properties of the CG method

It is clear that if the CG method terminates, then the solution is found.

Proposition 4.2: Success of CG in case of termination

If the CG method terminates at iteration k, then x_k is $x^\star$, the global minimizer.

Proof. If the CF method terminates at iteration k, it means that $\nabla f(x_k) = d_k = 0$. Since f is strongly convex, the unique stationary point is the global minimizer: $x_k = x^\star$. □

Now, let us prove some of the properties of the CG method.

Proposition 4.3: d_{k+1} and p_k are orthogonal

At each iteration, the following equality holds:

$${d_{k+1}}^\top p_k = 0 \tag{4.39}$$

Proof. The first-order optimality condition for the line search (4.36a) at iteration k yields:

$$0 = \nabla f(x_k + \alpha_k p_k)^\top p_k = \nabla f(x_{k+1})^\top p_k = -{d_{k+1}}^\top p_k$$

□

Proposition 4.4: p_k is never null

At every iteration k (where the algorithm has not terminated), $p_k \neq 0$.

Remark. This implies that the definition of α_k (4.36a) is well-defined, i.e. ${p_k}^\top Q p_k \neq 0$.

Proof. Since the algorithm has not terminated, we have $d_k \neq 0$.

Using (4.36e) and (4.39), at iteration $k-1$, we have:

$$d_k^\top p_k = \gamma_{k-1} \underbrace{d_k^\top p_{k-1}}_{=0} + d_k^\top d_k = \|d_k\|^2 > 0,$$

which implies that $p_k \neq 0$. □

Proposition 4.5: $\alpha_k \neq 0$

At every iteration k (where the algorithm has not terminated), $\alpha_k \neq 0$

Proof. By contradiction, assume that $\alpha_k = 0$. From equation (4.36c), this implies, we have $d_{k+1} = d_k$. Let us use (4.39) twice to derive the following:

$$\begin{aligned} 0 = {d_{k+1}}^\top p_k &= {d_k}^\top p_k \\ &= {d_k}^\top (\gamma_{k-1} p_{k-1} + d_k) \\ &= \gamma_{k-1} \underbrace{{d_k}^\top p_{k-1}}_{=0} + \|d_k\|^2 \\ &= \|d_k\|^2 \end{aligned}$$

This implies that $d_k = 0$, i.e. the algorithm should have terminated, which is a contradiction. □

4.3.5 The important property

Before we can prove the important theorem about the CG method, we need the following lemma.

Lemma 4.6: Some equations

$$p_{k-1}^\top d_{k+s} = p_{k-1}^\top d_{k+s-1} + \frac{\alpha_{k+s-1}}{\alpha_{k-1}}\left(d_k^\top p_{k+s-1} - \|d_{k+s-1}\|^2\right) \tag{4.40a}$$
$$- \frac{\alpha_{k+s-1}}{\alpha_{k-1}}\left(d_{k-1}^\top p_{k+s-1} - \|d_{k+s-1}\|^2\right)$$
$$d_k^\top p_{k+s} - \|d_{k+s}\|^2 = p_k^\top d_{k+s} - \gamma_{k-1} p_{k-1}^\top d_{k+s} + \gamma_{k+s-1}\left(d_k^\top p_{k+s-1} - \|d_{k+s-1}\|^2\right) \tag{4.40b}$$

Proof. We make the following manipulations for any $k \geq 0$ and $s \geq 0$:

$$\begin{aligned} p_{k-1}^\top d_{k+s} &= p_{k-1}^\top d_{k+s-1} + p_{k-1}^\top (d_{k+s} - d_{k+s-1}) \\ &= p_{k-1}^\top d_{k+s-1} - \alpha_{k+s-1} p_{k-1}^\top Q p_{k+s-1} \\ &= p_{k-1}^\top d_{k+s-1} + \frac{\alpha_{k+s-1}}{\alpha_{k-1}}\left(d_k - d_{k-1}\right)^\top p_{k+s-1} \\ &= p_{k-1}^\top d_{k+s-1} + \frac{\alpha_{k+s-1}}{\alpha_{k-1}}\left(d_k^\top p_{k+s-1} - \|d_{k+s-1}\|^2\right) - \frac{\alpha_{k+s-1}}{\alpha_{k-1}}\left(d_{k-1}^\top p_{k+s-1} - \|d_{k+s-1}\|^2\right) \end{aligned}$$

and

$$\begin{aligned} d_k^\top p_{k+s} - \|d_{k+s}\|^2 &= \gamma_{k+s-1} d_k^\top p_{k+s-1} + d_k^\top d_{k+s} - \|d_{k+s}\|^2 \\ &= d_k^\top d_{k+s} + \gamma_{k+s-1}\left(d_k^\top p_{k+s-1} - \frac{\|d_{k+s}\|^2}{\gamma_{k+s-1}}\right) \\ &= \left(p_k - \gamma_{k-1} p_{k-1}\right)^\top d_{k+s} + \gamma_{k+s-1}\left(d_k^\top p_{k+s-1} - \|d_{k+s-1}\|^2 \frac{\|d_{k+s}\|^2}{\|d_{k+s}\|^2}\right) \\ &= p_k^\top d_{k+s} - \gamma_{k-1} p_{k-1}^\top d_{k+s} + \gamma_{k+s-1}\left(d_k^\top p_{k+s-1} - \|d_{k+s-1}\|^2\right) \end{aligned}$$

□

Now we can prove the following theorem, which is key for the analysis of the CG method.

Theorem 4.2: The key property in CG

Assume that the CG method has not terminated at iteration j. Then, the following holds for all $s \geq 0$ and $k \geq 0$:

$$p_{k-1}^\top d_{k+s} = 0 \tag{4.41a}$$
$$d_k^\top p_{k+s} - \|d_{k+s}\|^2 = 0 \tag{4.41b}$$

(with the convention $p_{-1} = 0$).

Proof. We define:

$$\begin{aligned} a(s,k) &:= p_{k-1}^\top d_{k+s} \\ b(s,k) &:= d_k^\top p_{k+s} - \|d_{k+s}\|^2 . \end{aligned}$$

Then, (4.40) can be summarized as follows:

$$a(s,k) = a(s-1,k) + \frac{\alpha_{k+s}}{\alpha_k} b(s-1,k) - \frac{\alpha_{k+s}}{\alpha_k} b(s,k-1) \tag{4.42a}$$

$$b(s,k) = a(s-1,k+1) - \gamma_{k-1} a(s,k) + \gamma_{k+s-1} b(s-1,k) \tag{4.42b}$$

Now we will prove that $a(s,k) = 0$ and $b(s,k) = 0$ by induction on s.

- **Initialisation** ($s = 0$):
  - Regarding $a(0,k) = p_{k-1}^\top d_{k+s}$, we have $a(0,k) = 0$ directly from (4.39).
  - Regarding $b(0,k)$:

$$\begin{aligned} b(0,k) &= d_k^\top p_k - \|d_k\|^2 \\ &= d_k^\top \Big(\gamma_{k-1} p_{k-1} + d_k \Big) - \|d_k\|^2 \\ &= \gamma_{k-1} d_k^\top p_{k-1} = 0 \quad \text{(cf. (4.39))} \end{aligned}$$

- **Inductive step**: Hypothesis:

$$\forall k \geq 0, \quad a(s-1,k) = 0 \quad \text{and} \quad b(s-1,k) = 0 \tag{4.43}$$

Combine (4.43) with (4.42) to get:

$$a(s,k) = -\frac{\alpha_{k+s}}{\alpha_k} b(s,k-1) \tag{4.44a}$$

$$b(s,k) = -\gamma_{k-1} a(s,k) \tag{4.44b}$$

These imply:

$$\forall k \geq 0, \; a(s,k) = \frac{\alpha_{k+s}\gamma_{k-1}}{\alpha_k} a(s,k-1) \tag{4.45}$$

On the otherhand, $a(s,0) = 0$ because $p_{-1} = 0$.
This, combined with (4.45), implies that $a(s,k) = 0$ for all $k \geq 0$.
Finally, using (4.44b) we also have $b(s,k) = 0$ for all $k \geq 0$.
Overall, we have just proved:

$$\forall k \geq 0, \quad a(s,k) = 0 \quad \text{and} \quad b(s,k) = 0, \tag{4.46}$$

which concludes the induction, hence the proof of the theorem.

□

Corollary 4.1: The gradients are orthogonal

The familly of vectors $\{d_0, \dots, d_{n-1}\}$ is orthogonal:

$$\forall i \neq j \;\; d_i^\top d_j = 0 \tag{4.47}$$

Proof. Without loss of generality it is enough to prove that $d_k^\top d_{k+s+1} = 0$ for all $k \geq 0$ and $s \geq 0$. This can be proven as follows:

$$\begin{aligned} d_k^\top d_{k+s+1} &= \Big(p_k - \gamma_{k-1} p_{k-1}\Big)^\top d_{k+s+1} \\ &= \underbrace{p_k^\top d_{k+s+1}}_{=0} - \gamma_{k-1} \underbrace{p_{k-1}^\top d_{k+s+1}}_{=0} = 0 \end{aligned}$$

The two equalities come from the key property (4.41a) applied to $(s, k+1)$ and $(s+1, k)$ respectively. □

Corollary 4.2: The conjugacy property

The following *conjugacy property* holds:

$$\forall i \neq j, \; p_i^\top Q p_j = 0 \tag{4.48}$$

Proof. Without loss of generality, it is enough to prove that $p_{k-1}^\top Q p_{k+s} = 0$ for all $k \geq 0$ and $s \geq 0$.

$$\begin{aligned} p_{k-1}^\top Q p_{k+s} &= \frac{1}{\alpha_{k+s}} p_{k-1}^\top \big(d_{k+s} - d_{k+s+1}\big) \\ &= \frac{1}{\alpha_{k+s}} \underbrace{p_{k-1}^\top d_{k+s}}_{=0} - \frac{1}{\alpha_{k+s}} \underbrace{p_{k-1}^\top d_{k+s+1}}_{=0} = 0 \end{aligned}$$

The two equalities come from the key property (4.41a) applied to (s, k) and $(s+1, k)$ respectively. □

4.3.6 Termination of the CG method

The CG method has a very nice convergence property: it converges in at most n iterations. This is a consequence of the previous theorem.

Corollary 4.3: Linear independence of p_k

If the algorithm has not terminated at iteration t, then the vectors $d_0, \dots, d_t$ are linearly independent.

Proof. These vectors are non-zero and orthogonal, hence there are linearly independent. □

Corollary 4.4: Termination of CG

The CG method terminates in at most n iterations and yields the global minimizer $x^\star$.

Proof. We prove this corollary by contradiction. Assume that after n iterations, the CG method has not terminated. Using Corollary 4.3, this implies that the vectors $d_0, \dots, d_n$ are linearly independent. But since $\dim(\mathbb{R}^n) = n$, there can not be $n+1$ linearly independent vectors in $\mathbb{R}^n$. Since this is a contradiction, the CG method must terminate in at most n iterations.

Finally, using Proposition 4.2, we can conclude that the solution $x^\star$ is found at the termination of the algorithm. □

4.3.7 Computing Q^{-1} with the CG method

As we have mentioned in part 4.3.1, the CG method can be used as a linear solver for the system $Qx = c$. In fact, the CG method can even be used to compute Q^{-1}.

Proposition 4.6: Computation of Q^{-1}

Assume that the CG method has terminated at iteration n (i.e., the worst case). Then the following holds:

$$Q^{-1} = \sum_{k=0}^{n-1} \frac{1}{{p_k}^\top Q p_k} p_k {p_k}^\top \tag{4.49}$$

Proof. Let us define the matrix J as follows:

$$J := \left(\sum_{k=0}^{n-1} \frac{1}{{p_k}^\top Q p_k} p_k {p_k}^\top \right) Q = \sum_{k=0}^{n-1} \frac{1}{{p_k}^\top Q p_k} p_k {p_k}^\top Q \tag{4.50}$$

Then, for $t = 0, \ldots, n-1$, we have:

$$\begin{aligned} Jp_t &= \sum_{k=0}^{n-1} \frac{1}{{p_k}^\top Q p_k} p_k {p_k}^\top Q p_t \\ &= \sum_{k=0}^{n-1} \frac{1}{{p_k}^\top Q p_k} ({p_k}^\top Q p_t) p_k \\ &= p_t + \sum_{k \neq p_t} \frac{1}{{p_k}^\top Q p_k} \underbrace{{p_k}^\top Q p_t}_{=0} p_k \qquad \text{(cf. Corollary 4.2)} \\ &= p_t \end{aligned} \tag{4.51}$$

This implies that $Jp_t = p_t$ for $t = 0, \ldots, n-1$.
Using Corollary 4.3, the familly vectors $\{p_0, \ldots, p_{n-1}\}$ forms a basis of $\mathbb{R}^n$. This implies that $J = I_n$ (the identity matrix):

$$\left(\sum_{k=0}^{n-1} \frac{1}{{p_k}^\top Q p_k} p_k {p_k}^\top \right) Q = I_n, \tag{4.52}$$

which proves the property (4.49). □

Chapter 5

Stochastic Gradient Descent (SGD)

5.1 Motivation behind Stochastic Gradient Methods

5.1.1 Motivation from a learning point of view

In this course, we will discuss stochastic gradient methods for the case of input-to-output data and where the goal is to learn a model from the input to the output. This setup includes both regression and classification tasks, summarized by the following learning problem:

$$\text{find } x \in \mathbb{R}^n \text{ such that} \quad \hat{y} = \varphi(a; x) \approx y \quad \text{for } (a, y) \text{ in the training dataset} \tag{5.1}$$

where a denotes the inputs (or features), and y the outputs (or labels).

Given a distance $d(\hat{y}, y)$, we can translate the task (5.1) into minimizing $d(\varphi(a; x), y)$ for (a, y) in the dataset.

Remark. For regression tasks, we typically choose $d(\hat{y}, y) = \frac{1}{2} \left\| y - \hat{y} \right\|^2$

For easier notation, we will write $z \coloneqq (a, y)$ for the data samples, and define $l(z, x) \coloneqq d(\varphi(a; x), y)$. Doing so, the task (5.1) can be summarized as follows:

$$\text{find } x \in \mathbb{R}^n \text{ such that} \quad l(z, x) \text{ is as small as possible} \quad \text{for } z \text{ in the training dataset} \tag{5.2}$$

We deliberately leave (5.2) vague to allow it to fit two different (but related) frameworks.

Remark. One can extend this to the case of penalization by setting $l(z, x) \coloneqq d(\varphi(a; x), y) + \lambda \text{pen}(x)$.

Now, we are going to see two different frameworks to model (5.2), both leading to the definition of the stochastic gradient descent method, with unifying notations in the next section.

5.1.2 Minimizing the statistical risk

The first interesting setup is when the data samples $z = (a, y)$ are drawn from a distribution $\mathcal{Z}$.

Example 5.1: Outputs corrupted by Gaussian noise

A typical example is when the output is a function of the input but corrupted by Gaussian noise:

$$y = \varphi(a, x^{\text{true}}) + \varepsilon \tag{5.3}$$

where $\varepsilon \in \mathcal{N}(0, \Sigma)$ is the noise, which follows a normal (=Gaussian) distribution with zero-mean. In that case, the probability density function $p_z(\cdot)$ associated with $z = (a, y)$ can be written as follows:

$$p_z(z) = p_z((a, y)) = p_{\mathcal{N}}\big(y - \varphi(a, x^{\text{true}}); \Sigma\big) \times p_a(a) \tag{5.4}$$

where $p_{\mathcal{N}}(\varepsilon; \Sigma) = \frac{1}{\sqrt{2\pi \det(\Sigma)}} e^{-\frac{1}{2}\varepsilon^\top \Sigma^{-1} \varepsilon}$ is the Gaussian density function of the noise and p_a is the probability density function associated with the inputs a.

A good way to model the task (5.2) is to minimize the *statistical risk*.

Definition 5.1: Statistical risk

Given a data generating distribution $\mathcal{Z}$, and a loss function $l(z, x)$, the *statistical risk* minimization problem is defined as follows:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) := \underset{z \sim \mathcal{Z}}{\mathbb{E}} \big[l(z, x)\big] \tag{5.5}$$

Remark. An interesting fact is that for the example 5.1, the true parameter x^{true} is a solution of the problem (5.5) for the L_2 distance function $d(\bar{y}, y) = \|y - \bar{y}\|^2$.

The issue with the optimization problem (5.5) is that we can not even compute the objective function: only some samples $l(z, x)$ are available. Similarly, the gradient $\nabla f(x)$ can not be computed, but instead, the samples $\nabla_x l(z, x)$ are available.

Proposition 5.1: The gradient is unbiased (1)

When z is sampled from the distribution $\mathcal{Z}$, $\nabla_x l(z, x)$ is an *unbiased* estimate of of the gradient:

$$\underset{z \sim \mathcal{Z}}{\mathbb{E}} [\nabla_x l(z, x)] = \nabla f(x). \tag{5.6}$$

An adaptation of the gradient descent method where the samples $\nabla_x l(z, x)$ replace the true gradients $\nabla f(x)$ is a good candidate to solve (5.5).

Definition 5.2: The Stochastic Gradient method (SGD) for statistical risk minimization

In the case where we draw data samples $z = (a, y)$ from a distribution $\mathcal{Z}$, and the goal is to solve the optimization problem (5.5), the *Stochastic Gradient Method* is the following algorithm:

```
Input some guess $x_0$;
For $\ = 1, \dots, t$ :
    Sample $z_k \sim \mathcal{Z}$;                                (5.7)
    Update $x_{k+1} = x_k - \alpha_k \nabla_x l(z_k, x_k)$;
Output $x_t$.
```

where α_k is the step-size at iteration k.

Remark. This algorithm seems to be computationally cheap, compared to the complexity of the problem (5.5). In the next section, we will discuss another set-up, where similar conclusions can be drawn.

Remark. Since algorithm (5.7) is a weaker version of the gradient descent, it is expected that fewer properties can be proven about it. But surprisingly, the results that we are going to prove in the next section are actually not *that* bad.

5.1.3 The incremental Gradient Method

As opposed to always sampling new data points, we come back to the usual case of a fixed dataset. In the previous chapters, we have modeled the problem (5.2) by minimizing the average loss on the dataset. This is called *Empirical Risk Minimization*(ERM). We define it formally.

Definition 5.3: Empirical Risk Minimization

Given a loss function $l(z, x)$, and a dataset $\mathcal{D} = \{z_1, \dots, z_N\}$, the *Empirical Risk Minimization* problem is defined as follows:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) := \frac{1}{|\mathcal{D}|} \sum_{z \in \mathcal{D}} l(z, x) = \frac{1}{N} \sum_{j=1}^{N} l(z_j, x) \tag{5.8}$$

The *incremental gradient method* is an optimization algorithm to solve (5.8) efficiently when $N \gg 1$. When N is very large, (5.8) becomes similar to (5.5): the loss $f(x)$ and its gradient

$\nabla f(x)$ are too expensive to be computed. Indeed, computing the gradient ∇f requires computing the gradient of $\nabla_x l(z, x)$ for *all* $z \in \mathcal{D}$. Instead, using only a few samples is probably sufficient to get a decent descent direction. Here, we will discuss the case where only *one* sample is used per iteration.

Definition 5.4: The (stochastic) incremental gradient method (SGD)

The *incremental gradient method* is the following algorithm for solving the ERM problem (5.8):

$$\begin{aligned}
&\text{Input some guess } x_0;\\
&\text{For } = 1, \dots, t:\\
&\qquad \text{Sample } j_k \text{ uniformly from } \{1, \dots, N\};\\
&\qquad \text{Update } x_{k+1} = x_k - \alpha_k \nabla_x l(z_{j_k}, x_k);\\
&\text{Output } x_t.
\end{aligned} \tag{5.9}$$

where α_k is the step-size at iteration k.

Proposition 5.2: The gradient is unbiased (2)

Like in (5.6), we have:

$$\mathop{\mathbb{E}}_{j}\left[\nabla_x l(z_j, x)\right] = \frac{1}{N}\sum_{j=1}^{N} \nabla_x l(z_j, x) = \nabla f(x) \tag{5.10}$$

where the expected value is taken with respect to the uniform distribution $\mathcal{U}(\{1, \dots, N\})$.

5.1.4 Minibatching

A variant of the incremental gradient method is to consider more than one sample per iteration. This is called *minibatching*, and is a way to reduce the variance of the gradient estimate.

Definition 5.5: The minibatch incremental gradient method

To solve the ERM problem (5.8), the *minibatch incremental gradient method* is the following algorithm:

```
Input some guess $x_0$;
For $= 1, \dots, t$ :
    Sample $\mathcal{B}_k \subset \{1, \dots, N\}$ of size $p$;
    Update $x_{k+1} = x_k - \alpha_k \nabla \left( \frac{1}{p} \sum_{j \in \mathcal{B}_k} \nabla l(z_j, x_k) \right)$ ;        (5.11)
Output $x_t$.
```

where α_k is the step-size at iteration k.

Remark. A similar extension can be done for the algorithm (5.7) where the statistical risk is minimized.

Proposition 5.3: The gradient is unbiased (3)

Like in (5.6), we have:

$$\mathbb{E}\left[\frac{1}{p} \sum_{j \in \mathcal{B}} \nabla_x l(z_j, x) \right] = \nabla f(x) \tag{5.12}$$

where the expected value is taken with respect to $\mathcal{B} \subset \{1, \dots, N\}$.

5.2 Theory for stochastic gradient descent

5.2.1 A unifying framework

In this section, we are going to abstract from the concrete cases from Definitions 5.1 and 5.3 and consider the general optimization problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \tag{5.13}$$

Here, we will cast all of the algorithms from the previous section into the same framework: we have access to some function that approximates the gradient: $g(x) \approx \nabla f(x)$. Formally, the function g should also depend on something random. Hence, we will denote it by $g(x, \xi)$, where ξ is a random variable, following a distribution $\mathcal{P}$.

Consistently with the definitions (5.2), (5.4) and (5.5), we define the stochastic gradient descent algorithm as follows:

Definition 5.6: The Stochastic Gradient method (SGD)

The *stochastic gradient method* is the following algorithm:

```
Input some guess $x_0$;
For $= 1, \dots, t$ :
    Sample $\xi_k \sim \mathcal{P}$;                          (5.14)
    Update $x_{k+1} = x_k - \alpha_k g(x_k, \xi_k)$;
Output $x_t$.
```

where α_k is the step-size at iteration k.

5.2.2 Key Assumptions

As aleady implied by the three variants of SGD mentioned in the previous section, the key equation is that the random variable that replaces the gradient is equal to the gradient in expected value.

Assumption 5.1: The first key assumption

We assume the following equation about the function g:

$$\underset{\xi \sim \mathcal{P}}{\mathbb{E}} \left[g(x, \xi) \right] = \nabla f(x) \tag{5.15}$$

There is one additional assumption that we are going to need to make this work:

Assumption 5.2: The second key assumption

We assume that for some $\sigma > 0$:

$$\forall x \in \mathbb{R}^n, \quad \underset{\xi \sim \mathcal{P}}{\mathbb{E}} \left[\left\| g(x, \xi) - \nabla f(x) \right\|^2 \right] \leq \sigma^2 \tag{5.16}$$

Remark. This assumption holds whenever the variance of the random variable $g(x, \xi)$ is bounded uniformly in x.

5.2.3 Key inequalities

In this section, we will study the convergence properties of the SGD algorithm. To make it simpler, we will restrict ourselves to the case where f is an L-smooth and μ-strongly convex function. Since the setting here is stochastic, the notion of convergence is delicate. To make things easier, we choose to study the *convergence in expectation*, meaning, the convergence $\mathbb{E}\left[\|x_k - x^\star\|^2\right] \to 0$, where $x^\star$ is the minimizer of f.

Proposition 5.4: Descent equation for L-smooth functions

Let f be an L-smooth function. Let $x_0, x_1, \dots, x_{k+1}$ be the iterates of the SGD algorithm (5.14). Then for all $k \geq 0$:

$$\mathbb{E}\left[f(x_{k+1})\right] \leq \mathbb{E}\left[f(x_k)\right] - \left(\alpha_k - \alpha_k^2 \frac{L}{2}\right) \mathbb{E}\left[\|\nabla f(x_k)\|^2\right] + \alpha_k^2 \frac{L\sigma^2}{2} \tag{5.17}$$

where the expectation is taken w.r.t. *all* the random variables $\xi_0, \xi_1, \dots, \xi_k$.

Proof. To make the proof easier, we define $e_k \coloneqq g(x_k, \xi_k) - \nabla f(x_k)$. Then, Assumptions 5.1 and 5.2 read as:

$$\begin{aligned} \mathop{\mathbb{E}}_{\xi_k}\left[\|e_k\|^2\right] &\leq \sigma^2 \\ \mathop{\mathbb{E}}_{\xi_k}\left[e_k\right] &= 0 \end{aligned}$$

(it is actually the conditional expectation w.r.t. x_k). Then, we write the following:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) - \alpha_k \nabla f(x_k)^\top g(x_k, \xi_k) + \alpha_k^2 \frac{L}{2}\|g(x_k, \xi_k)\|^2 \\ &= f(x_k) - \alpha_k \nabla f(x_k)^\top \left(\nabla f(x_k) + e_k\right) + \alpha_k^2 \frac{L}{2}\|\nabla f(x_k) + e_k\|^2 \\ &= f(x_k) - \left(\alpha_k - \alpha_k^2 \frac{L}{2}\right)\|\nabla f(x_k)\|^2 - \left(\alpha_k - \alpha_k^2 L\right)\nabla f(x_k)^\top e_k + \alpha_k^2 \frac{L}{2}\|e_k\|^2 \end{aligned}$$

Taking the expectation w.r.t. $\xi_0, \ldots, \xi_k$ on both sides, we get:

$$\begin{aligned}
\underset{\xi_0,\ldots,\xi_k}{\mathbb{E}}\left[f(x_{k+1})\right] &\leq \underset{\xi_0,\ldots,\xi_k}{\mathbb{E}}\left[f(x_k) - \left(\alpha_k - \alpha_k^2 \frac{L}{2}\right)\|\nabla f(x_k)\|^2 - \left(\alpha_k - \alpha_k^2 L\right)\nabla f(x_k)^\top e_k + \alpha_k^2 \frac{L}{2}\|e_k\|^2\right] \\
&\leq \underset{\xi_0,\ldots,\xi_{k-1}}{\mathbb{E}}\left[f(x_k) - \left(\alpha_k - \alpha_k^2 \frac{L}{2}\right)\|\nabla f(x_k)\|^2 - \left(\alpha_k - \alpha_k^2 L\right)\nabla f(x_k)^\top \underbrace{\underset{\xi_k}{\mathbb{E}}\left[e_k\right]}_{=0} + \alpha_k^2 \frac{L}{2}\underbrace{\underset{\xi_k}{\mathbb{E}}\left[\|e_k\|^2\right]}_{\leq \sigma^2}\right] \\
&\leq \underset{\xi_0,\ldots,\xi_{k-1}}{\mathbb{E}}\left[f(x_k) - \left(\alpha_k - \alpha_k^2 \frac{L}{2}\right)\|\nabla f(x_k)\|^2 + \alpha_k^2 \frac{L}{2}\sigma^2\right]
\end{aligned}$$

which proves the result (5.17). □

Proposition 5.5: Descent equation for μ-strongly convex functions

Let f be a μ-strongly convex function, and L-smooth. Let $x_0, x_1, \ldots, x_{k+1}$ be the iterates of the SGD algorithm (5.14). Then for all $k \geq 0$:

$$\mathbb{E}\left[f(x_{k+1})\right] - f(x^\star) \leq \left(1 - 2\mu\alpha_k + \alpha_k^2 L\mu\right)\left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right) + \alpha_k^2 \frac{L\sigma^2}{2} \tag{5.18a}$$

$$\mathbb{E}\left[\|x_k - x^\star\|^2\right] \leq \frac{2}{\mu}\left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right) \tag{5.18b}$$

where the expectation is taken w.r.t. *all* the random variables $\xi_0, \xi_1, \ldots, \xi_k$.

Proof. Like in the non-stochastic case, we can prove the following inequality for μ-strongly convex functions:

$$\|\nabla f(x)\|^2 \geq 2\mu\left(f(x) - f(x^\star)\right)$$

Applying this to (5.17) leads to (5.18a)

$$\mathbb{E}\left[f(x_{k+1})\right] - f(x^\star) \leq \left(1 - 2\mu\alpha_k + \alpha_k^2 L\mu\right)\left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right) + \alpha_k^2 \frac{L\sigma^2}{2}$$

Regarding the inequality (5.18b), simply takes the expected value of the following inequality for μ-strongly convex functions:

$$f(x_k) \leq f(x^\star) + \frac{\mu}{2}\|x_k - x^\star\|^2$$

□

The inequalities (5.18) are very similar to the ones for the deterministic case, except that for the term $+\alpha_k^2 \frac{L\sigma^2}{2}$. This term adds an incentive for the stepsize α_k to be small. Interestingly, if the variance term σ^2 is small, then we can have similar convergence rates as in the deterministic case.

5.2.4 Result for a fixed stepsize

Theorem 5.1: Result for a fixed stepsize

Let f be a μ-strongly convex function, and L-smooth. Let $x_0, x_1, \dots, x_{k+1}$ be the iterates of the SGD algorithm (5.14). Then, if a fixed stepsize $\alpha_k = \alpha \in (0, \frac{2}{L})$ is used, then the solution exponentially converges to a ball around the optimal solution:

$$\mathbb{E}\left[\|x_k - x^\star\|^2\right] \leq \rho(\alpha)^k \frac{L}{\mu} \|x_0 - x^\star\|^2 + r(\alpha)^2 \tag{5.19}$$

with some values $\rho(\alpha), r(\alpha)$ dependent on α. For example, for $\alpha = \frac{1}{L}$, we have:

$$\rho(\alpha) = 1 - \frac{\mu}{L} \qquad r(\alpha) = \frac{\sigma}{\mu} \tag{5.20}$$

Remark. We impose $\alpha \in (0, \frac{2}{L})$ to ensure that $\rho(\alpha) \in (0, 1)$.

Remark. We did not define the notion of "convergence to a set". But this is basically "as one would imagine". The ball to which we converge is:

$$\left\{ x \text{ such that } \|x - x^\star\| \leq r(\alpha) \right\}$$

Proof. Taking the inequalities (5.18) and replacing α_k by α leads to:

$$\begin{aligned}
\mathbb{E}\left[f(x_{k+1})\right] - f(x^\star) &\leq \underbrace{\left(1 - 2\mu\alpha + \alpha^2 L\mu\right)}_{=\rho(\alpha)} \left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right) + \alpha^2 \frac{L\sigma^2}{2} \\
&= \rho(\alpha) \left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right) + \alpha^2 \frac{L\sigma^2}{2} \\
&= \rho(\alpha) \left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right) + (1 - \rho(\alpha))\alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))} \\
&= \rho(\alpha) \Bigg(\underbrace{\mathbb{E}\left[f(x_k)\right] - f(x^\star) - \alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))}}_{:=u_k}\Bigg) + \alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))}
\end{aligned}$$

Finally, rearranging a bit the terms, we get:

$$\underbrace{\mathbb{E}\left[f(x_{k+1})\right] - f(x^\star) - \alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))}}_{=u_{k+1}} \leq \rho(\alpha) \Bigg(\underbrace{\mathbb{E}\left[f(x_k)\right] - f(x^\star) - \alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))}}_{=:u_k}\Bigg)$$

Applying this equation recursively, we get $u_k \leq \rho(\alpha)^k u_0$, or more explicitly:

$$\mathbb{E}\left[f(x_k)\right] - f(x^\star) - \alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))} \leq \rho(\alpha)^k \left(f(x_0) - f(x^\star) - \alpha^2 \frac{L\sigma^2}{2(1-\rho(\alpha))}\right)$$

Finally, using the inequality (5.18b):

$$\begin{aligned}
\mathbb{E}\left[\|x_k - x^\star\|^2\right] &\leq \frac{2}{\mu}\Big(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\Big) \\
&\leq \frac{2}{\mu}\left(\alpha^2\frac{L\sigma^2}{2(1-\rho(\alpha))} + \rho(\alpha)^k\left(f(x_0) - f(x^\star) - \alpha^2\frac{L\sigma^2}{2(1-\rho(\alpha))}\right)\right) \\
&\leq \sigma^2\frac{L\alpha^2}{\mu(1-\rho(\alpha))} + \rho(\alpha)^k\frac{2}{\mu}\left(f(x_0) - f(x^\star)\right) \\
&\leq \frac{\sigma^2}{\mu^2}\frac{L\alpha}{2-L\alpha} + \rho(\alpha)^k\frac{2}{\mu}\left(f(x_0) - f(x^\star)\right)
\end{aligned}$$

Using a last rearrangement from the equation $f(x_0) - f(x^\star) \leq \frac{L}{2}\|x_0 - x^\star\|^2$, we find:

$$\mathbb{E}\left[\|x_k - x^\star\|^2\right] \leq \Bigg(\underbrace{\frac{\sigma}{\mu}\sqrt{\frac{L\alpha}{2-L\alpha}}}_{=:r(\alpha)}\Bigg)^2 + \rho(\alpha)^k\frac{L}{\mu}\|x_0 - x^\star\|$$

which concludes the proof with the following values:

$$\rho(\alpha) = 1 - 2\mu\alpha + \alpha^2 L\mu \qquad r(\alpha) = \frac{\sigma}{\mu}\sqrt{\frac{L\alpha}{2-L\alpha}}$$

Replacing $\alpha = \frac{1}{L}$, we find (5.20). □

5.2.5 Result for a decreasing stepsize

As we saw before, for a fixed stepsize $\alpha_k = \alpha$, we can only hope to converge to ball around the solution, with size proportional to σ. This is due to the fact that even if we reach the solution $x^\star$, the iterates x_k would still move because the step $g(x^\star, \xi_k)$ is not necessarily zero.

However, it seems from equation (5.18) that if the stepsizes converge to zero, i.e. $\alpha_k \to 0$, there is still a chance to converge.

More precisely, we show the following theorem:

Theorem 5.2: Convergence of SGD

Let f be a μ-strongly convex function, and L-smooth. Let $x_0, x_1, \dots, x_{k+1}$ be the iterates of the SGD algorithm (5.14). Assume that the stepsize is chosen according to the following form:

$$\alpha_k = \frac{a}{(k+b)^c} \tag{5.21}$$

with $a > 0$, $b > 0$ and $c \in \left(\frac{1}{2}, 1\right)$ threee constants.
Then x_k converges (in expected value) to the solution $x^\star$:

$$\forall k \geq \bar{k}, \quad \mathbb{E}\left[\|x_k - x^\star\|^2\right] \xrightarrow[k \to +\infty]{} 0 \tag{5.22}$$

Proof. Note the following fact:

$$(k+1)\left(2\mu\alpha_k - \alpha_k^2 L\mu\right) = 2\mu a \frac{k+1}{(k+b)^c} - L\mu a^2 \frac{k+1}{(k+b)^{2c}} \xrightarrow[k \to +\infty]{} +\infty$$

Using this fact, there exists k_0 such that for all $k \geq k_0$:

$$(k+1)\left(2\mu\alpha_k - \alpha_k^2 L\mu\right) \geq 1$$

Plugging this into the inequality (5.18a) from Proposition 5.5, we get that for all $k \geq k_0$:

$$\begin{aligned}
\mathbb{E}\left[f(x_{k+1})\right] - f(x^\star) &\leq \left(1 - 2\mu\alpha_k + \alpha_k^2 L\mu\right)\Big(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\Big) + \alpha_k^2 \frac{L\sigma^2}{2} \\
&\leq \left(1 - \frac{1}{k+1}\right)\Big(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\Big) + \alpha_k^2 \frac{L\sigma^2}{2} \\
&\leq \frac{1}{k+1}\left(k\Big(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\Big) + (k+1)\alpha_k^2 \frac{L\sigma^2}{2}\right)
\end{aligned}$$

which implies:

$$\underbrace{(k+1)\Big(\mathbb{E}\left[f(x_{k+1})\right] - f(x^\star)\Big)}_{=u_{k+1}} \leq \underbrace{k\Big(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\Big)}_{=:u_k} + \underbrace{(k+1)\alpha_k^2 \frac{L\sigma^2}{2}}_{=:b_k}$$

where we defined $u_k := k\Big(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\Big)$ and $b_k := (k+1)\alpha_k^2 \frac{L\sigma^2}{2}$.
Hence, for all $k \geq k_0$:

$$u_k \leq u_{k_0} + \sum_{j=k_0+1}^{k} b_j \qquad \leq u_{k_0} + \sum_{j=1}^{k} b_j$$

Furthermore, let us note that:

$$b_k = (k+1)\alpha_k^2 \frac{L\sigma^2}{2} = \frac{a^2 L\sigma^2}{2} \frac{k+1}{(k+b)^{2c}} \sim \frac{a^2 L\sigma^2}{2} \frac{1}{k^{2c-1}} \xrightarrow[k \to +\infty]{} 0$$

because $c > \frac{1}{2}$.
Using Cesaro's Lemma, this implies:

$$\frac{1}{k}\sum_{j=1}^{k} b_j \xrightarrow[k\to+\infty]{} 0$$

Now we can conclude the proof using equation (5.18b) from Proposition 5.5, we get:

$$\begin{aligned}\mathbb{E}\left[\|x_k - x^\star\|^2\right] &\leq \frac{2}{\mu}\left(\mathbb{E}\left[f(x_k)\right] - f(x^\star)\right)\\ &= \frac{2}{\mu}\frac{u_k}{k}\\ &\leq \frac{2}{\mu}\left(\underbrace{\frac{u_{k_0}}{k}}_{\to 0} + \underbrace{\frac{1}{k}\sum_{j=1}^{k} b_j}_{\to 0}\right)\end{aligned}$$

This implies $\mathbb{E}\left[\|x_k - x^\star\|^2\right] \xrightarrow[k\to+\infty]{} 0$, which concludes the proof. □

Remark. By carefully inspecting the proof, we can see that the convergence rate is on the order of $\mathcal{O}\left(\frac{1}{k^{2c-1}}\right)$, which is slightly worst than $\mathcal{O}\left(\frac{1}{k}\right)$, but can be made arbitrarily close to it by choosing c close to 1.

5.3 Practical considerations

5.3.1 A cycle through the dataset instead of sampling

In the incremental stochastic gradient approach described in (5.9), instead of choosing the data index j_k randomly, a very common things to do is to let it cycle through the dataset.
Similarly, for the mini-batch version, one can choose to split the dataset into a couple of batches and cycle through them, instead of always constructing a new random batch.
Doing so, we make sure to go through all the data points of the dataset. When this is done, we say that we have completed an *epoch.*
While this method seems to make more sense because every data point will be used in a similar amount, the convergence analysis of this algorithm is more difficult.

5.3.2 SGD with momentum

Very popular algorithms (yet, primarly based on heuristics) mix the ideas from methods using momentum (cf. Chapters 4) and the ideas from Stochastic gradient methods (cf. Chapters 5).

Differing a bit from the orginal momentum idea, now the idea is to use previous iterations to estimate the variance of the gradients and use this information to the direction.

Definition 5.7: Adagrad (adaptive gradient)

The most simple algorithm using this idea is the Adagrad algorithm:

$$v_{k+1} = v_k + g(x_k, \xi_k)^2 \tag{5.23a}$$

$$x_{k+1} = x_k - \alpha \frac{g(x_k, \xi_k)}{\sqrt{v_k} + \varepsilon} \tag{5.23b}$$

where v_k represents the estimate on the variance of the gradients up to iteration k.

Remark. In (5.23), the square, the root and the divisions here are done component-wise on vectors.

Definition 5.8: RMSprop (Root Mean Square Propagation)

The RMSprop algorithm is a slight modification of the Adagrad algorithm, where a forgetting factor $\beta \in (0, 1)$ is inctroduced for the variance estimates:

$$v_{k+1} = \beta v_k + (1 - \beta) g(x_k, \xi_k)^2 \tag{5.24a}$$

$$x_{k+1} = x_k - \alpha \frac{g(x_k, \xi_k)}{\sqrt{v_k} + \varepsilon} \tag{5.24b}$$

Definition 5.9: Adam (Adaptive Moment Estimation)

Adam is probably the most popular algorithm of optimization in deep learning. It combines the ideas from the Heavy-ball method and RMSprop:

$$p_{k+1} = \beta_1 p_k + (1 - \beta_1) g(x_k, \xi_k) \tag{5.25a}$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2) g(x_k, \xi_k)^2 \tag{5.25b}$$

$$x_{k+1} = x_k - \alpha \frac{p_{k+1}}{\sqrt{v_{k+1}} + \varepsilon} \tag{5.25c}$$

Remark. It is actually slightly more complicate, there is a correction due to the initial bad guess $p_k = 0$ and $v_k = 0$. The correction is that in (5.25c), we replace p_{k+1} and v_{k+1} by $\frac{p_{k+1}}{1-\beta_1^k}$ and $\frac{v_{k+1}}{1-\beta_2^k}$ respectively.

5.3.3 Usefull python libraries

- `Tensorflow` & `Pytorch`: the two most popular libraries for optimization for neural networks. They use backward automatic differentiation to compute the gradients of the loss function with respect to the parameters.
- `CasADi`: a symbolic framework for automatic differentiation and optimization.

It is more flexible: the derivatives of multi-output functions can be computed, and it can be used for optimization problems with constraints.
It can call *IPOPT*, a powerful constrained optimization solver, using the interior point method.

- `cvxpy`: a library for some specific convex optimization problems they called “disciplined” (e.g. Lasso, QPs, etc).

5.3.4 Optimization vocabulary used in Machine Learning

- **Full batch**: method where the whole dataset is used to compute the gradient at each iteration.
- **Mini-batch**: method where a small batch of the dataset is used to compute the gradient at each iteration.
- **Epoch**: number of times the algorithm goes through the whole dataset.
- **Learning rate**: step size α_k of the algorithm.
- **Risk / statistical risk**: expected value of the loss function: $f(x) = \mathbb{E}_\xi\left[f(x,\xi)\right]$.
- **Empirical risk**:average of the loss function over the dataset: $f(x) = \frac{1}{N}\sum\limits_{i=1}^{N} f(x,\xi_i)$.
- **Training / Validation / Test**
  - **Training**: the set of data used to train the algorithm, i.e. to compute the gradient.
  - **Validation set**: the set of data used to tune the hyperparameters of the algorithm or to design a stopping criterion.
  - **Test set**: the set of data used to evaluate the performance of the algorithm *after* the training phase.
- **Generalization**: ability of the trained-model to perform well on unseen data.
- **Overfitting**: phenomenon where the trained-model performs well on the training data but poorly on the test data.

Appendix A

Very basics of mathematics

In this small chapter, we provide a few basic mathematical concepts that are used throughout the book. These are basic definitions and well-known properties/theorems that will be stated without proof.

The goal is to have written support that can be referred to when needed. Note that the definitions, theorems, and properties mentioned here are not necessarily used in the rest of the script, but it is good to have them in mind when reading the script or attending the course.

A.1 Basics of linear algebra

Definition A.1: Euclidean norm

The norm $\|\cdot\|$ denotes the L2-norm, also called the *Euclidean norm* defined as follows:

$$\forall x \in \mathbb{R}^n, \quad \|x\| := \sqrt{x^\top x} = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{A.1}$$

Proposition A.1: The Cauchy-Schwarz inequality

If x and y are vectors in $\mathbb{R}^n$, then:

$$\left|x^\top y\right| \leq \|x\| \, \|y\| \tag{A.2}$$

Definition A.2: Eigenvalues of a matrix

Let $A \in \mathbb{R}^{n \times n}$ be a square matrix. An eigenvalue of A is a scalar $\lambda \in \mathbb{C}$ such that there exists a non-zero vector $x \in \mathbb{C}^n$ satisfying:

$$Ax = \lambda x \tag{A.3}$$

We write $\text{sp}(A)$ the set of eigenvalues of A.

Theorem A.1: Spectral theorem

Let $S \in \mathbb{R}^{n \times n}$ be a symmetric matrix (i.e. $S^\top = S$).
Then $\text{sp}(S) \subset \mathbb{R}$, i.e. the eigenvalues of S are real.
Furthermore, S is orthogonally diagonalizable:

$$S = P \Lambda P^\top \tag{A.4}$$

where $P \in \mathbb{R}^{n \times n}$ is an orthogonal matrix, i.e. such that $P^\top P = I$, and Λ is a diagonal matrix whose diagonal elements are the eigenvalues of S (possibly repeated).

Definition A.3: Positiveness of symmetric matrices

Let $S \in \mathbb{R}^{n \times n}$ be a symmetric matrix.
We say that S is positive semi-definite (resp. positive definite) if for all $x \in \mathbb{R}^n \setminus \{0\}$, $x^\top S x \geq 0$ (resp. $x^\top S x > 0$).
We denote this property by $S \succeq 0$ (resp. $S \succ 0$).

For $S_1, S_2 \in \mathbb{R}^{n \times n}$ two symmetric matrices, we write $S_1 \succcurlyeq S_2$ (resp. $S_1 \succ S_2$) when $S_1 - S_2 \succcurlyeq 0$ (resp. $S_1 - S_2 \succ 0$).

Proposition A.2: Characterization of positive symmetric matrices

Let $S \in \mathbb{R}^{n \times n}$ be a symmetric matrix. The two following properties are equivalent:

$$S \succcurlyeq 0 \text{ (resp. } S \succ 0) \tag{A.5a}$$

$$\forall \lambda \in \text{sp}(S),\ \lambda \geq 0 \text{ (resp. } \lambda > 0) \tag{A.5b}$$

Proposition A.3: Inequalities for symmetric matrices

Let $S \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Let λ_1, λ_2 be two scalars. Then the three following properties are equivalent:

$$\lambda_1 I_n \preccurlyeq S \preccurlyeq \lambda_2 I_n \tag{A.6a}$$
$$\forall x \in \mathbb{R}^n, \quad \lambda_1 \left\|x\right\|^2 \leq x^\top S x \leq \lambda_2 d x^2 \tag{A.6b}$$
$$\forall \lambda \in \mathrm{sp}(S), \qquad \lambda_1 \leq \lambda \leq \lambda_2 \tag{A.6c}$$

Definition A.4: Orthogonal sets

Let $E \subset \mathbb{R}^n$ be a vectorial subspace of $\mathbb{R}^n$. Then, we define the orthogonal of E as:

$$E^\perp = \{x \in \mathbb{R}^n \mid \forall y \in E,\ x^\top y = 0\} \tag{A.7}$$

Note that $E^\perp$ is also a vectorial subspace of $\mathbb{R}^n$.

Proposition A.4: Inclusion of orthogonal sets

Then $F^\perp \subset E^\perp$. Let $E, F \subset \mathbb{R}^n$ be two vectorial subspaces of $\mathbb{R}^n$ such that $E \subset F$.

Proposition A.5: Orthogonal sets of images and kernels

Let $A \in \mathbb{R}^{n\times m}$ be a matrix. Then the following properties hold:

$$\mathrm{Im}(A)^\perp = \mathrm{Ker}(A^\top) \tag{A.8}$$
$$\mathrm{Ker}(A)^\perp = \mathrm{Im}(A^\top) \tag{A.9}$$

A.2 Basics of differential calculus

Definition A.5: Differentiability

Let $\mathcal{X} \subset \mathbb{R}^n$. Let $f : \mathcal{X} \to \mathbb{R}$ be a function.
We say that f is differentiable at $x \in \mathbb{R}^n$ if there exists a vector $\nabla f(x) \in \mathbb{R}^n$ such that:

$$\forall d \in \mathbb{R}^n, \quad \frac{f(x + \varepsilon d) - f(x)}{\varepsilon} \xrightarrow[\varepsilon \to 0]{} \nabla f(x)^\top d \tag{A.10}$$

The vector $\nabla f(x)$ is called the gradient of f at x.
We say that f is differentiable on $\mathcal{X}$ if it is differentiable at all points of $\mathcal{X}$.
If $\nabla f(x)$ is a continuous function, we say that f is continuously differentiable.

Definition A.6: Twice differentiable functions

We say that f is twice differentiable if it is differentiable, and if there exists a matrix $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ such that:

$$\forall x \in \mathcal{X},\ \forall d \in \mathbb{R}^n, \quad \frac{\nabla f(x + \varepsilon d) - \nabla f(x)}{\varepsilon} \xrightarrow[\varepsilon \to 0]{} \nabla^2 f(x) d \tag{A.11}$$

The matrix $\nabla^2 f(x)$ is called the Hessian of f at x.
We say that f is twice continuously differentiable if $\nabla^2 f(x)$ is continuous.

Theorem A.2: Schwarz's theorem

If f is twice continuously differentiable, then $\nabla^2 f(x)$ is symmetric for all $x \in \mathcal{X}$.

Proposition A.6: First order Taylor's approximation

Assume that f is continuously differentiable. Then the following formula holds:

$$f(x + d) = f(x) + \nabla f(x)^\top d + r(x, d) \tag{A.12}$$

where $r(x, d)$ is defined as follows:

$$r(x, d) = \left(\int_0^1 \nabla f(x + sd) - \nabla f(x) \mathrm{d}s \right)^\top d \tag{A.13}$$

Proposition A.7: First-order Taylor's approximation with limits

The following holds:

$$\frac{r(x,d)}{\|d\|} \xrightarrow[d\to 0]{} 0 \tag{A.14}$$

Proposition A.8: First order Taylor's approximation for a twice differentiable function

Furthermore, if f is twice continuously differentiable, then:

$$r(x,d) = d^\top \left(\int_0^1 s\nabla^2 f(x+sd)\mathrm{d}s \right) d \tag{A.15}$$

Proposition A.9: Inequality for first order Taylor's approximation

If f is twice continuously differentiable, and:

$$\forall s \in [0,1], \quad \lambda_{\min} I_n \preccurlyeq \nabla^2 f(x+sd) \preccurlyeq \lambda_{\max} I_n, \tag{A.16}$$

then:

$$\lambda_{\min} \|d\|^2 \leq r(x,d) \leq \lambda_{\max} \|d\|^2 \tag{A.17}$$

A.3 Basics of topology

Definition A.7: Neighborhoods

Let $x \in \mathbb{R}^n$. We say that $\mathcal{N} \subset \mathbb{R}^n$ is a neighborhood of x if there exists an $\varepsilon > 0$ such that:

$$\forall y \in \mathbb{R}^n, \quad \|x-y\| < \varepsilon \Rightarrow y \in \mathcal{N}. \tag{A.18}$$

Definition A.8: Open sets

We say that x is in the interior of a set $\mathcal{X}$ when it admits a neighborhood $\mathcal{N}$ such that $\mathcal{N} \subset \mathcal{X}$.

Definition A.9: Open sets

We say that a set $\mathcal{O} \subset \mathbb{R}^n$ is open when it is its own interior.

Definition A.10: Closed-set

We say that a set $\mathcal{C} \subset \mathbb{R}^n$ is closed when its complement $\mathbb{R}^n \setminus \mathcal{C}$ is open.

Remark. Interestingly, the empty set and $\mathbb{R}^n$ are both open and closed.

Proposition A.10: Characterization of closed-sets

A set $\mathcal{C}$ is closed if and only if when $(x_k)_{k\in\mathbb{N}}$ in $\mathcal{C}$ converges to $x \in \mathbb{R}^n$, $x \in \mathcal{C}$.

Definition A.11: Compact sets

We say that $\mathcal{K} \subset \mathbb{R}^n$ is compact when is closed and bounded, i.e.

$$\exists M > 0, \quad \forall x \in \mathcal{K}, \quad \|x\| \leq M \tag{A.19}$$

Definition A.12: Accumulation points

Let $(x_k)_{k\in\mathbb{N}}$ be a sequence in $\mathbb{R}^n$. We say that $\bar{x} \in \mathbb{R}^n$ is an accumulation point of the sequence $(x_k)_{k\in\mathbb{N}}$ if there exists an increasing sequence of integers k_j such that $x_{k_j} \xrightarrow[j\to\infty]{} \bar{x}$.

Theorem A.3: Bolzano-Weierstrass theorem

If $\mathcal{K} \subset \mathbb{R}^n$ is a compact set, then any sequence $(x_k)_{k\in\mathbb{N}}$ in $\mathcal{K}$ has at least an accumulation point in $\mathcal{K}$.

Bibliography

[1] Stephen J Wright and Benjamin Recht. *Optimization for data analysis*. Cambridge University Press, 2022.