MA333

Optimisation for Machine Learning lecture notes

2022/23

Prof. László Végh
l.vegh@lse.ac.uk

Preface

Machine learning combines tools from statistics, mathematics, and computer science for a broad range of problems in data analytics. Applications of machine learning have transformed society over the last two decades. These recent advancements, in particular in deep learning, resulted from simultaneous improvements in the methodology and the scaling up of computational hardware and data availibility.

The purpose of this course is to focus on one of the fundamental ingredients, *optimization*. At a high level, most key problems in machine learning are optimization problems: for $K \subseteq \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$, solve

$$\min_{x \in K} f(x) \, .$$

The data set is already encoded inside the function f. For example, the variables can be certain parameters, and we are trying to find a parametric function of a certain form that best fits our training dataset. This also includes neural networks where the parameters are the different weights, and the function is the output of the network at the given inputs.

Solving such optimization problems are *hard*: not only in the everyday meaning of the word, but also in a strict sense in computational complexity. The first step in tackling them is to understand the special assumptions under which they become tractable. The key here is *convexity*: in case the domain K is a convex set and f is a convex function, optimality can be characterised *locally*.

Theory and algorithms for convex optimisation have been developed since the 1950s. The methods particularly important for machine learning are those that can be implemented at scale and speed, since the algorithms must work on massive datasets. The most important such family is *first order methods*: these algorithms use only simple local information such as the gradient of the function at a certain point, and follow the descent direction defined by the gradient.

Even though *gradient descent* is a very simple method, there is a rich and ever expanding theory of different variants and implementations. Functions arising in machine learning can have various forms, representations, and properties: we will see a range of variants to suit such requirements.

Second order methods are allowed to use more information about the functions, such as the Hessians. This may not be available or prohibitively expensive to compute. But when such information is available, we can obtain faster and more powerful optimisation algorithms.

Many important problems however require solving *non-convex* optimisation problems. We cannot expect to optimally solve them in general. Still, convex optimisation provides powerful methods that can also be applied in this context, even though the performance guarantees are missing in most cases.

There are several excellent textbooks and lecture notes available. Much of the course material is based on the recent book *Algorithms for convex optimization* by Vishnoi [6]. Further recommended books are *Convex optimization* by Boyd and Vandenberghe [1], *Lectures on convex optimization* by Nesterov, *First-order and stochastic optimization methods for machine learning* by Lan [5], and the lecture notes *Optimization for machine learning* by Gärtner and Jaggi [2].

The course focuses on the optimisation aspects on machine learning, and we do not discuss in detail the statistical background. As a strating point we recommend *An introduction to statistical learning* by James, Witten, Hastie, and Tibshirani [4].

Contents

Chapter 1

Preliminaries

Norms and the Cauchy–Schwarz inequality For a vector $x \in \mathbb{R}^n$ and $p \in [1, \infty)$, the *ℓ_p-norm* of the vector is defined as

$$\|x\|_p = (|x_1|^p + \ldots + |x_n|^p)^{1/p} .$$

For $p = \infty$, we have $\|x\|_\infty = \max_i |x_i|$; this is also called the *maximum norm*. We will most frequently use the ℓ_2-norm; if not specified otherwise, $\|x\|$ will refer to $\|x\|_2$.

The *standard inner product* is $\langle x, y \rangle = x^\top y = \sum_{i=1}^n x_i y_i$ for vectors $x, y \in \mathbb{R}^n$. In particular, $\|x\| = \sqrt{\langle x, y \rangle}$. Recall the Cauchy-Schwarz inequality.

Theorem 1.1 (Cauchy–Schwarz–Bunyakovski)**.** *For $x, y \in \mathbb{R}^n$, we have*

$$| \langle x, y \rangle | \leq \|x\| \cdot \|y\| .$$

Further, equality holds if and only if x and y are linearly dependent, that is, $x = \alpha y$ for some $\alpha \in \mathbb{R}$.

Norms can be defined more generally, according to the following definition:

Definition 1.2. *(Norm) A* norm *is a function $\|.\| : \mathbb{R}^n \to \mathbb{R}_+$ such that*

(i) $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$ *for any $x \in \mathbb{R}^n$ and $\alpha > 0$,*

(ii) $\|x\| = 0$ *if and only if $x = 0$, and*

(iii) $\|x + y\| \leq \|x\| + \|y\|$ *(triangle inequality).*

Concepts from topology We restrict ourselves to the standard Euclidean topology with ℓ_2-norm. For $x \in \mathbb{R}^n$ and $R > 0$, the open ball centered at x with radius $R > 0$ is $\{y \in \mathbb{R}^n : \|x - y\| < R\}$. A set $K \subseteq \mathbb{R}^n$ is *open* if K contains an open ball around every point $x \in K$. A set $K \subseteq \mathbb{R}^n$ is *closed* if $\mathbb{R}^n \setminus K$ is open. We let $\text{int}(K)$ denote the *interior* of K, which is the set of points $x \in K$ such that K contains an open ball around x.

The *closure* $\text{cl}(K)$ of K is the unique smallest closed set containing K. Note that if K is closed then $\text{cl}(K) = K$. Equivalenly, $\text{cl}(K)$ is the set of all limit points of convergent sequences in K. We let $\partial K := \text{cl}(K) \setminus \text{int}(K)$ denote the *boundary* of K.

The diameter of a set $K \subseteq \mathbb{R}^n$ is $\sup\{\|x - y\| : x, y \in K\}$; the set K is *bounded*, if its diameter is finite. The set K is *compact* if it is closed and bounded.

1.1 Linear algebra

A *hyperplane* in $\mathbb{R}^n$ is a set of the form $\{x \in \mathbb{R}^n \mid \langle a, x \rangle = \beta\}$, where a is a nonzero vector of $\mathbb{R}^n$ and $\beta \in \mathbb{R}$.
A *half-space* in $\mathbb{R}^n$ is a set of the form $\{x \in \mathbb{R}^n \mid \langle a, x \rangle \leq \beta\}$, where a is a nonzero vector of $\mathbb{R}^n$ and $\beta \in \mathbb{R}$.
A *polyhedron* in $\mathbb{R}^n$ is the intersection of a finite number of half-spaces. Equivalently, a polyhedron is a set that can be written in the form $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Recall that this is the feasible region of a linear programming (LP) problem.

Linear combinations, linear spaces

The vector $x \in \mathbb{R}^n$ is a *linear combination* of the vectors $x^1, \ldots, x^q \in \mathbb{R}^n$ if there exist scalars $\lambda_1, \ldots, \lambda_q$ such that

$$x = \sum_{j=1}^{q} \lambda_j x^j.$$

The vectors $x^1, \ldots, x^q \in \mathbb{R}^n$ are *linearly independent* if $\lambda_1 = \ldots = \lambda_q = 0$ is the unique solution to the system $\sum_{j=1}^{q} \lambda_j x^j = 0$.

A nonempty subset $\mathcal{L}$ of $\mathbb{R}^n$ is a *linear space* if $\mathcal{L}$ is closed under taking linear combinations, i.e. every linear combination of vectors in $\mathcal{L}$ belongs to $\mathcal{L}$. A subset $\mathcal{L}$ of $\mathbb{R}^n$ is a linear space if and only if $\mathcal{L} = \{x \in \mathbb{R}^n \mid Ax = 0\}$ for some matrix $A \in \mathbb{R}^{m \times n}$.

A *basis* of a linear space $\mathcal{L}$ is a maximal set of linearly independent vectors in $\mathcal{L}$. All bases have the same cardinality. This cardinality is called the *dimension* of $\mathcal{L}$. If $\mathcal{L} = \{x \in \mathbb{R}^n \mid Ax = 0\}$, then the dimension of $\mathcal{L}$ is $n - \operatorname{rank}(A)$. We note that a hyperplane in $\mathbb{R}^n$ is an $n-1$ dimensional linear subspace.

The inclusionwise minimal linear space containing a set $S \subseteq \mathbb{R}^n$ is the *linear space generated* by S, and is denoted by $\operatorname{span}(S)$. Given any maximal set S' of linearly independent vectors in S, we have that $\operatorname{span}(S) = \operatorname{span}(S')$.

Convex combinations, convex sets

A point x in $\mathbb{R}^n$ is a *convex combination* of the points $x^1, \ldots, x^q \in \mathbb{R}^n$ if there exist *nonnegative* scalars $\lambda_1, \ldots, \lambda_q \geq 0$ such that

$$x = \sum_{j=1}^{q} \lambda_j x^j, \quad \sum_{j=1}^{q} \lambda_j = 1.$$

In particular, given three points x, y, z in $\mathbb{R}^n$, the point z is a *convex combination* of x and y if there exists $\lambda \in [0,1]$ such that $z = \lambda x + (1-\lambda)y$, that is, z is contained in the line segment joining x and y. This line segment will also be denoted as $[x,y] = \{\lambda x + (1-\lambda)y \mid 0 \leq \lambda \leq 1\}$. If $x \neq y$ and $\lambda \in (0,1)$, then we say that z is a *proper convex combination* of x and y.

Definition 1.3. *A set $C \subseteq \mathbb{R}^n$ is* convex *if C contains all convex combinations of points in C. Equivalently, $C \subseteq \mathbb{R}^n$ is convex if for any two points $x, y \in C$, the line segment $[x,y]$ is contained in C.*

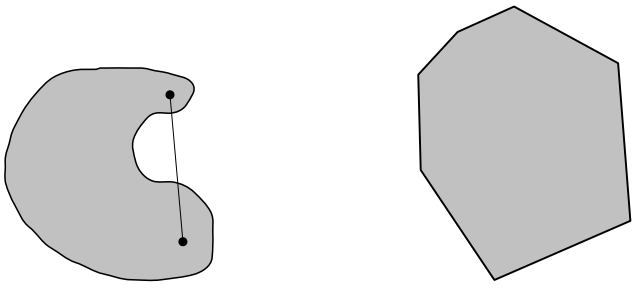


Figure 1.1: The set on the left is not convex, the one on the right is convex.

It is an easy exercise to show that every half-space is convex, and that the intersection of convex sets is convex. This shows that every polyhedron (and thus the feasible region of an LP problem) is a convex set.

Given a set $S \subseteq \mathbb{R}^n$, the *convex hull* of S, denoted by $\operatorname{conv}(S)$, is the inclusionwise minimal convex set containing S. As the intersection of convex sets is a convex set, $\operatorname{conv}(S)$ exists. Moreover, it is the

set of all points that are convex combinations of points in S. That is,

$$\mathrm{conv}(S) = \left\{ \sum_{j=1}^{q} \lambda_j x^j \mid x^1, \ldots, x^q \in S,\ \lambda_1, \ldots, \lambda_q \geq 0,\ \sum_{j=1}^{q} \lambda_j = 1 \right\}.$$

1.1.1 Orthogonal projections

For a nonempty closed convex set $K \subseteq \mathbb{R}^n$, we let $\Pi_K : \mathbb{R}^n \to \mathbb{R}$ denote the *orthogonal projection* to K, defined as follows. For $x \in \mathbb{R}^n$, $\Pi_K(x) \in K$ denotes the point in K at the minimum ℓ_2-distance from x, that is,

$$\Pi_K(x) := \arg\min_{v \in K} \|x - v\| .$$

By the closedness of K and the strong convexity of the ℓ_2-norm (defined later), there is a unique such point.

Orthogonal projections will be particularly important for linear spaces: let $\mathcal{L} \subseteq \mathbb{R}^n$ be a linear space. Then, it can be shown that $\Pi_{\mathcal{L}}(x)$ is a linear operator. That is, there exists a matrix $P \in \mathbb{R}^{n \times n}$ such that $\Pi_{\mathcal{L}}(x) = Px$. It is easy to see that P is a symmetric matrix (see Section 1.1.3) and $P^2 = P$.

1.1.2 Separating and supporting hyperplanes for convex sets

We now derive a basic but very crucial consequence of convexity.

Definition 1.4 (Separating and supporting hyperplanes)**.** *Let $K \subseteq \mathbb{R}^n$ be a convex set, and $y \in \mathbb{R}^n \setminus K$. The hyperplane $H = \{x \in \mathbb{R}^n \mid \langle a, x \rangle = \beta\}$* separates *$y$ from K if*

$$\langle a, x \rangle \leq \beta \quad \forall x \in K , \quad \textit{and} \quad \langle a, y \rangle > \beta .$$

If $y \in \partial K$ is on the boundary, then H is a supporting hyperplane at *y if*

$$\langle a, x \rangle \leq \beta \quad \forall x \in K , \quad \textit{and} \quad \langle a, y \rangle = \beta .$$

Theorem 1.5. *Let $K \subseteq \mathbb{R}^n$ be a nonempty closed convex set. For every $y \in \mathbb{R}^n \setminus K$, there exists a hyperplane that separates y from K, and for every $y \in \partial K$ there exists a supporting hyperplane at y.*

Proof. For $y \in \mathbb{R}^n \setminus K$, let $x^* = \Pi_K(y)$ denote the projection of y to K. Let us define

$$a := y - x^* \quad \text{and} \quad \beta := \langle a, x^* \rangle .$$

Clearly, $a \neq 0$. We claim that $H = \{x \in \mathbb{R}^n \mid \langle a, x \rangle = \beta\}$ separates y from K.

The part $\langle a, y \rangle > \beta$ follows from the definition:

$$\langle a, y \rangle - \beta = \langle y - x^*, y \rangle - \langle y - x^*, x^* \rangle = \|y - x^*\|^2 > 0 .$$

We need to show $\langle a, x \rangle \leq \beta$ for any $x \in K$. Let us pick any $\varepsilon \in (0, 1]$. By convexity of K, $\bar{x} = (1 - \varepsilon)x^* + \varepsilon x = x^* + \varepsilon(x - x^*) \in K$. Since x^* was chosen as a minimum-norm point, we have

$$\|y - \bar{x}\|^2 \geq \|y - x^*\|^2 .$$

Rewriting $y - \bar{x} = (y - x^*) - \varepsilon(x - x^*)$, this yields

$$\|y - x^*\|^2 - 2\varepsilon \langle y - x^*, x - x^* \rangle + \varepsilon^2 \|x - x^*\|^2 \geq \|y - x^*\|^2 .$$

Rearranging and dividing by 2ε,

$$\frac{\varepsilon}{2} \|x - x^*\|^2 \geq \langle y - x^*, x - x^* \rangle .$$

Since this is true for any choice of $\varepsilon > 0$, we can conclude that

$$0 \geq \langle y - x^*, x - x^* \rangle = \langle a, x - x^* \rangle = \langle a, x \rangle - \beta , \tag{1.1}$$

completing the proof.

For the second part, let $y \in \partial K$ be on the boundary. Let us select a sequence of points $y_i \in \mathbb{R}^n \setminus K$, $y_i \to y$ (the existence of such a sequence follows from y being on the boundary). For each such point, we get a separating hyperplane $H_i = \{x \in \mathbb{R}^n \mid \langle a_i, x\rangle = \beta_i\}$. Since $a_i \neq 0$, we can normalise such that $\|a_i\| = 1$ for each i; it is easy to check that the β_i's should also be bounded. We can thus select a subsequence where (a_i, β_i) is also convergent, let (a, β) denote the limit. Then, it is easy to verify that $H = \{x \in \mathbb{R}^n \mid \langle a, x\rangle = \beta\}$ is a supporting hyperplane at y. □

1.1.3 Symmetric matrices

Let $A \in \mathbb{R}^{n\times n}$ be a *symmetric* matrix, that is, $a_{ij} = a_{ji}$ for every i, j. We say that A is *positive definite* if, for all $x \in \mathbb{R}^n \setminus \{0\}$, $x^\top Ax > 0$. We say that A is *positive semidefinite* if, for all $x \in \mathbb{R}^n$, $x^\top Ax \geq 0$.

Further, we say that A is *negative definite* if $-A$ is positive definite, that is, if for all $x \in \mathbb{R}^n \setminus \{0\}$, $x^\top Ax < 0$. Similarly, A is *negative semidefinite*, if, for all $x \in \mathbb{R}^n$, $x^\top Ax \leq 0$.

Clearly, the identity matrix is positive definite. Also, a diagonal matrix with all positive entries is positive definite (PD), and a diagonal matrix with all nonnegative entries is positive semidefinite (PSD). A basic property is that positive and negative definite matrices are invertible:

Lemma 1.6. *If $A \in \mathbb{R}^{n\times n}$ is positive definite or negative definite, then A is invertible.*

Proof. Recall that the matrix $A \in \mathbb{R}^{n\times n}$ is invertible if and only if the n column vectors are independent, that is, $Ax = 0$ for $x \in \mathbb{R}^n$ implies $x = 0$. Let A be positive definite, and for a contradiction assume there exists an $x \in \mathbb{R}^n$, $x \neq 0$ such that $Ax = 0$. We get a contradiction as $0 < x^\top Ax = x^\top 0 = 0$. The analogous argument works for negative definite matrices. □

A well-known example of positive semidefinite matrices is the covariance matrix of a random vector. Indeed, recall that, if $p \in \mathbb{R}^n$ is a random vector with mean $\bar{p}$, then its covariance matrix is the matrix Σ whose (i, j) entry is $\Sigma_{ij} = \mathbb{E}[(p_i - \bar{p}_i)(p_j - \bar{p}_j)]$. One can easily compute that, for all $x \in \mathbb{R}^n$ $\operatorname{Var}\langle p, x\rangle = x^\top \Sigma x$. Since the variance of random variable is always nonnegative, it follows that $x^\top \Sigma x \geq 0$ for all $x \in \mathbb{R}^n$.

A matrix can be neither positive nor negative semidefinite. Such matrices are called *indefinite*. For example, let $A = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix}$. Then, for $x = \binom{1}{0}$, $x^\top Ax = 1$, and for $y = \binom{0}{1}$, $y^\top Ay = -2$.

Definition 1.7 (Eigenvalues and eigenvectors)**.** *For a square matrix $A \in \mathbb{R}^{n\times n}$, $\lambda \in \mathbb{R}$ is an* eigenvalue *if there exists a vector $v \in \mathbb{R}^n$ such that $Av = \lambda v$.*

We can recognise positive/negative (semi)definite matrices based on their eigenvalues.

Theorem 1.8. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. The following hold.*

(i) A is positive definite if and only if all its eigenvalues are positive. A is negative definite if and only if all its eigenvalues are negative.

(ii) A is positive semidefinite if and only if all its eigenvalues are nonnegative. A is negative semidefinite if and only if all its eigenvalues are nonpositive.

Proof. We only prove the statements for positive (semi)definite matrices; the statements for a negative (semi)definite A then follow by applying these to $-A$.

Part (i) *"only if" direction:* Assume A has a negative eigenvalue $\lambda < 0$. For the corresponding eigenvector $v \in \mathbb{R}^n$, we have $v \neq 0$ and $Av = \lambda v$. Then, $v^\top Av = v^\top(\lambda v) = \lambda\|v\|^2 < 0$.

"if" direction: Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of A (with multiplicities). It is well-known that any symmetric matrix A can be orthogonally diagonalised. That is, we can write

$$A = P^\top DP,$$

where P is an orthogonal matrix, and D is a diagonal matrix with the diagonal entries being the eigenvalues: $D_{ii} = \lambda_i$. Consider now any vector $x \in \mathbb{R}^n$, and let $y = Px$. Then,

$$x^\top Ax = x^\top P^\top DPx = y^\top Dy = \sum_{i=1}^{n} \lambda_i y_i^2.$$

Since P is nonsingular, $y = 0$ if and only if $x = 0$. If $\lambda_i > 0$ for all i, then it follows that $x^\top Ax > 0$ whenever $x \neq 0$.

Part (ii) Follows the same way, replacing > 0 by ≥ 0. □

Several different matrix norms are used; a fundamental one is the following.

Definition 1.9 (Spectral norm)**.** *For $A \in \mathbb{R}^{n\times m}$, the* spectral norm *or $\ell_2 \to \ell_2$* norm *is defined as*

$$\|A\|_2 := \sup_{x\in\mathbb{R}^m} \frac{\|Ax\|}{\|x\|}.$$

Theorem 1.10. *If $A \in \mathbb{R}^{n\times n}$ is a positive semidefinite matrix, then $\|A\|_2$ equals the largest eigenvalue of A.*

Ordering of positive semidefinite matrices For positive semidefinite (PSD) matrices $P, Q \in \mathbb{R}^{n\times n}$, we say that P is *PSD-smaller* than Q, denoted by $P \preceq Q$, if $Q - P$ is also PSD matrix. Equivalently, this means that for any vector $v \in \mathbb{R}^n$, $v^\top Pv \leq v^\top Qv$. Thus, P is a PSD matrix if and only if $P \succeq 0$. We also use $P \prec Q$ if $Q - P$ is positive definite; thus, P is positive definite if and only if $P \succ 0$.

1.2 Gradients and Hessians

Functions and graphs Given a function $f : \mathbb{R}^n \to \mathbb{R}$, we denote by $\mathbf{dom}\, f$ the *domain* of f, that is, the set of points x in $\mathbb{R}^n$ for which $f(x)$ is defined. For example, the domain of the function $x \mapsto \log x$ is the set $\{x \in \mathbb{R} \,|\, x > 0\}$, whereas the domain of the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $(x_1, x_2) \mapsto x_1/x_2$ is the set $\mathbf{dom}\, f = \{(x_1, x_2) \in \mathbb{R}^2 \,|\, x_2 \neq 0\}$.

Definition 1.11 (Graph and epigraph)**.** *The* graph *of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the set $\{(x, f(x)) \in \mathbb{R}^{n+1} \,|\, x \in \mathbf{dom}\, f\}$, and the* epigraph *is the set of points in $\mathbb{R}^{n+1}$ that "lie above" the graph of f, that is, the set $\{(x, t) \in \mathbb{R}^{n+1} \,|\, x \in \mathbf{dom}\, f,\ f(x) \leq t\}$.*

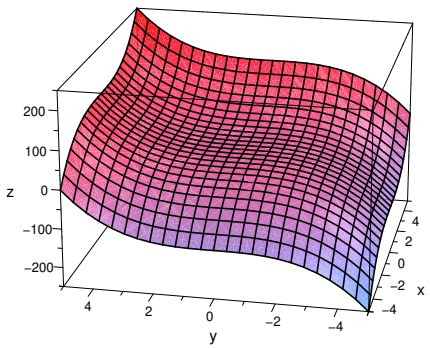


Figure 1.2: Graph of the function $f(x_1, x_2) = x_1^3 + x_2^3$. The epigraph of f is the region above the shaded surface.

Gradients The *gradient* of the function f at point $x \in \mathbf{dom}\, f$ is the vector

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

where $\frac{\partial f(x)}{\partial x_i}$ is the i-th partial derivative of f at point x, i.e., the derivative of f at point x taken with respect to the variable x_i. In particular, when $n = 1$ (i.e. f is a function of one variable), the gradient is simply the derivative of f.

Definition 1.12 (Differentiable function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is* differentiable *at a point x in the interior of* $\mathbf{dom}\, f$ *if*

$$\lim_{z \in \mathbf{dom}\, f,\, z \to x} \frac{|f(z) - f(x) - \langle \nabla f(x), z - x \rangle|}{\|z - x\|} = 0.$$

Geometrically, differentiability at x means that, near x, f is well approximated by the affine function $h : \mathbf{dom}\,(f) \to \mathbb{R}$ defined by $h(z) = f(x) + \langle \nabla f(x), z - x \rangle$.

We say that a continuous function f is *differentiable* if $\mathbf{dom}\, f$ is an open set and f is differentiable at every point $x \in \mathbf{dom}\, f$.

For example, the function $f : \mathbb{R}^2 \to \mathbb{R}$, defined over $\mathbf{dom}\, f = \{x \in \mathbb{R}^2 \,|\, x_1, x_2 > 0\}$ by $f(x_1, x_2) = \log(x_1/x_2)$, is differentiable, and its gradient at any point $x \in \mathbf{dom}\, f$ is

$$\nabla f(x) := \begin{bmatrix} 1/x_1 \\ -1/x_2 \end{bmatrix}.$$

The function $f : x \mapsto |x|$ is not differentiable, because its derivative does not exist at $x = 0$.

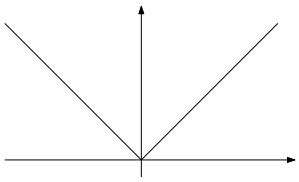


Figure 1.3: Graph of the function $x \mapsto |x|$.

Recall that, if f is differentiable at $\bar{x}$ in the interior of $\mathbf{dom}\, f$, then the gradient of f at $\bar{x}$ points in the direction of steepest ascent of the graph of f at point $(\bar{x}, f(\bar{x}))$. More formally, the hyperplane in $\mathbb{R}^{n+1}$ defined by

$$H = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \,|\, t = f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle\},$$

is the hyperplane tangent to the graph of f at point $(\bar{x}, f(\bar{x}))$. The direction of steepest ascent for the tangent hyperplane H is the direction of the gradient $\nabla f(\bar{x})$, and the slope of H in the direction of $\nabla f(\bar{x})$ is the magnitude $\|\nabla f(x)\|$ of the gradient. This will be further discussed in Section 5.

Another geometric interpretation is that the gradient at point $\bar{x}$ is a vector orthogonal to the contour of f at point $\bar{x}$ – that is, the set $\{x \in \mathbb{R}^n \,|\, f(x) = f(\bar{x})\}$ – pointing in the direction of ascent of the function at $\bar{x}$ (see Figure 1.5).

1.2.1 Directional derivatives

Given $f : \mathbb{R}^n \to \mathbb{R}$, a point $\bar{x} \in \mathbf{dom}\, f$ and a vector $p \in \mathbb{R}^n$, we can define the univariate function $g_p : \mathbb{R} \to \mathbb{R}$ as $g(t) := f(\bar{x} + tp)$. The *directional derivative* of f at point $\bar{x}$ in the direction p is the derivative of g_p at $t = 0$, that is

$$\frac{\partial f}{\partial p}(\bar{x}) = g_p'(0).$$

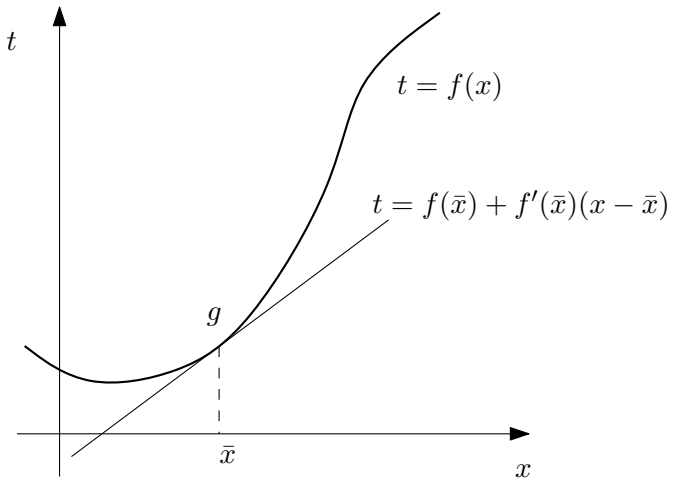


Figure 1.4: For functions of one variables, the gradient at point $\bar{x}$ is the slope of the tangent to the graph at point $(\bar{x}, f(\bar{x}))$.

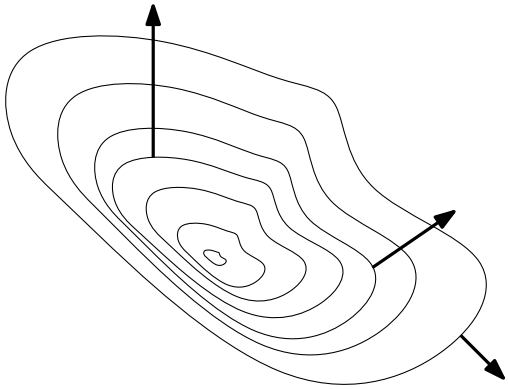


Figure 1.5: For functions of one variables, $\nabla f(\bar{x})$ is orthogonal to the contour at $\bar{x}$ and pointing in the direction of ascent.

The geometric meaning of the above is that the directional derivative $\partial f(\bar{x})/\partial p$ measures the rate of change of f at point $\bar{x}$ when moving in the direction of p.

If f is differentiable, then we have

$$\frac{\partial f}{\partial p}(\bar{x}) = \langle \nabla f(\bar{x}), p \rangle \ . \tag{1.2}$$

We now give an important property of directional derivatives that can be derived from the fundamental theorem of calculus we recall here.

Theorem 1.13 (Fundamental theorem of calculus, second part)**.** *Let $f : [a, b] \to \mathbb{R}$ be a continuously differentiable univariate function. Then,*

$$\int_a^b \dot{f}(t)dt = f(b) - f(a)\,.$$

It is easy to show the following corollary:

Lemma 1.14. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, let* **dom** f *be convex, and let $x, y \in$* **dom** f*. Let us define $g : [0, 1] \to \mathbb{R}$ as*

$$g(t) := f(x + t(y - x))\,.$$

Then, the following hold:

(i) $\dot{g}(t) = \langle \nabla f(x + t(y - x)), y - x \rangle$, *and*

(ii) $f(y) = f(x) + \int_0^1 \dot{g}(t)dt$.

1.2.2 The Hessian

The function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be *twice differentiable* if f and ∇f are both differentiable. Note that ∇f is differentiable if, for $j = 1, \ldots, n$, the function $x \mapsto (\nabla f(x))_j$ defined by the j-th component of ∇f is differentiable. The *Hessian* of a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at point $x \in \mathbf{dom}\, f$ is the $n \times n$ matrix

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{bmatrix}$$

that is, the (i, j) entry of $\nabla^2 f(x)$ is the second partial derivative of f at x with respect to the variables x_i and x_j. Note that $\nabla^2 f(x)$ is symmetric since $\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}$.

If $n = 1$, then $\nabla^2 f$ is simply the second derivative of f. Assume that $f(x) = \sum_{i=1}^n f_i(x_i)$, where $f_i : \mathbb{R} \to \mathbb{R}$ is a univariate function in x_i for each $i = 1, 2, \ldots, n$. Then the Hessian is a diagonal matrix where the ith entry is $f_i''(x_i)$.

1.2.3 Taylor expansion

Recall the second order Taylor-expansion of a univariate function.

Theorem 1.15. *Assume that $f : \mathbb{R} \to \mathbb{R}$ is twice differentiable on $\mathbf{dom}\, f$. Then for every $x, y \in \mathbf{dom}\, f$, we can write*

$$f(x) = f(y) + f'(y)(x - y) + \frac{1}{2} f''(\bar{x})(x - y)^2$$

for some $\bar{x} \in [x, y]$.

The Taylor expansion shows that for a small $\varepsilon > 0$, the function f in the interval $[y - \varepsilon, y + \varepsilon]$ can be well approximated by the linear function $f(y) + f'(y)(x - y)$. We now give the extension of Theorem 1.15 to multivariate functions.

Theorem 1.16 (Taylor expansion of multivariate functions)**.** *Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable on $\mathbf{dom}\, f$. Then for every $x, y \in \mathbf{dom}\, f$, we can write*

$$f(x) = f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2}(x - y)^\top \nabla^2 f(\bar{x})(x - y)$$

for some $\bar{x} \in [x, y]$.

1.3 Global and local optima

Given a function $f : \mathbb{R}^n \to \mathbb{R}$ and a set $X \subseteq \mathbf{dom}\, f$, we say that a point $x^* \in X$ is a *global minimum* for f in X if $f(x) \geq f(x^*)$ for all $x \in X$. We say that $x^* \in X$ is a *global maximum* for f in X if $f(x) \leq f(x^*)$ for all $x \in X$.

A point $x^* \in X$ is said a *local minimum* for f in X if there exists $\varepsilon > 0$ such that $f(x) \geq f(x^*)$ for all $x \in X$ such that $\|x - x^*\| \leq \varepsilon$. We say that $x^* \in X$ is a *local maximum* for f in X if there exists $\varepsilon > 0$ such that $f(x) \leq f(x^*)$ for all $x \in X$ such that $\|x - x^*\| \leq \varepsilon$.

When $X = \mathbf{dom}\, f$, we refer simply to global or local maxima or minima.

Consider, for example, the single variable function whose graph is represented in Figure 1.6. The point x' is a local minimum of f, because we can find a small interval around x' where no point has value smaller than x'. However, x' is not a global minimum because there are points with lower objective value.

We recall the following facts from calculus concerning local optima of unconstrained problems.

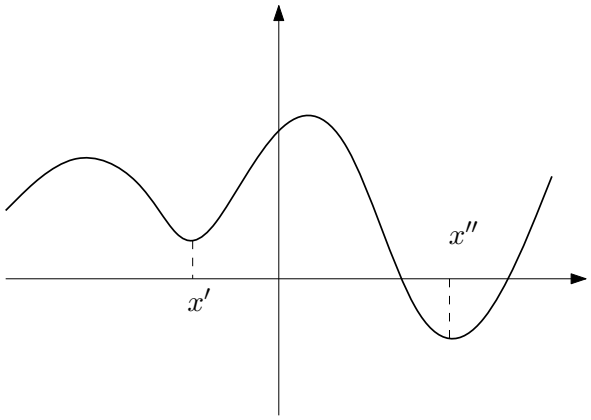


Figure 1.6: Point x' is a local minimum, but not a global minimum.

Theorem 1.17 (First-order necessary conditions)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable, and let x^* be a point in* $\mathbf{dom}\, f$*. If x^* is a local maximum or a local minimum for f, then $\nabla f(x^*) = 0$.*

A point x^* such that $\nabla f(x^*) = 0$ is called a *critical point.* Hence, every local optimum is a critical point, but there can also be further critical points, see Figure 1.7.

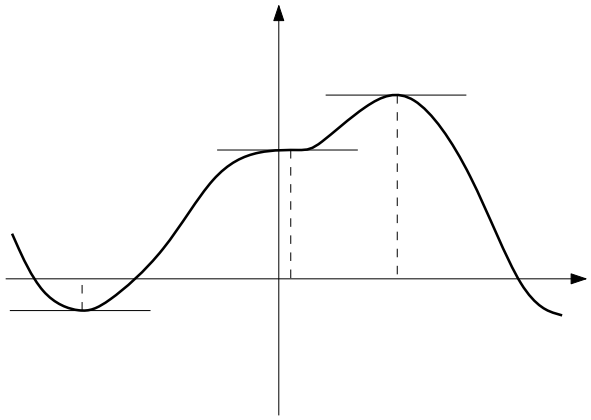


Figure 1.7: Three points with zero gradient. From left to right, the first point is a local minimum, the second a saddle point, and the third a local maximum.

1.3.1 Second-order criteria for critical points

If the function is twice differentiable, the Hessian can be used to analyse critical points. Recall that for a twice differentiable univariate function $f : \mathbb{R} \to \mathbb{R}$, if x^* is a local minimum (maximum), then $f'(x^*) = 0$, and $f''(x^*) \geq 0$ ($f''(x^*) \leq 0$). This statement naturally extends to multivariate functions.

Theorem 1.18. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable, and let x^* be a critical point in* $\mathbf{dom}\, f$*. Then*

(i) If the Hessian $\nabla^2 f(x^*)$ is positive definite, then x^* is a local minimum.

(ii) If x^* is a local minimum for f, then $\nabla^2 f(x^*)$ is positive semidefinite.

(iii) If the Hessian $\nabla^2 f(x^*)$ is negative definite, then x^* is a local maximum.

(iv) If x^* is a local maximum for f, then $\nabla^2 f(x^*)$ is negative semidefinite.

Note the gap between *(i)* and *(ii)*, as well as between *(iii)* and *(iv)*. If the Hessian is positive semidefinite or negative semidefinite, we cannot infer anything about x^*. As an example, consider the

univariate $f(x) = x^3$ at $x^* = 0$. We have $\nabla f(0) = 0$ and $\nabla^2 f(0) = 0$; this is at the same time a positive and a negative semidefinite 1×1 matrix.

Definition 1.19 (Saddle point)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable, and let x^* be a critical point in* $\mathbf{dom}\, f$*. If $\nabla f^2(x^*)$ is indefinite (has both positive and negative eigenvalues), then x^* is called a* saddle point*.*

From a saddle point, we can find directions where the function value increases, as well as directions where it decreases, see Figure 1.8.

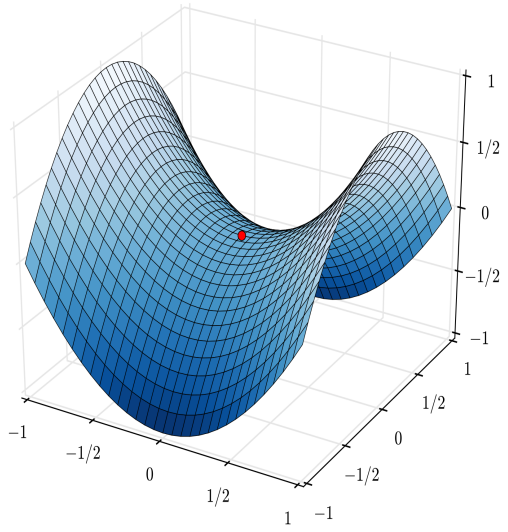


Figure 1.8: Saddle point of a function.

Chapter 2

Basic concepts in convex optimization

2.1 Convex functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if $\mathbf{dom}\, f$ is convex and, for every $x, y \in \mathbf{dom}\, f$, and $\lambda \in [0,1]$,

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)\,. \tag{2.1}$$

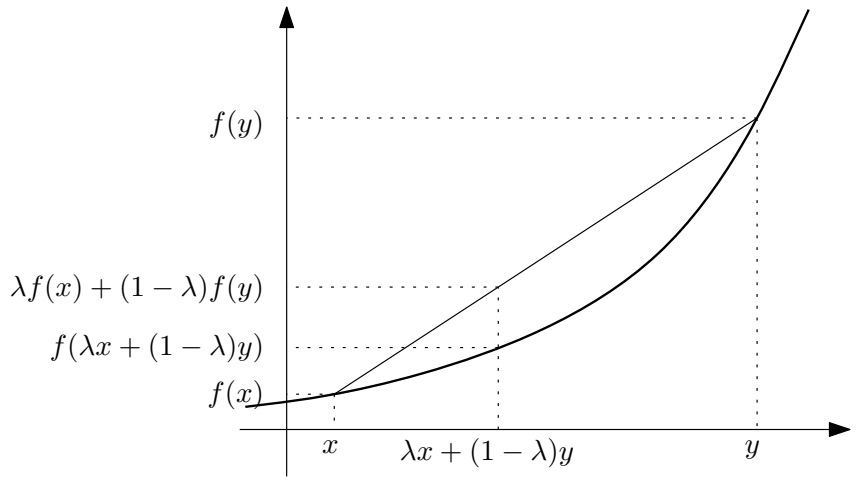


Figure 2.1: A function f is convex if the line segment joining any two points in the graph of f is contained in the epigraph of f.

Figure 2.1 provides a geometric interpretation of the above definition. Note the point $(\lambda x + (1-\lambda)y, f(\lambda x + (1-\lambda)y))$ in $\mathbb{R}^{n+1}$ is a point on the graph of f, therefore (2.1) says that the point $(\lambda x + (1-\lambda)y, \lambda f(x) + (1-\lambda) f(y))$ belongs to the epigraph of f. Observe that the set of all points $(\lambda x + (1-\lambda)y, \lambda f(x) + (1-\lambda) f(y))$ for $\lambda \in [0,1]$ is the line segment joining $(x, f(x))$ to $(y, f(y))$, therefore (2.1) means that the epigraph of f contains the line segment joining any two points in the graph of f. This implies the following.

Proposition 2.1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the epigraph of f is a convex set.*

The simplest example of convex functions are *affine functions*. The function $f : \mathbb{R}^n \to \mathbb{R}$ is *affine* if there exists $p \in \mathbb{R}^n$ and $r \in \mathbb{R}$ such that $f(x) = \langle p, x \rangle + r$; note that linear functions are affine functions where $r = 0$.

It will be convenient to extend the definition of a convex function f to the points not in the domain, by considering $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, where $f(x) = +\infty$ for every $x \notin \mathbf{dom}\, f$. Note that this convention respects the definition of convexity given by (2.1). Indeed, if one among $x, y \in \mathbb{R}^n$ is not in $\mathbf{dom}\, f$, then the right-hand-side of (2.1) is $+\infty$, and the inequality is still verified. With this notation, it follows that $\mathbf{dom}\, f = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$.

We remark, without proof, that convex functions are always continuous.

Theorem 2.2. *If a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex and* $\mathbf{dom}\, f$ *is open, then f is continuous on* $\mathbf{dom}\, f$.

2.1.1 Univariate convex functions

Let us recall the familiar case of univariate functions $f : \mathbb{R} \to \mathbb{R}$. A practical test involves the second derivatives.

Theorem 2.3. *Assume that $f : \mathbb{R} \to \mathbb{R}$ is twice differentiable on* $\mathbf{dom}\, f$. *Then f is convex* if and only if *the second derivative f'' is nonnegative on* $\mathbf{dom}\, f$.

Using this criterion, we can easily verify the convexity of the following univariate functions.

- $f(x) = -\log x$, where $\mathbf{dom}\, f = \{x \in \mathbb{R} \,|\, x > 0\}$.
- $f(x) = e^x$, where $\mathbf{dom}\, f = \mathbb{R}$.
- $f(x) = 1/x$, where $\mathbf{dom}\, f = \{x \in \mathbb{R} \,|\, x > 0\}$. Observe that $f(x) = 1/x$ can be defined over $\mathbb{R} \setminus \{0\}$, but it is not convex over the negative reals.
- $f(x) = x \log x$, where $\mathbf{dom}\, f = \{x \in \mathbb{R} \,|\, x > 0\}$.

2.1.2 Simple constructions of convex functions

Let us show two simple operations that enable constructing convex functions from other convex functions. The first one is *nonnegative linear combination.*

Proposition 2.4. *If $f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex and $\gamma_1, \ldots, \gamma_m \geq 0$, then $f = \gamma_1 f_1 + \cdots + \gamma_m f_m$ is convex.*

Proof. We need to show that for every $x, y \in \mathbf{dom}\, f$, and $\lambda \in [0, 1]$, $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$. This follows using the convexity of the f_i's:

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= \sum_{i=1}^{m} \gamma_i f_i(\lambda x + (1-\lambda)y) \leq \sum_{i=1}^{m} \gamma_i (\lambda f_i(x)) + (1-\lambda) f_i(y)) \\ &= \lambda \sum_{i=1}^{m} \gamma_i f_i(x) + (1-\lambda) \sum_{i=1}^{m} \gamma_i f_i(y) = \lambda f(x) + (1-\lambda) f(y). \end{aligned}$$

□

The second operation is taking *point-wise supremum.* This is illustrated in Figure 2.2.

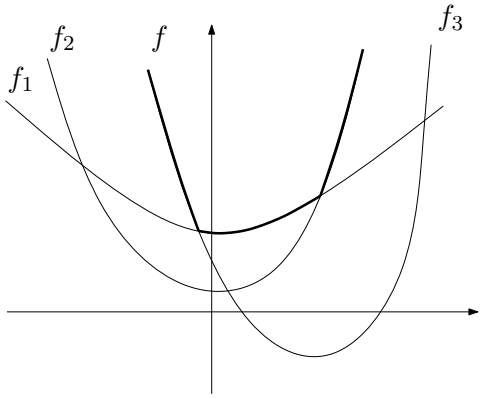


Figure 2.2: The graph of the point-wise maximum of the three convex functions in the picture is in boldface.

Proposition 2.5. *If $f_\alpha : \mathbb{R}^n \to \mathbb{R}$ ($\alpha \in \mathcal{A}$) is a family of convex functions indexed by the elements of a set $\mathcal{A}$ (possibly infinite), then the function f defined by*

$$f(x) := \sup_{\alpha \in \mathcal{A}} f_\alpha(x)$$

is convex.

We present two different proofs.

Proof 1. Let $K_\alpha := \{(x,t) \in \mathbb{R}^{n+1}, t \geq f_\alpha(x)\}$ denote the epigraph of f_α and $K := \{(x,t) \in \mathbb{R}^{n+1}, t \geq f(x)\}$ the epigraph of f.

Recall from Proposition 2.1 that a function is convex if and only if its epigraph is convex. Thus, each K_α is convex. It is easy to see that $K = \cap_{\alpha \in \mathcal{A}} K_\alpha$. Since the intersection of any number of convex sets is convex, it follows that K is convex. By the equivalence in Proposition 2.1, we get the convexity of f. □

Proof 2. Again, we need to show that for every $x, y \in \mathbf{dom}\, f$, and $\lambda \in [0,1]$, $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$. This follows by showing that for any $\varepsilon > 0$, $f(\lambda x + (1-\lambda)y) - \varepsilon \leq \lambda f(x) + (1-\lambda) f(y)$.

Let us now select an arbitrary $\varepsilon > 0$. By the definition of supremum, there exists an $\alpha \in \mathcal{A}$ such that $f(\lambda x + (1-\lambda)y) - \varepsilon \leq f_\alpha(\lambda x + (1-\lambda)y)$. Then we use the convexity of f_α:

$$f(\lambda x + (1-\lambda)y) - \varepsilon \leq f_\alpha(\lambda x + (1-\lambda)y) \leq \lambda f_\alpha(x) + (1-\lambda) f_\alpha(y) \leq \lambda f(x) + (1-\lambda) f(y).$$

The last inequality follows by the definition of f.

We note that if $\mathcal{A}$ is a finite set, then ε is not needed. In this case, we always have an $\alpha \in \mathcal{A}$ such that $f(\lambda x + (1-\lambda)y) = f_\alpha(\lambda x + (1-\lambda)y)$. □

2.1.3 First order characterisation of convexity

Theorem 2.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then f is convex if and only if, for all $x, y \in \mathbf{dom}\, f$,*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle \ .$$

Proof. "⇒" Assume f is convex. By (2.1), for every λ, $0 < \lambda \leq 1$, it follows

$$f(y) + \lambda(f(x) - f(y)) = \lambda f(x) + (1-\lambda) f(y) \geq f(\lambda x + (1-\lambda)y) = f(y + \lambda(x-y)).$$

Subtracting $f(y)$ on both sides and dividing by λ on both sides, we get

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

for $0 < \lambda \leq 1$. Taking the limit for $\lambda \to 0^+$,

$$f(x) - f(y) \geq \lim_{\lambda \to 0^+} \frac{f(y + \lambda(x-y)) - f(y)}{\lambda} = \frac{\partial f(y)}{\partial (x-y)} = \langle \nabla f(y), x - y \rangle \ ,$$

where $\partial f(y)/\partial(x-y)$ is the directional derivative at y along the vector $x - y$, and the last equation follows from (1.2).

"⇐" Assume $f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle$ for all $x, y \in \mathbf{dom}\, f$. For all $y \in \mathbf{dom}\, f$, define the function $f_y : x \mapsto f(y) + \langle \nabla f(y), x - y \rangle$. Note that the function f_y is affine, therefore it is convex. Also, by assumption $f(x) \geq f_y(x)$ for all $x \in \mathbf{dom}\, f$, and by definition $f(x) = f_x(x)$. It follows that, for all $x \in \mathbf{dom}\, f$,

$$f(x) = \max_{y \in \mathbf{dom}\, f} f_y(x).$$

Thus $f(x)$ is the point-wise supremum of a family of convex functions, and is therefore convex by Proposition 2.5. □

For a geometric intuition of Theorem 2.6, recall that the hyperplane tangent to the graph of f at point $(y, f(y))$ is $H = \{(x,t) \in \mathbb{R}^n \times \mathbb{R} \,|\, t = f(y) + \langle \nabla f(y), x - y\rangle$, therefore the theorem states that a function is convex if and only if, for every y, the graph of the function lies above the hyperplane tangent to the graph of f at point $(y, f(y))$.

For a twice differentiable univariate function $f : \mathbb{R} \to \mathbb{R}$, the univariate Taylor expansion (Theorem 1.15) gives that for $x, y \in \mathbf{dom}\, f$, there exists a $\bar{x} \in [x, y]$ such that

$$f(x) = f(y) + f'(y)(x-y) + \frac{1}{2} f''(\bar{x})(x-y)^2 \geq f(y) + f'(y)(x-y)\,,$$

where the inequality follows by Theorem 2.3. This gives an alternative proof for the first direction in Theorem 2.6 for the twice differentiable case. Note however that Theorem 2.6 and the above proof are valid even for functions that are not twice differentiable.

Bregman divergence An important quantity in the analysis of gradient methods is defined as follows:

Definition 2.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. The* Bregman divergence *of f at $x, y \in \mathbf{dom}\, f$ is*

$$D_f(x,y) := f(x) - f(y) - \langle \nabla f(y), x - y\rangle\ .$$

Note that this is *not symmetric*: $D_f(x,y) \neq D_f(y,x)$ is possible. According to Theorem 2.6, f is convex if and only if $D_f(x,y) \geq 0$ for all $x, y \in \mathbf{dom}\, f$.

For a convex function, we can think of $f(y) + \langle \nabla f(y), x - y\rangle$ as a lower estimate on $f(x)$; the Bregman divergence $D_f(x,y)$ measures the gap between the estimate and the actual value. Under certain assumptions on this function, we will show lower and upper bounds on $D_f(x,y)$.

2.1.4 Minima of convex functions

The following is a fundamental property of convex functions.

Theorem 2.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, and let $X \subseteq \mathbf{dom}\, f$ be a convex set. Then every local minimum of f in X is also a global minimum of f in X.*

Proof. Let x^* be a local minimum for f in X. By definition, there exists $\varepsilon > 0$ such that $f(x^*) \leq f(x)$ for all $x \in X$ such that $\|x - x^*\| < \varepsilon$. Suppose by contradiction that x^* is not a global minimum. Then there exists a point $y \in X$ such that $f(y) < f(x^*)$. Select a point $\bar{x} := \lambda x^* + (1-\lambda)y$ for some $\lambda \in [0,1)$ such that $\|\bar{x} - x^*\| < \varepsilon$. By convexity of X, $\bar{x} \in X$, and by convexity of f,

$$\begin{aligned} f(\bar{x}) &= f(\lambda x^* + (1-\lambda)y) \\ &\leq \lambda f(x^*) + (1-\lambda) f(y) = f(x^*) + (1-\lambda)(f(y) - f(x^*)) < f(x^*), \end{aligned}$$

a contradiction. □

Furthermore, the first order necessary conditions in Theorem 1.17 are also sufficient for convex functions.

Theorem 2.9 (First order conditions for unconstrained convex minimization)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function. A point $x^* \in \mathbf{dom}\, f$ is a global minimum of f if and only if $\nabla f(x^*) = 0$.*

Proof. We have already established in Theorem 1.17 that $\nabla f(x^*) = 0$ if x^* is a global minimum of f. Conversely, assume $\nabla f(x^*) = 0$. By Theorem 2.6, for every $x \in \mathbf{dom}\, f$,

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), x - x^*\rangle = f(x^*).$$

It follows that $f(x^*)$ is a global minimum. □

The above theorem holds for *unconstrained* convex minimization problems, i.e., problems where we want to find the global minimum over the entire domain. The next theorem gives necessary and sufficient conditions for the case where we want to find the minimizer in a given convex set X.

Theorem 2.10 (First order conditions for constrained convex minimization)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function, and let $X \subseteq \mathbf{dom}\, f$ be a convex set. A point $x^* \in X$ is a global minimum of f over X if and only if*

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0 \text{ for all } x \in X.$$

Proof. We first prove the "if" direction. That is, assume $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ for all $x \in X$. It follows by Theorem 2.6, for every $x \in \mathbf{dom}\, f$,

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

which implies that x^* is a global minimum for f over X.

For the "only if" direction, assume x^* is a global minimum over X. Given any $x \in X$, note that the point $\lambda x + (1-\lambda)x^*$ is in X for every λ such that $0 < \lambda \leq 1$, because X is convex. Since x^* is a global minimum over X, and noting that $\lambda x + (1-\lambda)x^* = x^* + \lambda(x - x^*)$ it follows that

$$f(x^*) \leq f(x^* + \lambda(x - x^*))$$

for every λ such that $0 < \lambda \leq 1$. Subtracting $f(x^*)$ and dividing by λ on both sides, we have

$$0 \leq \frac{f(x^* + \lambda(x - x^*)) - f(x^*)}{\lambda}.$$

Taking the limit for $\lambda \to 0^+$ we get the directional derivative in the direction $x - x^*$ bounded as

$$0 \leq \lim_{\lambda \to 0^+} \frac{f(x^* + \lambda(x - x^*)) - f(x^*)}{\lambda} = \langle \nabla f(x^*), x - x^* \rangle \ ,$$

which shows that $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ for all $x \in X$. □

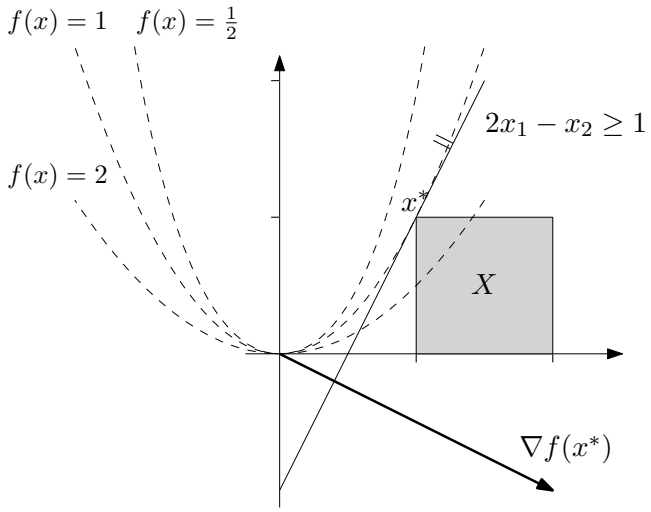
For a geometric interpretation of the previous theorem, let us consider the two following cases:

- If x^* is in the interior of X, the previous theorem implies that x^* is a global minimum if and only if $\nabla f(x^*) = 0$. Indeed, if x^* is in the interior, then for $\varepsilon > 0$ sufficiently small the point $x = x^* - \varepsilon \nabla f(x^*)$ is also in X, thus $\langle \nabla f(x^*), x - x^* \rangle = -\varepsilon \langle \nabla f(x^*), \nabla f(x^*) \rangle = -\varepsilon \|\nabla f(x^*)\|^2 \leq 0$ and so the inequality $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ implies $\nabla f(x^*) = 0$.
- If x^* is a point on the boundary of X and $\nabla f(x^*) \neq 0$, then the previous theorem states that x^* is a minimizer in X if and only if X is contained in the half-space $\{x \,|\, \langle \nabla f(x^*), x \rangle \geq \langle \nabla f(x^*), x^* \rangle\}$. This means that, all directions pointing towards X starting from x^* on the boundary are directions of ascent.

Example 2.11. Let us consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined as $f(x) = \frac{x_1^2}{x_2}$ over $\mathbf{dom}\, f = \{x \in \mathbb{R}^2 \,|\, x_2 > 0\}$. (In Example 2.13 we verify that f is convex.) Let $X = \{x \in \mathbb{R}^2 \,|\, 1 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 1\}$. We will show that $x^* = \binom{1}{1}$ minimizes f in X. The gradient of f is

$$\nabla f(x) = \begin{pmatrix} 2x_1/x_2 \\ -x_1^2/x_2^2 \end{pmatrix},$$

thus $\nabla f(x^*) = \binom{2}{-1}$. According to Theorem 2.10, x^* is a minimizer if and only if $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ for all $x \in X$. Noting that $\langle \nabla f(x^*), x^* \rangle = 1$, we need to verify that X is contained in the half-space $\{x \in \mathbb{R}^2 \,|\, (2, -1)x \geq 1\}$. This is indeed the case, since $x_1 \geq 2$ and $x_2 \leq 1$ imply $2x_1 - x_2 \geq 1$. This is shown in the figure below.



Concave functions A function f is *concave* if $-f$ is convex. Observe that Theorems 2.8 and 2.10 remain true if we replace "convex" with "concave" and "minimum" with "maximum". We will also extend concave functions f to the value range $\mathbb{R} \cup \{-\infty\}$ with $f(x) = -\infty$ for all $x \notin \mathbf{dom}\, f$.

2.1.5 Second order characterisation of convexity

We now present a characterisation of twice differentiable convex functions. Recall the second order characterisation of critical points from Theorem 1.18. As an immediate use of this characterisation, we will be able to decide whether a twice differentiable function is convex based on the Hessian.

Theorem 2.12. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable.*

(i) If $\nabla^2 f(z)$ is positive semidefinite for every $z \in \mathbf{dom}\, f$, then f is convex.

(ii) Assume that f is convex, and $\nabla^2 f$ is continuous on $\mathbf{dom}\, f$. Then, $\nabla^2 f(z)$ is positive semidefinite for every $z \in \mathbf{dom}\, f$.

Proof. We only prove part *(i)*. This is immediate from Theorem 1.16 and Theorem 2.6. Let us select any $x, y \in \mathbf{dom}\, f$, and consider the Taylor expansion; pick $\bar{x} \in [x, y]$ as in Theorem 1.16. By assumption, $\nabla^2 f(\bar{x})$ is positive semidefinite, and therefore

$$D_f(x, y) = f(x) - f(y) - \nabla f(y)^\top (x - y) = \frac{1}{2}(x - y)^\top \nabla^2 f(\bar{x})(x - y) \geq 0.$$

□

Example 2.13. Consider the function as in Example 2.11, that is, $f(x) = \dfrac{x_1^2}{x_2}$, where $\mathbf{dom}\, f = \{x \in \mathbb{R}^2 \,|\, x_2 > 0\}$. Then,

$$\nabla^2 f(x) = \begin{pmatrix} \frac{2}{x_2} & -\frac{2x_1}{x_2^2} \\ -\frac{2x_1}{x_2^2} & \frac{2x_1^2}{x_2^3} \end{pmatrix} = \frac{2}{x_2^3} \begin{pmatrix} x_2^2 & -x_1 x_2 \\ -x_1 x_2 & x_1^2 \end{pmatrix}.$$

Pick any vector $v \in \mathbb{R}^2$. Then,

$$v^\top \nabla^2 f(x) v = \frac{2}{x_2^3} \cdot (v_1, v_2)^\top \begin{pmatrix} x_2^2 & -x_1 x_2 \\ -x_1 x_2 & x_1^2 \end{pmatrix} (v_1, v_2) = \frac{2}{x_2^3} \cdot (v_1 x_2 - v_2 x_1)^2 \geq 0,$$

using that $x_2 > 0$. We have thus shown that $f(x)$ is convex.

2.1.6 Convex quadratic functions

By a *quadratic function* $f : \mathbb{R}^n \to \mathbb{R}$ we mean a degree two polynomial function. For example, $f(x_1, x_2) = -x_1^2 + 3x_1x_2 + 2x_2^2 - 5x_1 + 6x_2 + 3$. We can write every quadratic function in the form

$$f(x) = x^\top Qx + \langle p, x\rangle + r\,,$$

where $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $p \in \mathbb{R}^n$, $r \in \mathbb{R}$. For the particular example above, the representation is $Q = \begin{pmatrix} -1 & 1.5 \\ 1.5 & 3 \end{pmatrix}$, $p = \begin{pmatrix} -5 \\ 6 \end{pmatrix}$, $r = 3$.

Theorem 2.14. *Let $Q \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $p \in \mathbb{R}^n$, and $r \in \mathbb{R}$. The quadratic function $f(x) = x^\top Qx + \langle p, x\rangle + r$ is convex if and only if Q is positive semidefinite.*

Proof. The function is twice continuously differentiable, and it is easy to see that $\nabla f^2(x) = 2Q$. The claim follows using Theorem 2.12. □

In the special case $n = 1$, we obtain the well-know fact that $f(x) = ax^2 + bx + c$ is convex if and only if $a \geq 0$.

2.2 Convex optimization problems

The general form of a mathematical optimization problem is

$$\begin{aligned} \inf \quad & f_0(x) \\ & f_i(x) \leq 0\,, \quad i = 1, \ldots, m\,, \\ & h_i(x) = 0\,, \quad i = 1, \ldots, k\,. \end{aligned} \tag{2.2}$$

The *domain* of problem (2.2) is the set of points for which the objective function and the constraints functions are defined, that is

$$\mathcal{D} = \left(\bigcap_{i=0}^{m} \mathbf{dom}\, f_i\right) \bigcap \left(\bigcap_{i=1}^{k} \mathbf{dom}\, h_i\right).$$

The *feasible region* is the set X of all points in $\mathcal{D}$ satisfying the constraints. If $\mathcal{D} = \mathbb{R}^n$ and $m = k = 0$, then the problem is called an *unconstrained* optimization problem.

We say that the above problem is a *convex optimization problem* if $f_0, \ldots, f_m$ are convex functions, and $h_1, \ldots, h_k$ are affine functions; that is, there exist $a_1, \ldots, a_k \in \mathbb{R}^m$ and $b_1, \ldots, b_k \in \mathbb{R}$ such that $h_i(x) = \langle a_i, x\rangle - b_i$ $(i = 1, \ldots, k)$. The equality constraints can therefore be expressed as $\langle a_i, x\rangle = b_i$.

Note that the requirement that $h_1, \ldots, h_k$ are affine is needed in order for the definition to be consistent. Indeed, if we replaced each equality constraint $h_i(x) = 0$ with the two inequality constraints $h_i(x) \leq 0$, $-h_i(x) \leq 0$, then a convex problem should satisfy that both h_i and $-h_i$ are convex, that is, h_i needs to be both concave and convex. The only functions that are both concave and convex are the affine ones, therefore we need to require that h_i are affine.

Note that, since $f_1, \ldots, f_m$ are convex functions, the sets $\{x \,|\, f_i(x) \leq 0\}$ are convex sets (this is easy to show). The sets $\{x \,|\, \langle a_i, x\rangle = b_i\}$ are hyperplanes, and therefore convex.

These facts and Theorem 2.8 imply the following important facts.

Remark 2.15. *If problem* (2.2) *is convex, then*

1. *The feasible region X is convex, because it is the intersection of convex sets.*
2. *Every local optimum for f_0 in X is also a global optimum.*

On the existence of solutions Note that the objective in (2.2) is infimum instead of minimum. Even for convex optimization problems, it is possible that the infimum value exists but there is no optimal solution. As a simple example take $\inf 1/x$, over the domain $x > 0$.

Even in cases where an optimal solution exists to a convex program, it may not be a rational number, already for e.g. convex cubic objectives. For both of these reasons, we usually aim to find an *approximately optimal solution*

Definition 2.16 (Approximately optimal solutions)**.** *Let p^* denote the optimum value in* (2.2)*. We say that $x \in X$ is an* ε-approximately optimal solution *or a* ε-approximate solution*, if*

$$f(x) \leq p^* + \varepsilon.$$

Concave maximization Convex optimization problems have been defined as minimization problems. However, if in a maximization problem of the form

$$\begin{aligned} \sup \quad & f_0(x) \\ & f_i(x) \leq 0\,, \quad i = 1, \ldots, m\,, \\ & h_i(x) = 0\,, \quad i = 1, \ldots, k\,. \end{aligned}$$

the objective function f_0 is concave, while $f_1, \ldots, f_m$ are convex and $h_1, \ldots, h_k$ affine, we will also say that the problem is a convex optimization problem. This is justified by the fact that the equivalent minimization problem obtained by replacing "$\sup f_0(x)$" with "$\inf -f_0(x)$" is a convex optimization problem, because $-f_0$ is convex.

2.3 Convexity in regression problems

We now discuss some basic statistical models from a convex optimisation perspective. We do not cover in detail the statistical context and applications; see the textbook [4] for such details.

2.3.1 Linear regression

Linear regression is one of the most fundamental models in statistics and machine learning. We are given a dataset of m points; each data point has n *features* or *predictor variables* described by real numbers. Thus, the j-th data point can be described as a vector $(a_{j1}, a_{j2}, \ldots, a_{jn}) \in \mathbb{R}^n$. Further, we have a *dependent variable* or *target variable* b_j; the goal is to predict this value from the feature values. We assume a linear dependence of the form

$$b_i = \beta_0 + \beta_1 a_{j1} + \beta_2 a_{j2} + \ldots + \beta_n a_{jn} + \varepsilon_i\,,$$

where $\beta = (\beta_0, \beta_1, \ldots, \beta_n) \in \mathbb{R}^{n+1}$ is an (unknown) vector of coefficients, including a *bias* term β_0. Each ε_i is a random error variable, that all come from the same normal distribution and are independent from each other.

The input for the linear regression problem comprises a dataset on m feature vectors $(a_{j1}, \ldots, a_{jn}) \in \mathbb{R}^n$ and dependent variables $b_j \in \mathbb{R}$, $j = 1, 2, \ldots, m$. The goal is to estimate the unknown coefficients $\beta \in \mathbb{R}^n$ that describe the linear dependence. For simplicity of notation, we expand the feature vectors to $n + 1$ dimensions including a bias term $a_{j0} = 1$; we use $a_j = (1, a_{j1}, \ldots, a_{jn}) \in \mathbb{R}^{n+1}$ to denote this extended feature vector.

Given an estimate vector $\hat{\beta} \in \mathbb{R}^n$, the *residual* of the i-th data point is the absolute value of the difference between the predicted value and the target variable,

$$\left|\hat{\beta}_0 + \hat{\beta}_1 a_{j1} + \ldots + \hat{\beta}_n a_{jn} - b_j\right| = \left|\left\langle a_j, \hat{\beta}\right\rangle - b_j\right|\,.$$

The *least squares estimate* finds the estimate $\hat{\beta} \in \mathbb{R}^n$ that minimizes the squared sum of the residuals, that is, $\hat{\beta}$ is the optimal solution

$$\min_{\beta \in \mathbb{R}^{n+1}} \sum_{j=1}^{m} \left(\langle a_j, \beta\rangle - b_j\right)^2\,. \tag{2.3}$$

Problem (2.3) turns out to be an *unconstrained convex quadratic optimization problem*. Let $A \in \mathbb{R}^{m\times(n+1)}$ be the matrix with entries a_{ji}, $j = 1,\dots,m$, $i = 0,\dots,n$, and $b \in \mathbb{R}^m$ the vector of dependent variables. Thus, the i'th row of A is the vector $a_i^\top$. We can rewrite the objective in the form

$$\begin{aligned}\sum_{j=1}^{m} \beta^\top a_j^\top a_j \beta - 2b_j \langle a_j, \beta\rangle + b_j^2 = &\quad \beta^\top \left(\sum_{j=1}^{m} a_j a_j^\top\right)\beta - 2\left(\sum_{j=1}^{m} b_j a_j\right)^\top \beta + \sum_{j=1}^{m} b_j^2 \\ = &\quad \beta^\top \left(A^\top A\right)\beta - 2\left(A^\top b\right)^\top \beta + \sum_{j=1}^{m} b_j^2 \,.\end{aligned} \tag{2.4}$$

This is a quadratic objective function. By Theorem 2.14, convexity follows by showing that the matrix $Q = A^\top A$ is positive semidefinite, which is an easy exercise to check.

Orthogonal projection viewpoint A geometric interpretation can be given as follows. For any $\beta \in \mathbb{R}^{n+1}$, the predicted values are given by the vector $A\beta \in \mathbb{R}^m$; taken over all $\beta \in \mathbb{R}^{n+1}$, these form a linear subspace $\mathcal{L} \subseteq \mathbb{R}^m$. The objective function in (2.3) can be written as the squared distance of the vectors $A\beta$ and b:

$$\|A\beta - b\|^2 \,,$$

which is exactly the nearest point to b in $\mathcal{L}$; in other words, the orthogonal projection $\Pi_{\mathcal{L}}(b)$.

Explicit solution The convex optimization problem (2.3) is very simple: in contrast to most problems we will encounter in this course, an optimal solution can be explicitly computed using basic matrix algebra. From the above viewpoint, this amounts to computing the projection matrix $\Pi_{\mathcal{L}}$.

We can use Theorem 2.9 asserting that β is an optimal solution to minimizing $f(\beta)$ over $\beta \in \mathbb{R}^{n+1}$ if and only if $\nabla f(\beta) = 0$. Using (2.4), the gradient of the squared loss objective function in (2.3) can be written as

$$2A^\top A\beta - 2A^\top b \,,$$

hence, the optimal β has to be the solution to the system of linear equations

$$A^\top A\beta = A^\top b \,.$$

Let us first assume that the columns of the matrix A are linearly independent: there is no predictor variable that can be written as the exact linear combination of other predictor variables (including the bias). Under this assumption, the matrix $A^\top A \in \mathbb{R}^{(n+1)\times(n+1)}$ is positive definite and consequently invertible (check!). Then, one can write the optimal solution to (2.3) as

$$\hat{\beta} = \left(A^\top A\right)^{-1} A^\top b \,. \tag{2.5}$$

Without the independence assumption, $A^\top A$ may not be invertible, and in fact the minimum-norm solution is not unique: we can have $A\beta = A\beta'$ for $\beta \neq \beta'$. The solution is then chosen as $\hat{\beta} = \left(A^\top A\right)^\dagger Ab$, where $Q^\dagger$ denotes the *pseudoinverse* of the matrix Q; we do not discuss this here.

2.3.2 Regularisation: Lasso and Ridge regression

Least squares regression finds the model $\hat{\beta}$ that gives the best fit for the given dataset on m points. However, the usual goal in statistical learning is not finding the best model for a given set, but building *predictive models*: given a data sample called the *training set*, we aim to find a model that gives the best prediction on yet unseen data points. For this reason, optimizing for the training set may lead to *overfitting*.

Ridge and *Lasso* regression are two *regularised* variants of the standard least squares regression model that can give better predictions. The input is the same: a set of m data points (a_j, b_j), where

$a_j \in \mathbb{R}^{n+1}$, $b_j \in \mathbb{R}$, and $a_{j0} = 1$ corresponds to the bias term. The model uses a parameter $t > 0$

$$\begin{aligned} &\min \sum_{j=1}^{m} (\langle a_j, \beta \rangle - b_j)^2 \\ &\sum_{i=1}^{n} \beta_i^2 \leq t\,. \end{aligned} \tag{2.6}$$

That is, we add a bound on the ℓ_2-norm of the vector $(\beta_1, \ldots, \beta_n)$; the limit $t = \infty$ corresponds to ordinary least-squares regression. This limits the norm of the coefficients of the predictor variables; note that the bias term β_0 is *not* included. Hence, we obtain a more robust prediction; see [4, Section 6.2] for more statistical background and applications. The book defines the problem in a different form; this will be discussed in the next chapter.

From our perspective, (2.6) adds a convex constraint to (2.3); this follows by the convexity of the function $\sum_{i=1}^{n} \beta_i^2$. Hence, we obtain a constrained convex optimization problem.

More generally, every p-norm for $p \geq 1$ is a convex function. *Lasso regression* (least absolute shrinkage and selection operator) is the same as Ridge regression, but replaces the ℓ_2-norm by ℓ_1-norm:

$$\begin{aligned} &\min \sum_{j=1}^{m} (\langle a_j, \beta \rangle - b_j)^2 \\ &\sum_{i=1}^{n} |\beta_i| \leq t\,. \end{aligned} \tag{2.7}$$

Despite the similar form, these can result in considerably different β solutions. An advantage of Lasso is that it will typically use only a subset of the coefficients while setting the others to zero; in contrast, Ridge regression typically uses all of them. Hence, Lasso can also be interpreted as a subset selection method that identifies a relevant subset of the coefficients. To understand why this happens, let us visualise the optimisation problems as in Figure 2.3. The red lines show the level sets of the objective function. In case of Lasso, we are likely to hit the feasible polytope at a 'corner' (in general, a lower dimensional face); this corresponds to setting some variables to 0.

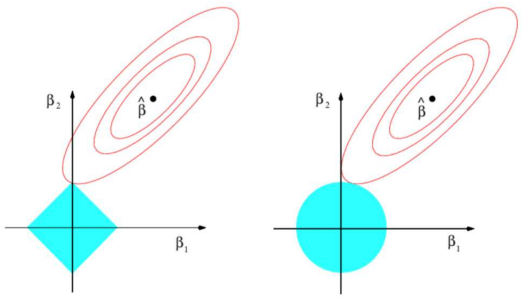


Figure 2.3: Lasso (left) and Ridge (right) regression (source: [4, Figure 6.7]).

2.3.3 Logistic regression

The above regression models assumed that the target variable is a real value. This is not the case for *classification* problems, where the target variable is in a discrete set of possible outcomes. The simplest case is *binary classification* with target variable $b_j \in \{0, 1\}$; these can be interpreted as no/yes (negative/positive, etc.). Similarly to linear regression, we assume that the predictor variables $(a_{j1}, \ldots, a_{jn}) \in \mathbb{R}^n$ are continuous. We could apply linear regression for the data points (a_j, b_j), but the predicted outcome for a new data point could be any real number, e.g. negative or greater than 1.

Logistic regression is a standard model that outputs values between 0 and 1; this can be interpreted as a probabilistic outcome. Similarly to linear regression, the model is defined by an $n+1$ dimensional vector $(\beta_0, \beta_1, \ldots, \beta_n)$, and we again extend the data points by a coordinate $a_{j0} = 1$ to account for the bias term; we let $a_j = (a_{j0}, a_{j1}, \ldots, a_{jn}) \in \mathbb{R}^{n+1}$. We use *logistic function* to map the linear outcome $\langle a_j, \beta \rangle$ to a value in $[0, 1]$:

$$p_\beta(a_j) = \frac{1}{1 + e^{-\langle a_j, \beta \rangle}} .$$

Logistic regression can be derived as a *maximum likelihood estimation*. Given the feature vectors a_j, $j = 1, \ldots, m$, assume that each target value $b_j \in \{0, 1\}$ is sampled according to the probabilies $p_\beta(a_j)$ for a given coefficients $\beta \in \mathbb{R}^{n+1}$. That is, with probability $p_\beta(a_j)$, we set $b'_j = 1$, and with probability $1 - p_\beta(a_j)$, we set $b'_j = 0$. Then, the probability of getting the true target values $b_j = b'_j$ is given by the *likelihood function*:

$$\ell(\beta) = \prod_{j : b_j = 1} p_\beta(a_j) \prod_{j : b_j = 0} (1 - p_\beta(a_j)) .$$

Logistic regression selects $\hat{\beta}$ as the optimal solution to

$$\hat{\beta} = \max_{\beta \in \mathbb{R}^{n+1}} \ell(\beta) . \tag{2.8}$$

Since the log function is monotone on $\mathbb{R}_{>0}$, this is equivalent to maximising $\log \ell(\beta)$ that is further equivalent to minimising $-\log \ell(\beta)$ that can be written as

$$-\log \ell(\beta) = \sum_{j : b_j = 1} \log \left(1 + e^{-\langle a_j, \beta \rangle}\right) + \sum_{j : b_j = 0} \log \left(1 + e^{\langle a_j, \beta \rangle}\right) .$$

In the exercises, we will show that this is a convex function. Hence, logistic regression can also be seen as an unconstrained convex optimisation problem. However, in contrast to linear regression, there is no simple closed form for the optimal solution, and one needs to invoke convex optimisation algorithm to find the (approximately) optimal coefficients.

Chapter 3

Lagrangian duality

3.1 The Lagrangian dual

Lagrangian The *Lagrangian* of problem (2.2) is the function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^k \to \mathbb{R}$ defined by

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{k} \nu_i h_i(x)\,, \tag{3.1}$$

where $\mathbf{dom}\, L = \mathcal{D} \times \mathbb{R}^m_+ \times \mathbb{R}^k$. Here λ and ν are vectors of variables in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. Variable λ_i is the *Lagrange multiplier* of constraint $f_i(x) \leq 0$ and are required to be nonnegative, whereas ν_i is the Lagrange multiplier of constraint $h_i(x) = 0$.

Lemma 3.1. *For every feasible point $\bar{x}$ for (2.2) and every $(\lambda, \nu) \in \mathbb{R}^m \times \mathbb{R}^k$, $\lambda \geq 0$, we have that*

$$L(\bar{x}, \lambda, \nu) \leq f_0(\bar{x}) \tag{3.2}$$

Proof. Since $\bar{x}$ is feasible, it follows that $f_i(\bar{x}) \leq 0$ and $h_i(x) = 0$. Therefore

$$L(\bar{x}, \lambda, \nu) = f_0(\bar{x}) + \sum_{i=1}^{m} \lambda_i f_i(\bar{x}) + \sum_{i=1}^{k} \nu_i h_i(\bar{x}) \leq f_0(\bar{x})\,,$$

because $\lambda_i \geq 0$ and thus $\lambda_i f_i(x) \leq 0$ for $i = 1, \ldots, m$. □

Lagrange dual function We define the *Lagrange dual function* $g : \mathbb{R}^m \times \mathbb{R}^k \to \mathbb{R}$ as

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu)\,. \tag{3.3}$$

Recall that $X \subseteq \mathcal{D}$ is the feasible region of the problem. If we denote by p^* the optimal value of problem (2.2)—that is, $p^* = \inf_{x \in X} f_0(x)$—it follows from (3.2) that, for all $(\lambda, \nu) \in \mathbb{R}^m \times \mathbb{R}^k$, $\lambda \geq 0$,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq \inf_{x \in X} L(x, \lambda, \nu) \leq \inf_{x \in X} f_0(x) = p^*\,. \tag{3.4}$$

That is, for every choice of $(\lambda, \nu) \in \mathbb{R}^m \times \mathbb{R}^k$, $\lambda \geq 0$, the value $g(\lambda, \nu)$ is a lower-bound on the optimal value p^*.

Next we point out an interesting property of the Lagrange dual function g.

Lemma 3.2. *The Lagrange dual function g is concave.*

Proof. For every $x \in \mathcal{D}$, the function $\theta_x : \mathbb{R}^m_+ \times \mathbb{R}^k \to \mathbb{R}$ defined by $\theta_x(\lambda, \nu) = L(x, \lambda, \nu)$ is affine, and therefore concave. By definition,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} \theta_x(\lambda, \nu),$$

therefore g is the pointwise infimum of the family of concave functions $\{\theta_x \,|\, x \in \mathcal{D}\}$, and thus concave. This follows since Proposition 2.5 is applicable to $-g$, showing that a function defined as the pointwise supremum of convex functions is convex. Consequently, g is concave. □

Lagrangian dual problem As we have seen, for every $\lambda \geq 0$ and $\nu \in \mathbb{R}^k$, the value $g(\lambda, \nu)$ provides a lower bound to the optimal value of (2.2). The *Lagrangian dual* is the problem of finding the best such lower bound. That is,

$$\begin{array}{ll} \sup & g(\lambda, \nu) \\ \text{s.t.} & \lambda \geq 0 \\ & (\lambda, \nu) \in \mathbf{dom}\, g \end{array} \tag{3.5}$$

By Lemma 3.2, the above problem is a convex optimization problem (because the constraints are linear and the objective is to maximize a concave function). Note that this is the case even when the primal problem (2.2) is not convex!

The following is an immediate consequence of (3.4).

Theorem 3.3 (Week Lagrangian Duality)**.** *Let p^* be the optimal value of the primal problem* (2.2)*, and let d^* be the optimal value of the dual problem* (3.5)*. Then*

$$d^* \leq p^*.$$

We call $p^* - d^*$ the *duality gap* of problem (2.2). We discuss below that strong duality always holds for linear programming (LP) problems, but not in general for convex problems.

Example 3.4. Consider the problem

$$\begin{array}{ll} \min & x^2 + 1 \\ & (x-2)(x-4) \leq 0 \end{array}$$

Note that it is a convex optimization problem, since both functions $x \mapsto x^2+1$ and $x \mapsto (x-2)(x-4)$ are convex quadratic functions. Since both functions are defined over all of $\mathbb{R}$, the domain of the problem is $\mathcal{D} = \mathbb{R}$.

The feasible region is the interval $[2, 4]$, and the minimum is attained at $x^* = 2$, with optimal objective value $p^* = 5$.

The Lagrangian of the above problem is the function of two variables

$$\begin{aligned} L(x, \lambda) &= x^2 + 1 + \lambda(x-2)(x-4) \\ &= (1+\lambda)x^2 - 6\lambda x + 8\lambda + 1. \end{aligned}$$

To compute the Lagrangian dual function, we need to compute, for all $\lambda \in \mathbb{R}$,

$$g(\lambda) = \inf_{x \in \mathbb{R}} L(x, \lambda) = \inf_{x \in \mathbb{R}} (1+\lambda)x^2 - 6\lambda x + 8\lambda + 1.$$

Observe that for $\lambda \leq -1$ the above infimum is $-\infty$. For $\lambda > -1$, the function $(1+\lambda)x^2 - 6\lambda x + 8\lambda + 1$ is a convex quadratic function, therefore its global minima in $\mathbb{R}$ are the points with zero derivative. We compute the derivative of $L(x, \lambda)$ with respect to x and set it to zero.

$$\frac{\partial L(x, \lambda)}{\partial x} = 2(1+\lambda)x - 6\lambda = 0.$$

The zero of the above equation is the point

$$\bar{x} = \frac{3\lambda}{1+\lambda},$$

thus, for all $\lambda > -1$,

$$\begin{aligned} g(\lambda) &= L(\bar{x}, \lambda) = (1+\lambda)\left(\frac{3\lambda}{1+\lambda}\right)^2 - 6\lambda\frac{3\lambda}{1+\lambda} + 8\lambda + 1 \\ &= \frac{-\lambda^2 + 9\lambda + 1}{1+\lambda}, \end{aligned}$$

and $\mathbf{dom}\,(g) = \{\lambda \in \mathbb{R} \,|\, \lambda > -1\}$. The Lagrangian dual is therefore

$$\begin{array}{ll} \max & \dfrac{-\lambda^2 + 9\lambda + 1}{1+\lambda} \\ \text{s.t.} & \lambda \geq 0 \end{array}$$

To solve the above, we compute the derivative of g and set it to zero

$$\begin{aligned} g'(\lambda) &= \frac{(-2\lambda+9)(1+\lambda) - (-\lambda^2+9\lambda+1)}{(1+\lambda)^2} \\ &= -\frac{\lambda^2+2\lambda-8}{(1+\lambda)^2} = 0 \end{aligned}$$

The only non-negative solution to the above equation is $\lambda^* = 2$, which is therefore the dual optimal solution. The optimal value of the dual is therefore $d^* = g(2) = 5$. Note that in this case $p^* = d^*$, thus strong duality holds.

Example 3.5. Ridge regression Consider the Ridge regression problem in (2.6). The input is a set of m data points (a_j, b_j), where $a_j \in \mathbb{R}^{n+1}$, $b_j \in \mathbb{R}$, and $a_{j0} = 1$; we let $A \in \mathbb{R}^{m\times(n+1)}$ denote the data matrix and $b \in \mathbb{R}^m$ the vector of target variables. We rewrite the norm constraint $\sum_{i=1}^n \beta_i^2 \leq t$ as $\sum_{i=1}^n \beta_i^2 - t \leq 0$ to get the desired form. Let $J \in \mathbb{R}^{(n+1)\times(n+1)}$ denote the diagonal matrix with $J_{00} = 0$ and $J_{ii} = 1$ for $i = 1, 2, \ldots, n$ (that is, we replace the top left entry of the identity matrix by 0). Then, we can write $\sum_{i=1}^n \beta_i^2 = \beta^\top J \beta$.

The Lagrangian is

$$L(\beta,\lambda) = \sum_{j=1}^m (\langle a_j, \beta\rangle - b_j)^2 + \lambda\left(\sum_{i=1}^n \beta_i^2 - t\right) = \beta^\top\left(A^\top A + \lambda J\right)\beta - 2\left(A^\top b\right)^\top \beta + \sum_{j=1}^m b_j^2 - \lambda t\,. \quad (3.6)$$

Let us compute $g(\lambda) := \inf_{\beta\in\mathbb{R}^{n+1}} L(\beta,\lambda)$ for $\lambda \geq 0$. For fixed nonnegative $\lambda \geq 0$, $L(\beta,\lambda)$ is a convex function in β. The minimum is taken where the gradient (with respect to the β variables) is 0, that is,

$$2\left(A^\top A + \lambda J\right)\beta - 2A^\top b = 0\,,$$

For $\lambda > 0$, the matrix $A^\top A + \lambda J$ is positive definite[1] and thus invertible; hence, the optimal β vector is

$$\beta = \left(A^\top A + \lambda J\right)^{-1} A^\top b\,. \quad (3.7)$$

Thus, we can substitute

$$g(\lambda) = -b^\top A\left(A^\top A + \lambda J\right)^{-1} A^\top b - \lambda t + \sum_{j=1}^m b_j^2\,.$$

We do not solve the Lagrangian problem here. However, let us note that Ridge regression is commonly defined in the Lagrangian form: instead of fixing the parameter $t > 0$, a parameter $\lambda > 0$ is selected, and Ridge regression is defined as minimizing $L(\beta,\lambda)$ for this fixed λ, with the closed form formula (3.7).

We also note that there is a (nonlinear) 1-to-1 mapping between the parameters $t \in \mathbb{R}_+ \cup \{\infty\}$ and the corresponding optimal $\lambda \in \mathbb{R}_+ \cup \{\infty\}$ values. In particular, $\lambda \to 0$ corresponds to $t \to \infty$, and $\lambda \to \infty$ corresponds to $t \to 0$.

Example 3.6. (Convex problem with strict duality gap) Consider the problem

$$\begin{array}{lll} p^* = \min & e^{-x_1} \\ \text{s.t.} & \dfrac{x_1^2}{x_2} \leq 0 \end{array}$$

[1] this uses the fact that the first column of A is the all ones vector

defined over $\mathcal{D} = \{x \in \mathbb{R}^2 \,|\, x_2 > 0\}$. One can verify that this is a convex optimization problem. Observe that the feasible region of the problem is the set $X = \{x \in \mathbb{R}^2 \,|\, x_1 = 0,\, x_2 > 0\}$. In particular, every feasible solution has objective value 1, therefore $p^* = 1$.

Let us now compute the Lagrangian dual. The Lagrangian is $L(x, \lambda) = e^{-x_1} + \lambda \frac{x_1^2}{x_2}$. We need to compute $g(\lambda) := \inf_{x \in \mathcal{D}} L(x, \lambda)$ for $\lambda \geq 0$. Observe that $L(x, \lambda) \geq 0$ for all $x \in \mathcal{D}$ and all $\lambda \geq 0$, therefore $g(\lambda) \geq 0$. On the other hand, there are $x \in \mathcal{D}$ for which $L(x, \lambda)$ takes arbitrarily small value (it suffices to consider any sequence of points in $\mathcal{D}$ for which $x_1 \to +\infty$ and $\frac{x_1^2}{x_2} \to 0$). If follows that $g(\lambda) = 0$ for all $\lambda \geq 0$, and therefore $d^* = 0$. Thus the duality gap is $p^* - d^* = 1 - 0 = 1$.

3.1.1 Duality for linear programming

Consider an LP problem of the form

$$\begin{array}{ll} \min & \langle c, x \rangle \\ \text{s.t.} & Ax \geq b \end{array} \tag{3.8}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Rewriting the constraints in the form

$$b - Ax \leq 0,$$

the Lagrangian of the above problem is

$$L(x, \lambda) = \langle c, x \rangle + \langle \lambda, b - Ax \rangle = \langle b, \lambda \rangle + \left\langle c - A^\top \lambda, x \right\rangle ,$$

where λ is a vector of m variables. Observe that $L(x, \lambda)$ is an affine function in x, therefore it is either a constant function or can go to $-\infty$. Note that $L(x, \lambda)$ is constant if and only if $A^\top \lambda - c = 0$, therefore the Lagrangian dual function is

$$g(\lambda) = \inf_{x \in \mathbb{R}^n} L(x, \lambda) = \begin{cases} \langle b, \lambda \rangle & \text{if} A^\top \lambda = c \\ -\infty & \text{otherwise.} \end{cases}$$

It follows that the Lagrangian dual function is given by $g(\lambda) = \langle b, \lambda \rangle$, and it is defined over $\mathbf{dom}\, g = \{\lambda \in \mathbb{R}^m \,|\, A^\top \lambda = c\}$. The Lagrangian dual of the LP function is therefore

$$\begin{array}{ll} \max & \langle b, \lambda \rangle \\ \text{s.t.} & A^\top \lambda = c \\ & \lambda \geq 0. \end{array} \tag{3.9}$$

This is the usual LP dual. Recall the strong duality theorem:

Theorem 3.7 (Strong duality theorem of linear programming)**.** *If both programs* (3.8) *and* (3.9) *are feasible, then their optimum values are equal. The program* (3.8) *is unbounded if and only if* (3.9) *is infeasible, and conversely,* (3.9) *is unbounded if and only if* (3.8) *is infeasible.*

3.1.2 Slater's condition

Despite cases in which convex optimization problems have positive duality gaps, such as the one in Example 3.6, strong duality holds for convex optimization problems under fairly general conditions.

Definition 3.8 (Slater's condition)**.** *We say that a convex optimization problem* (2.2) *satisfies Slater's condition if there exists a feasible solution $\bar{x}$ in the interior of $\mathcal{D}$ such that $f_i(\bar{x}) < 0$ for $i = 1, \ldots, m$.*

Theorem 3.9 (Strong duality under Slater's condition)**.** *Strong duality holds for every convex optimization problem satisfying Slater condition. Furthermore, in this case the dual has an optimal solution.*

We do not present the proof of the above theorem here. Note that the Ridge regression satisfies Slater's condition, hence, strong duality follows. In contrast, the problem in Example 3.6 does not satisfy Slater's condition, because every feasible solution satisfies the only constraint of the problem to equality.

3.2 Karush-Kuhn-Tucker conditions

Consider a general problem (not necessarily convex) of the form (2.2), and assume that the functions $f_0, f_1, \ldots, f_m$, $h_1, \ldots, h_k$ are differentiable.

Suppose that strong duality holds, and that both the primal and the dual problem admit an optimal solution, say x^* for the primal and (λ^*, ν^*) for the dual problem. It follows that

$$\begin{aligned} f_0(x^*) &= g(\lambda^*, \nu^*) \\ &= \inf_{x \in \mathcal{D}} L(x, \lambda^*, \mu^*) \\ &\leq L(x^*, \lambda^*, \mu^*) \qquad (*) \\ &= f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{k} \nu_i^* h_i(x^*) \\ &\leq f_0(x^*)\,. \qquad (**) \end{aligned}$$

This shows that equality must holds throughout in the above chain of inequalities. In particular, inequality (*) implies

$$\inf_{x \in \mathcal{D}} L(x, \lambda^*, \mu^*) = L(x^*, \lambda^*, \mu^*).$$

That is, x^* must be a global minimum for $L(x, \lambda^*, \mu^*)$. Since $f_0, f_1, \ldots, f_m$ are differentiable, it follows that $L(x, \lambda^*, \mu^*)$ is differentiable. By Theorem 1.17, it follows that the gradient of $L(x, \lambda^*, \mu^*)$ computed at x^* must be the zero vector. That is,

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{k} \nu_i^* \nabla h_i(x^*) = 0. \tag{3.11}$$

Inequality (**) must also be satisfied at equality. Observe that $h_i(x^*) = 0$, because x^* is feasible, and $\lambda_i^* f_i(x^*) \leq 0$ because x^* is feasible and $\lambda^* \geq 0$. Hence,

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, \ldots, m. \tag{3.12}$$

Equations (3.12) are called *complementary slackness conditions*. They state that, if an optimal primal solution satisfies the ith constraint as strict inequality, then the corresponding dual variable λ_i should be zero in an optimal dual solution.

We summarize the above discussion in the following statement.

Lemma 3.10. *Let $f_0, f_1, \ldots, f_m$, $h_1, \ldots, h_k : \mathbb{R}^n \to \mathbb{R}$ be differentiable functions. Assume that strong duality holds for the optimization problem* (2.2)*. If* (2.2) *admits an optimal solution x^* and its dual* (3.5) *admit an optimal solution (λ^*, ν^*), then these must satisfy the following conditions:*

$$\begin{aligned} f_i(x^*) &\leq 0 \quad (i = 1, \ldots, m) \\ h_i(x^*) &= 0 \quad (i = 1, \ldots, k) \\ \lambda_i^* &\geq 0 \quad (i = 1, \ldots, m) \\ \lambda_i^* f_i(x^*) &= 0 \quad (i = 1, \ldots, m) \\ \nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{k} \nu_i^* \nabla h_i(x^*) &= 0. \end{aligned} \tag{3.13}$$

These are known as the *Karush–Kuhn–Tucker (KKT) conditions*. Observe that the above result only says that the KKT conditions are necessarily satisfied by every pair of optimal primal/dual solutions (x^*, λ^*, ν^*) if strong duality holds. However, the next example illustrates that it is not true in general that for every solution (x^*, λ^*, ν^*) to the KKT system the point x^* is a primal optimum.

Example 3.11. Consider the (non-convex) optimization problem $\min\{x^3 \mid x^2 \leq 1\}$. Clearly, the only optimal solution is $x = -1$, with value -1. The Lagrangian is $L(x, \lambda) = x^3 + \lambda(x^2 - 1)$, thus the KKT

conditions are

$$\begin{aligned} x^2 - 1 &\leq 0 \\ \lambda &\geq 0 \\ \lambda(x^2-1) &= 0 \\ 3x^2 + 2\lambda x &= 0. \end{aligned}$$

The point $(x^*, \lambda^*) = (0, 0)$ is a solution for the KKT conditions, but the point $x^* = 0$ is not a primal optimum (it has value 0, whereas the optimal value is -1).

For convex optimization problems, however, the KKT conditions are also sufficient, as shown in the following theorem.

Theorem 3.12. *Let $f_0, f_1, \ldots, f_m$ be convex differentiable functions and $h_1, \ldots, h_k$ be affine functions. If the KKT conditions for problem (2.2) have a solution (x^*, λ^*, ν^*), then x^* is optimal for problem (2.2), (λ^*, ν^*) is optimal for its dual (3.5), and strong duality holds.*

Proof. Assume that (x^*, λ^*, ν^*) is a solution to the KKT conditions (3.13). In particular, x^* is feasible for (2.2) and $\lambda^* \geq 0$. Thus it suffices to show that $f_0(x^*) \leq g(\lambda^*, \nu^*)$.

We only need to show that $f_0(x^*) = g(\lambda^*, \nu^*)$. We have

$$\begin{aligned} f_0(x^*) &= f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{k} \nu_i^* h_i(x^*) \\ &= \inf_{x \in \mathcal{D}} f_0(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x) + \sum_{i=1}^{k} \nu_i^* h_i(x) \\ &= g(\lambda^*, \nu^*). \end{aligned}$$

The first equality follows from the fact that $h_i(x^*) = 0$ $(i = 1, \ldots, k)$ and $\lambda^* f_i(x^*) = 0$ $(i = 1, \ldots, m)$. The second equality follows from Theorem 2.10, since the function $L(x, \lambda^*, \nu^*)$ is convex in x (because $f_1, \ldots, f_m$, $h_1, \ldots, h_k$ are convex), and its gradient at x^* is zero (from condition (3.11)). □

Example 3.13. Consider the problem

$$\begin{array}{ll} \min & \frac{1}{2} x^\top P x + \langle q, x \rangle + r \\ s.t. & -1 \leq x_i \leq 1 \qquad i = 1, 2, 3 \end{array}$$

where

$$P = \begin{pmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{pmatrix}, \quad q = \begin{pmatrix} -22 \\ -29/2 \\ 13 \end{pmatrix}, \quad r = 1.$$

We will show that the point $x^* = (1, 1/2, -1)^\top$ is a global optimum.

The above is a convex optimization problem, because the constraint functions are affine, while the objective function is convex quadratic (indeed, matrix P is positive semidefinite). To show that x^* is optimal, we will find a dual solution that satisfies the KKT conditions together with x^*.

The constraints of the problems can be written as $-x_i - 1 \leq 0$ and $x_i - 1 \leq 0$, $i = 1, 2, 3$. We assign Lagrange multipliers λ_i^0 to constraint $-x_i - 1 \leq 0$, and λ_i^0 to constraint $x_i - 1 \leq 0$, $i = 1, 2, 3$. We denote by λ^0, λ^1 the corresponding vectors. The Lagrangian is

$$L(x, \lambda^0, \lambda^1) = \frac{1}{2} x^\top P x + q^\top x + r + \sum_{i=1}^{3} \lambda_i^0 (-x_i - 1) + \sum_{i=1}^{3} \lambda_i^1 (x_i - 1).$$

Recall that the gradient of the objective function f at any given point x is

$$\nabla f(x) = Px + q.$$

Therefore, the KKT conditions are

$$\begin{aligned} -x_i - 1 &\leq 0 \quad i = 1,2,3 \\ x_i - 1 &\leq 0 \quad i = 1,2,3 \\ \lambda^0, \lambda^1 &\geq 0 \\ \lambda_i^0(-x_i - 1) &= 0 \quad i = 1,2,3 \\ \lambda_i^1(x_i - 1) &= 0 \quad i = 1,2,3 \\ Px + q - \lambda^0 + \lambda^1 &= 0 \end{aligned}$$

Clearly $-1 \leq x_i^* \leq 1$. Furthermore, from the complementary slackness conditions we get

$$\begin{aligned} x_1^* > -1 &\Rightarrow \lambda_1^0 = 0 \\ -1 < x_2^* < 1 &\Rightarrow \lambda_2^0, \lambda_2^1 = 0 \\ x_3^* < 1 &\Rightarrow \lambda_3^1 = 0 \end{aligned}$$

Finally, substituting x^* into the last KKT equation and setting Lagrangian multiplier to zero as above, we obtain

$$Px^* + q - \lambda^0 + \lambda^1 = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \lambda_3^0 \end{pmatrix} + \begin{pmatrix} \lambda_1^1 \\ 0 \\ 0 \end{pmatrix} = 0.$$

The only solution to the above system is $\lambda_3^0 = 2 \geq 0$, $\lambda_1^1 = 1 \geq 0$. It follows that x^* is an optimal primal solution. An optimal dual solution is defined by $\lambda^0 = (0,0,2)^\top$ and $\lambda^1 = (1,0,0)^\top$.

Example 3.14. (KKT conditions for an LP problem.) Consider again an LP problem of the form

$$\begin{aligned} \min \quad & \langle c, x \rangle \\ \text{s.t.} \quad & Ax \geq b\,, \end{aligned}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. As before, the Lagrangian of the above problem is

$$L(x, \lambda) = \langle c, x \rangle + \langle \lambda, b - Ax \rangle = \langle b, \lambda \rangle + \left\langle c - A^\top \lambda, x \right\rangle .$$

Thus

$$\nabla L(x, \lambda) = c - A^\top \lambda\,.$$

It follows that the KKT conditions are

$$\begin{aligned} Ax &\geq b \\ \lambda &\geq 0 \\ (b - a_i^\top x)\lambda_i &= 0 \qquad i = 1, \ldots, m \\ A^\top \lambda &= c \end{aligned}$$

Thus the KKT conditions, when specialized to linear programming, enforce that x is a primal feasible solution, λ is a dual feasible solution, and that x and λ are in complementary slackness. These are the usual primal-dual slackness conditions for linear programming.

Chapter 4

Gradient descent

In this chapter, we start our journey on first order optimisation algorithms. We start with the simplest case, *unconstrained minimisation*, that is, for a convex function $f : \mathbb{R}^n \to \mathbb{R}$, we want to solve

$$\min_{x \in \mathbf{dom}\, f} f(x).$$

Througout, we will assume that an optimal solution x^* exists (recall that this may not always be the case). We let $p^* = f(x^*)$ denote the optimum value. If there are multiple optimal solutions, we let x^* denote an arbitrary one among them.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. According to Theorem 2.9, x^* is a global minimum point if and only if $\nabla f(x^*) = 0$. Recall from Taylor expansion that the gradient $\nabla f(x)$ provides a linear approximation of f around x. In particular, for a vector $x \in \mathbf{dom}\, f$, direction Δx, and step size $\eta > 0$,

$$f(x + \eta \Delta x) \approx f(x) + \eta \langle \nabla f(x), \Delta x \rangle \tag{4.1}$$

Since we wish to decrease the function value, we need to select a direction Δx such that $\langle \nabla f(x), \Delta x \rangle < 0$. A natural choice is $\Delta x = -\nabla f(x)$, the direction opposite to the gradient. The basic gradient descent method is as follows.

GRADIENT DESCENT
Input: A convex function $f : \mathbb{R}^n \to \mathbb{R}$, a starting point $x^{(0)} \in \mathbf{dom}\, f$, and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in \mathbf{dom}\, f$
Determine the number of iterations T and the step-size $\eta > 0$ based on ε and other parameters.
For $t = 0, 1, 2 \ldots, T-1$ do
    $x^{(t+1)} = x^{(t)} - \eta \nabla f(x^{(t)})$;
Return $x^{(\text{out})} = \arg\min_t f(x^{(t)})$

A few remarks are in order:

- The way we determine the number of iterations is not specified here. It may depend on different parameters of the function we will discuss later.
- The *step-size* η will also depend on different parameters. We note that there are also variants of gradient descent that use varying step-sizes η_t.
- Ideally, gradient *descent* should produce a sequence of iterates with decreasing function values $f(x^{(t)}) > f(x^{(t+1)})$. As we will see, this may not always be the case: we may 'overshoot' and use a step-size η where the Taylor approximation (4.1) is no longer valid (that is, the second order term overtakes the linear approximation). For this reason, the output solution may not be the final iterate $x^{(T)}$ but an earlier one.

- The value $f(x^*)$ is (in general) unknown. Hence, we cannot know a priori whether the current iterate $x^{(t)}$ is an ε-approximate solution. We will see how, for an appropriate choice of T, we can guarantee that one of the first T iterates includes an ε-approximate solution.
- The update rule may produce iterates $x^{(t+1)}$ outside $\mathbf{dom}\, f$. We ignore this possibility and assume all iterates fall inside $\mathbf{dom}\, f$. This can be addressed as in Chapter 5 for constrained optimisation.

Motivation for the gradient direction It turns out that the direction of $-\nabla f(x^{(t)})$ corresponds to a *steepest descent* direction in the standard Euclidean norm. Namely, let us consider any descent direction v. In a small neighbourhood of x where (4.1) gives a good approximation, the rate of decrease in the function value can be written as

$$\frac{f(x) - f(x+\eta v)}{\eta} \approx \frac{f(x) - (f(x) + \eta \langle \nabla f(x), v\rangle)}{\eta} = -\langle \nabla f(x), v\rangle \ .$$

For a consistent comparison, let us normalise the direction v as $\|v\| = 1$, and look for the direction where the rate of decrease is the highest:

$$\max_{v\in\mathbb{R}^n:\, \|v\|=1} -\langle \nabla f(x), v\rangle$$

By the Cauchy-Schwarz inequality (Theorem 1.1), $-\langle \nabla f(x), v\rangle \le \|\nabla f(x)\| \cdot \|v\| = \|\nabla f(x)\|$. Further, the inequality is tight if and only if the two vectors are parallel, that is, $v = \lambda(-\nabla f(x))$ for $\lambda > 0$. Thus, the optimal choice is $v = -\nabla f(x)/\|\nabla f(x)\|$, corresponding to the gradient descent direction.

4.1 Basic analysis

Let us know show a general bound on the average decrease in the $f(x^{(t)}) - p^*$ values for a given step-size $\eta > 0$. We have $\nabla f(x^{(t)}) = (x^{(t)} - x^{(t+1)})/\eta$ in each iteration. By the first order characterisation of convexity (Theorem 2.10), we have

$$f(x^{(t)}) - p^* \le \left\langle \nabla f(x^{(t)}), x^{(t)} - x^* \right\rangle = \frac{1}{\eta}\left\langle x^{(t)} - x^{(t+1)}, x^{(t)} - x^* \right\rangle \tag{4.2}$$

We apply the cosine theorem: $2v^\top w = \|v\|^2 + \|w\|^2 - \|v-w\|^2$ to the vectors $v = x^{(t)} - x^{(t+1)}$ and $w = x^{(t)} - x^*$ to obtain

$$\begin{aligned} f(x^{(t)}) - p^* &\le \frac{1}{2\eta}\left(\left\|x^{(t+1)} - x^{(t)}\right\|^2 + \left\|x^{(t)} - x^*\right\|^2 - \left\|x^{(t+1)} - x^*\right\|^2\right) \\ &= \frac{\eta}{2}\left\|\nabla f(x^{(t)})\right\|^2 + \frac{1}{2\eta}\left(\left\|x^{(t)} - x^*\right\|^2 - \left\|x^{(t+1)} - x^*\right\|^2\right) \end{aligned} \tag{4.3}$$

Let us now average these inequalities for $t = 0, 1, \ldots, T-1$. There is a telescoping sum cancelling out most $\left\|x^{(t)} - x^*\right\|^2$ terms:

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1}\left(f(x^{(t)}) - p^*\right) &\le \frac{\eta}{2T}\sum_{t=0}^{T-1}\left\|\nabla f(x^{(t)})\right\|^2 + \frac{1}{2T\eta}\left(\left\|x^{(0)} - x^*\right\|^2 - \left\|x^{(T)} - x^*\right\|^2\right) \\ &\le \frac{\eta}{2T}\sum_{t=0}^{T-1}\left\|\nabla f(x^{(t)})\right\|^2 + \frac{1}{2T\eta}\left\|x^{(0)} - x^*\right\|^2 \ . \end{aligned} \tag{4.4}$$

Hence, we obtain an ε-approximate solution among the first T iterates whenever the left hand side can be bounded by ε. In what follows, we formulate conditions on f and $x^{(0)}$ that enable such upper bounds.

4.2 Gradient descent for Lipschitz-continuous functions

Let us start with investigating a class of functions where (4.4) already gives a useful estimate. We start by recalling the concept of *Lipschitz-continuity*.

Definition 4.1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is* Lipschitz-continuous *with parameter $L > 0$, or* L-Lipschitz *if for any $x, y \in \mathbf{dom}\,(f)$,*

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

For differentiable functions, this can be equivalently characterised by bounded gradients:

Theorem 4.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Then, f is Lipschitz-continuous with parameter L if and only if $\|\nabla f(x)\| \leq L$ for every $x \in \mathbf{dom}\, f$.*

Proof. Let us first show that every Lipschitz-continuous function has bounded gradients. Let us consider the directional derivative at x in the direction $v = \nabla f(x)$, and use the Lipschitz-estimate $|f(x + vt) - f(x)| \leq L\|vt\| = L|t|\|v\|$:

$$\|\nabla f(x)\|^2 = \langle \nabla f(x), v\rangle = \frac{\partial f(x)}{\partial v} = \lim_{t\to 0} \frac{f(x + vt) - f(x)}{t} \leq \lim_{t\to 0} \frac{L|t|\|v\|}{|t|} = L\|v\| = L\|\nabla f(x)\|\,,$$

implying $\|\nabla f(x)\| \leq L$.

Conversely, assume $\|\nabla f(z)\| \leq L$ for every $z \in \mathbf{dom}\, f$, and select any $x, y \in \mathbf{dom}\, f$. We apply the fundamental theorem of calculus (Theorem 1.13). Let us define $g : [0, 1] \to \mathbb{R}$ as $g(t) = f(y+t(x-y))$. Then,

$$|f(x) - f(y)| = \left|\int_0^1 \dot{g}(t)dt\right| = \left|\int_0^1 \langle \nabla f(y + t(x - y)), x - y\rangle\, dt\right| \leq L\|x - y\|\,.$$

We used Lemma 1.14(ii) in the second equation. We then used the Cauchy-Schwarz inequality and the uniform bound on the gradient to show that $|\langle \nabla f(y + t(x - y)), x - y\rangle| \leq L\|x - y\|$. □

For example, the function $\sin(t)$ is Lipschitz-continuous with constant $L = 1$, since its derivative is $|\cos(t)| \leq 1$. However, the function $f(t) = t^\alpha$ on the domain $t \geq 0$ is only Lipschitz-continuous for $\alpha = 1$. If $\alpha > 1$, then $\lim_{t\to\infty} \dot{f}(t) \to \infty$, and if $\alpha < 1$, then $\lim_{t\to 0} \dot{f}(t) \to \infty$.

Let us analyse gradient descent with constant step-size η for a Lipschitz-continuous function with parameter L. Continuing from (4.4), we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1} \left(f(x^{(t)}) - p^*\right) \leq \frac{\eta L^2}{2} + \frac{1}{2T\eta}\left\|x^{(0)} - x^*\right\|^2\,.$$

Assume further we have a bound R available such that $\left\|x^{(0)} - x^*\right\| \leq R$. Then, the right hand side can be bounded as

$$\frac{\eta L^2}{2} + \frac{R^2}{2T\eta}\,.$$

For a given iteration number T, this expression is minimised by choosing

$$\eta = \frac{R}{L\sqrt{T}}\,.$$

For this choice, the bound becomes $RL/\sqrt{T}$. In order to guarantee a ε-approximate solution, we need to pick T such that $RL/\sqrt{T} \leq \varepsilon$, that is,

$$T \geq \frac{R^2L^2}{\varepsilon^2}\,.$$

Thus, we have proved the following.

Theorem 4.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz-continuous and convex differentiable function with Lipschitz-parameter L. Assume that a global minimum x^* exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$. Then, for any $\varepsilon > 0$, gradient descent finds an ε-approximate solution within $T \geq R^2L^2/\varepsilon^2$ iterations, using step-size $\eta = R/(L\sqrt{T})$.*

This gives our first running time bound on gradient descent. The applicability is limited, since Lipschitz-continuity is a rather restrictive assumption. The quadratic dependence $1/\varepsilon^2$ on the desired accuracy can be prohibitive even for small R and L values.

Note however that already this running time bound is independent on n, the dimension of the problem. This is a crucial feature that allows gradient descent to be applied to high dimensional problems.

Guessing the parameters Here, as well as in subsequent results, we use parameters such as R and L. Such estimates may be available a priori, but can be difficult to obtain in many cases. In such scenarios, we can run our algorithm with *'guesses'* $\hat{R}$ and $\hat{L}$.

Assume we run the algorithm with a guess $\hat{L}$ on L, initialised say as $\hat{L} = 1$. When running the algorithm, we can check whether $\left\|\nabla f(x^{(t)})\right\| \leq \hat{L}$ holds in each iteration. If this is violated at any point, then we can update $\hat{L}$ to $\min\{\left\|\nabla f(x^{(t)})\right\|, 2L\}$, and restart the algorithm. If the assumption holds throughout, then the analysis remains valid with the value $\hat{L}$ instead of L (even if the function was not L-Lipschitz).

The estimate $\hat{R}$ is not verifiable in a similar sense: the algorithm does not provide any evidence on whether $\|x^{(0)} - x^*\| \leq \hat{R}$.

4.3 Gradient descent for M-smooth functions

We now focus on another important family of functions, when the gradient (as an $\mathbb{R}^n \to \mathbb{R}^n$ function) is Lipschitz-continuous.

Definition 4.4. *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is* M-smooth *for a parameter $M > 0$ if for any $x, y \in \mathbf{dom}\,(f)$,*

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|\,. \tag{4.5}$$

There are multiple ways to characterise M-smoothness. Part (ii) the analogue of Theorem 4.2.

Theorem 4.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function.*

(i) f is M-smooth if and only if

$$|D_f(x, y)| \leq \frac{M}{2}\|x - y\|^2 \quad \forall x, y \in \mathbf{dom}\, f\,. \tag{4.6}$$

(ii) If f is twice differentiable, then f is M-smooth if and only if

$$-MI_n \preceq \nabla^2 f(z) \preceq MI_n \quad \forall z \in \mathbf{dom}\, f\,. \tag{4.7}$$

Proof. **Part (i):** We only show the 'only if' direction. Assume f is M-smooth. We again use Theorem 1.13 similarly as in the proof of Theorem 4.2. Let us define $g : [0, 1] \to \mathbb{R}$ as $g(t) = f(y + t(x - y))$.

$$\begin{aligned}
D_f(x, y) &= f(x) - f(y) - \langle \nabla f(y), x - y\rangle \\
&= \int_0^1 \dot{g}(t)dt - \langle \nabla f(y), x - y\rangle \\
&= \int_0^1 \langle \nabla f(y + t(x - y)), x - y\rangle\, dt - \int_0^1 \langle \nabla f(y), x - y\rangle\, dt \\
&= \int_0^1 \langle \nabla f(y + t(x - y)) - \nabla f(y), x - y\rangle\, dt\,.
\end{aligned}$$

Therefore, we can use the Cauchy-Schwarz inequality to bound

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

In the second inequality we used the definition of M-smoothness.

Part (ii) Here, we prove only the 'if' direction. According to part *(i)*, it suffices to that (4.7) implies (4.6). This follows easily from the Taylor-expansion: for any $x, y \in \mathbf{dom}\, f$,

$$D_f(x,y) = \frac{1}{2}(x-y)^\top \nabla^2 f(z)(x-y)$$

for some point $z \in [x,y]$. By the definition of the semidefinite ordering $\preceq$, condition (4.7) implies that

$$(x-y)^\top \nabla^2 f(z)(x-y) \leq (x-y)^\top (MI_n)(x-y) = M\|x-y\|^2 .$$

Similarly, we get the lower bound

$$(x-y)^\top \nabla^2 f(z)(x-y) \geq -M\|x-y\|^2 .$$

Thus, $|D_f(x,y)| \leq \frac{M}{2}\|x-y\|^2$ follows. □

The L-Lipschitz and M-smoothness properties mutually do not imply each other. The function $f(t) = t^2$ on $\mathbf{dom}\, f = \mathbb{R}$ is 2-smooth, but not Lipschitz for any constant. We will see in the exercises that the converse direction is also not true: the L-Lipschitz property does not imply bounded smoothness.

M-smoothness implies descent The variant of gradient descent for L-Lipschitz functions in Section 4.2 is not a true *descent* method: $f(x^{(t+1)}) > f(x^{(t)})$ may be possible. In contrast, for M-smooth functions we get a significant decrement for step-size $\eta = 1/M$.

Lemma 4.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an M-smooth function. Using the step-size $\eta = 1/M$, the subsequent iterates satisfy*

$$f(x^{(t+1)}) \leq f(x^{(t)}) - \frac{1}{2M}\|\nabla f(x^{(t)})\|^2 . \tag{4.8}$$

Proof. We first argue with an arbitrary step-size $\eta > 0$. Thus, $x^{(t+1)} - x^{(t)} = -\eta \nabla f(x^{(t)})$. The M-smoothness property gives

$$\begin{aligned}f(x^{(t+1)}) &= f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x^{(t+1)} - x^{(t)} \right\rangle + D_f(x^{(t+1)}, x^{(t)}) \\ &\leq f(x^{(t)}) - \eta\|\nabla f(x^{(t)})\|^2 + \frac{M}{2}\|x^{(t+1)} - x^{(t)}\|^2 \\ &= f(x^{(t)}) + \left(\frac{M\eta^2}{2} - \eta\right)\|\nabla f(x^{(t)})\|^2 .\end{aligned}$$

We can see that the minimum of this expression is reached for the choice $\eta = 1/M$, leading to the claimed bound. □

This lemma, together with the analysis in Section 4.1, leads to the following theorem.

Theorem 4.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an M-smooth function. Assume that a global minimum x^* exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$. Then, for any $\varepsilon > 0$, gradient descent finds an ε-approximate solution $x^{(T)}$ within $T \geq MR^2/(2\varepsilon)$ iterations, using step-size $\eta = 1/M$.*

Proof. Let us substitute $\eta = 1/M$ in (4.4); this gives

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(f(x^{(t)}) - p^*\right) \leq \frac{1}{2MT}\sum_{t=0}^{T-1}\left\|\nabla f(x^{(t)})\right\|^2 + \frac{M}{2T}\left\|x^{(0)} - x^*\right\|^2 .$$

Summing up (4.8) for $t = 0, 1, \ldots, T-1$ and after cancellations, we obtain

$$\frac{1}{2M}\sum_{t=0}^{T-1}\left\|\nabla f(x^{(t)})\right\|^2 \leq f(x^{(0)}) - f(x^{(T)}) .$$

Substituting, this leads to

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(f(x^{(t)}) - p^*\right) \leq \frac{1}{T}\left(f(x^{(0)}) - f(x^{(T)})\right) + \frac{M}{2T}\left\|x^{(0)} - x^*\right\|^2 .$$

By adding $-p^*/T$ and p^*/T to the LHS,

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(f(x^{(t)}) - p^*\right) \leq \frac{1}{T}\left((f(x^{(0)}) - p^*) - (f(x^{(T)}) - p^*)\right) + \frac{M}{2T}\left\|x^{(0)} - x^*\right\|^2 .$$

Moving the terms $f(x^{(0)}) - p^*$ and $-(f(x^{(T)}) - p^*)$ to the RHS, note that the index in the summation changes from $t = 0, \ldots, T-1$ to $t = 1, \ldots, T$. That is,

$$f(x^{(T)}) - p^* \leq \frac{1}{T}\sum_{t=1}^{T}\left(f(x^{(t)}) - p^*\right) \leq \frac{M}{2T}\left\|x^{(0)} - x^*\right\|^2 \leq \frac{MR^2}{2T} ,$$

where the first inequality follows by the descent property in Lemma 4.6, and the last inequality uses the definition of R. Hence, if we select

$$T \geq \frac{MR^2}{2\varepsilon} ,$$

then $f(x^{(T)})$ must be a ε-approximate solution. □

This theorem has a much more favourable parameter dependence compared to Theorem 4.3. Namely, the number of steps is proportional to $1/\varepsilon$ instead of $1/\varepsilon^2$. E.g. for $\varepsilon = 0.01$, this means a hundred times fewer steps.

Smoothness on the sublevel set Theorem 4.7 assumes that f is M-smooth on its entire domain. This assumption can be weakened. Consider the *sublevel set* defined by the initial point $x^{(0)}$:

$$S = \{x \in \mathbf{dom}\, f \mid f(x) \leq f(x^{(0)})\}. \tag{4.9}$$

For the arguments above, it suffices to require that f is M-smooth on the set S, that is, (4.5) holds for points $x, y \in S$. This is again because of Lemma 4.6: that guarantees all iterates stay inside S (and the proof of the lemma only uses (4.5) holds for points $x, y \in S$). Note that the convexity of f implies that S is a convex set.

4.3.1 Accelerated gradient descent

[Non examinable]

For M-smooth functions, gradient descent with step-length $\eta = 1/M$ achieves a $1/\varepsilon$-dependence for finding a ε-accurate solution. This is not the best possible: Nesterov's *accelerated gradient descent* method improves this to $1/\sqrt{\varepsilon}$-dependence. We do not cover this method in the course: the interested readers may find a description, along with the explanation of the background, e.g. in [6, Chapter 8].

Interestingly, the $1/\sqrt{\varepsilon}$-dependence turns out to be the best possible for a gradient method. Consider any algorithm that satisfies the following general property. Starting form a given $x^{(0)}$, we always pick

$$x^{(t+1)} \in x^{(0)} + \operatorname{span}(\{\nabla f(x^{(0)}), \nabla f(x^{(1)}), \dots, \nabla f(x^{(t)})\}), \tag{4.10}$$

that is, we are only allowed to move in the linear space spanned by the gradients seen thus far. This is clearly the case for gradient descent, where we can write $x^{(t+1)} = x^{(0)} - \eta \sum_{j=0}^{t} \nabla f(x^{(j)})$.

Nemirovski and Yudin constructed a family of functions such that any algorithm that chooses updates comforming (4.10) must take at least $c\sqrt{M}R/\sqrt{\varepsilon}$ steps to obtain a ε-approximate solution for some constant $c > 0$. Consequently, the accelerated gradient descent method is essentially the best possible algorithm that uses only gradient information.

4.4 Gradient descent for well-conditioned functions

We next discuss variants of gradient descent that obtain a dramatically faster, $\log(1/\varepsilon)$-dependence. As noted above, this may not be possible for all M-smooth functions. We make another natural assumption on strong convexity.

4.4.1 Strong convexity

Definition 4.8. *Let $m > 0$. We say that the function $f : \mathbb{R}^n \to \mathbb{R}$ is* strongly convex on a convex set S with parameter m, *if*

$$D_f(x, y) \geq \frac{m}{2}\|x - y\|^2 \quad \forall x, y \in S\,.$$

For $S = \mathbf{dom}\, f$, we simply say that the function is strongly convex with parameter m.

Strong convexity thus gives the opposite bound as M-smoothness: it shows that the lower bound $f(y) - \langle \nabla f(y), x - y\rangle$ is always strictly below $f(x)$. Hence, affine functions $f(x) = \langle w, x\rangle + d$ are not strongly convex for any $m > 0$. Analogously to Theorem 4.5, we can relate strong convexity to the Hessian for twice differentiable functions.

Theorem 4.9. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function. Then f is m-strongly convex if and only if*

$$\nabla^2 f(z) \succeq m I_n \quad \forall z \in \mathbf{dom}\, f\,. \tag{4.11}$$

The proof of the 'if' direction follows analogously to the proof of Theorem 4.5, using the Taylor expansion.

An important property of strongly convex functions is that they have unique optimal solutions.

Proposition 4.10. *If f is strongly convex and a local minimum exists, then there exists a unique global minimiser.*

Proof. Let x^* be a local minimum and $x \in \mathbf{dom}\, f$ be an arbitrary point. Convexity already implies that x^* is a global minimum; we now argue for uniqueness. By the definition of strong convexity, for any $x \neq x^*$ we obtain

$$f(x) \geq f(x^*) + \langle \nabla f(x^*), x - x^*\rangle + \frac{m}{2}\|x - x^*\|^2 = f(x^*) + \frac{m}{2}\|x - x^*\|^2 > f(x^*).$$

This completes the proof. □

Convex quadratic functions An important example is convex quadratic functions. Let $f(x) = x^\top Qx + \langle p, x\rangle + r$; recall from Theorem 2.14 that f is convex if and only if Q is positive semidefinite. Then, $\nabla f(x) = 2Qx + p$, and the Hessian is $\nabla^2 f(x) = 2Q$. Let $\lambda_1 \geq 0$ be the smallest eigenvalue of Q. Then, $v^\top Qv \geq \lambda_1 \|v\|^2$ for any $v \in \mathbb{R}$. Thus, if Q is positive definite, that is, if $\lambda_1 > 0$, then $f(x)$ is strongly convex with $m = 2\lambda_1$. If Q is positive semidefinite but not positive definite, then $\lambda_1 = 0$ and therefore the function is not strongly convex.

Bounding the distance from optimality

For arbitrary convex functions, we cannot determine whether our current solution is already near the optimum. A significant advantage of strongly convex functions is that a small gradient $\|\nabla f(x)\|$ indicates that $f(x)$ is approximately optimal.

Proposition 4.11. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbf{dom}\, f$, and let $p^* = f(x^*)$ denote the optimum value. Assume f is strongly convex with parameter m on $S = \{z \in \mathbb{R}^n : f(z) \leq f(x)\}$. Then,*

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

In particular, if $\|\nabla f(x)\| \leq \sqrt{2m\varepsilon}$, then $f(x) - p^* \leq \varepsilon$.

Proof. The definition of strong convexity gives $D_f(z, x) \geq \frac{m}{2}\|z - x\|^2$ for any $z \in S$, that is,

$$f(z) \geq f(x) + \langle \nabla f(x), z - x\rangle + \frac{m}{2}\|z - x\|^2. \tag{4.12}$$

Let us consider the function $g(z) = \langle \nabla f(x), z - x\rangle + \frac{m}{2}\|z - x\|^2$ for the fixed x. This is a convex quadratic function in z. The minimum is taken where the gradient is 0, which is at

$$0 = \nabla g(z) = \nabla f(x) + m(z - x).$$

Consequently, the minimiser of $g(z)$ is $\tilde{z} = x - \frac{1}{m}\nabla f(x)$, that is, for any $z \in \mathbb{R}^n$, $g(z) \geq g(\tilde{z}) = -\frac{1}{2m}\|\nabla f(x)\|^2$. From (4.12), we obtain

$$f(z) \geq f(x) + g(z) \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|^2.$$

This holds true for any $z \in S$, in particular, for $z = x^*$, in which case we obtain the desired

$$p^* \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|^2. \tag{4.13}$$

□

4.4.2 The condition number

Assume now that the function is m-strongly convex and M-smooth at the same time. Then, we have that for every $x \in S$,

$$mI_n \preceq \nabla^2 f(x) \preceq MI_n.$$

Thus, the eigenvalues of $\nabla^2 f(x)$ fall in the range $[m, M]$. We call the ratio

$$\kappa = \frac{M}{m}$$

the *condition number* of f. We will see that the convergence of gradient descent can be bounded in terms of this condition number.

4.4.3 Convergence analysis for bounded condition number

Let us apply gradient descent to a convex function f, starting with an initial point $x^{(0)} \in \mathbf{dom}\, f$. Assume that the function is both m-strongly convex and M-smooth for $0 < m \leq M$ on the sublevel set

$$S = \{x \in \mathbf{dom}\, f \mid f(x) \leq f(x^{(0)})\}.$$

We show that with step-length $\eta = 1/M$, the running time of gradient descent can be bounded in terms of $\kappa = M/m$ and $\log(1/\varepsilon)$.

We can apply Lemma 4.6 for M-smooth functions with step-length $\eta = 1/M$ to obtain

$$f(x^{(t+1)}) \leq f(x^{(t)}) - \frac{1}{2M}\|\nabla f(x^{(t)})\|^2 .$$

Subtracting the optimum value p^* from both sides, we see that

$$f(x^{(t+1)}) - p^* \leq f(x^{(t)}) - p^* - \frac{1}{2M}\|\nabla f(x^{(t)})\|^2.$$

Using $\|\nabla f(x^{(t)})\|^2 \geq 2m(f(x^{(t)}) - p^*)$ from Proposition 4.11, we get

$$f(x^{(t+1)}) - p^* \leq \left(1 - \frac{m}{M}\right) \cdot \left(f(x^{(t)}) - p^*\right) .$$

Hence, the distance from optimality decreases by a factor $1 - m/M = 1 - 1/\kappa$ in every iteration. Applying this argument at every iteration, we see that for every $t \geq 1$,

$$f(x^{(t)}) - p^* \leq \left(1 - \frac{1}{\kappa}\right)^t \cdot \left(f(x^{(0)}) - p^*\right) .$$

This shows that $f(x^{(t)})$ converges quickly to p^* as $t \to \infty$. We can also relate this to the bound R on $\|x^{(0)} - x^*\|$. We have

$$f(x^{(0)}) - p^* = \left\langle \nabla f(x^*), x^{(0)} - x^* \right\rangle + D_f(x^{(0)}, x^*) \leq 0 + \frac{M}{2}\|x^{(0)} - x^*\|^2 \leq \frac{MR^2}{2} .$$

We can terminate once

$$\left(1 - \frac{1}{\kappa}\right)^t \cdot \frac{MR^2}{2} \leq \varepsilon .$$

After taking logarithms, we get

$$t \log\left(1 - \frac{1}{\kappa}\right) + \log\left(\frac{MR^2}{2\varepsilon}\right) \leq 0 .$$

Rearranging,

$$\log\left(\frac{MR^2}{2\varepsilon}\right) \leq t \log\left(1 + \frac{1}{\kappa - 1}\right)$$

Using the inequality $\log(1 + \alpha) \leq \alpha$ for all $\alpha > -1$, we obtain

$$(\kappa - 1) \log\left(\frac{MR^2}{2\varepsilon}\right) \leq t .$$

We have proved the following theorem.

Theorem 4.12. *For $f : \mathbb{R}^n \to \mathbb{R}$ with starting point $x^{(0)} \in \mathbf{dom}\, f$, assume f is m-strongly convex and M-smooth on the sublevel set S. Let $\kappa = M/m > 1$. Assume that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$ and the global optimum x^*. Then, gradient descent with step-length $\eta = 1/M$ obtains an ε-approximate solution $x^{(T)}$ within*

$$T \leq (\kappa - 1) \log\left(\frac{MR^2}{2\varepsilon}\right)$$

iterations.

Compared to Theorem 4.7, note that we take the logarithm of the term $\frac{MR^2}{2\varepsilon}$, giving a dramatically better bound. However, the condition number κ could be potentially very large.

We also note that, in contrast to the previous variants, we do not necessarily need to perform the number of iterations prescribed in the theorem. Proposition 4.11 gives a simple stopping criterion: we can stop once $\|\nabla f(x^{(t)}\| \leq 2m\varepsilon$ for the first time.

Chapter 5

Gradient methods for constrained optimisation

In this chapter, we consider the more general *constrained optimisation* setting. For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, and a nonempty closed convex set $K \subseteq \mathbf{dom}\, f$ we aim to solve

$$\min_{x \in K} f(x). \tag{5.1}$$

Again, we assume that an optimal solution $x^* \in K$ exists and let $p^* = f(x^*)$ denote the optimum value.

We consider two fundamental approaches: the *projected gradient method* in Section 5.1 that adds an additional projection step to gradient descent to force the iterates back to K; and the *conditional gradient method* in Section 5.2 that uses gradients to find directions inside K.

5.1 Projected gradient method

The overall idea is simple: we run the standard gradient method, but whenever the next step would go outside the feasible region K, we project it back to K. This is done using the projection mapping introduced in Section 1.1.1 defined as

$$\Pi_K(x) = \arg\min_{v \in K} \|x - v\| .$$

We recall that for a nonempty closed convex K, there is a unique optimal solution, and $\Pi_K(x) = x$ if and only if $x \in K$.

PROJECTED GRADIENT DESCENT
Input: A convex function $f : \mathbb{R}^n \to \mathbb{R}$, a nonempty closed convex set $K \subseteq \mathbf{dom}\, f$, a starting point $x^{(0)} \in K$, and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in K$
Determine the number of iterations T and the step-size $\eta > 0$ based on ε and other parameters.
For $t = 0, 1, 2 \ldots, T-1$ do
  $y^{(t+1)} = x^{(t)} - \eta \nabla f(x^{(t)})$;
  $x^{(t+1)} = \Pi_K(y^{(t+1)})$;
Return $x^{(\text{out})} = \arg\min_t f(x^{(t)})$

5.1.1 Properties of the projection map

Recall from the proof of Theorem 1.5 that projections give rise to separating hyperplanes. Let us now state this property explicitly:

Proposition 5.1. *Let $K \subseteq \mathbb{R}^n$ be a nonempty closed convex set. For every $y \in \mathbb{R}^n$ and $x \in K$, we have*

$$\langle y - \Pi_K(y), x - \Pi_K(y) \rangle \leq 0 \ .$$

Proof. If $y \in K$ then $\Pi_K(y) = y$, and thus the statement holds trivially. For $y \notin K$, this is shown as (1.1) in the proof of Theorem 1.5. □

Using this proposition, one can easily verify the following inequality. The geometric interpretation is that the vectors $y - \Pi_K(y)$ and $x - \Pi_K(y)$ have an obtuse angle if $y \notin K$, $x \in K$, see Figure 5.1.

Lemma 5.2. *Let $K \subseteq \mathbb{R}^n$ be a nonempty closed convex set. For every $y \in \mathbb{R}^n$ and $x \in K$, we have*

$$\|y - \Pi_K(y)\|^2 + \|x - \Pi_K(y)\|^2 \leq \|x - y\|^2 \, .$$

Proof. We apply the identity $2\langle u, v \rangle = \|u\|^2 + \|v\|^2 - \|u - v\|^2$ for $u = y - \Pi_K(y)$ and $v = x - \Pi_K(y)$. □

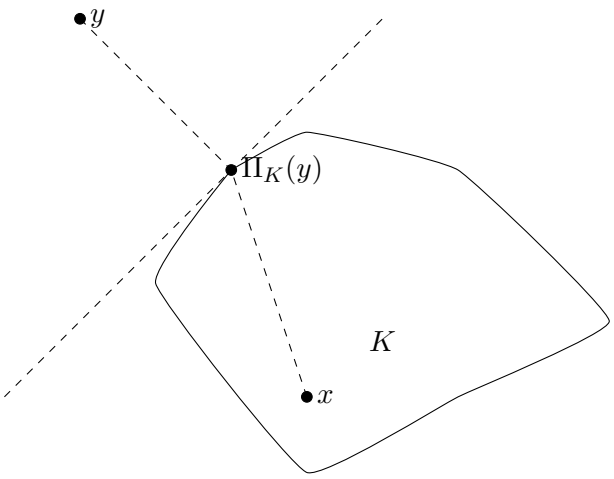


Figure 5.1: The projection map

The algorithm requires a subroutine for computing $\Pi_K(z)$. For general K, this is itself a constrained convex quadratic optimisation problem. Hence, conditional gradient can only be implemented if minimising the quadratic objective $\|x - z\|^2$ over $x \in K$ is inherently 'simpler' than minimising the objective function $f(x)$.

Example 5.3. Euclidean ball Let $K = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$ be the Euclidean ball of radius 1 around the origin. Then, it is immediate that

$$\Pi_K(x) = \begin{cases} x & \text{if } \|x\| \leq 1 \, , \\ \frac{x}{\|x\|} & \text{if } \|x\| > 1 \, . \end{cases}$$

Example 5.4. ℓ_1-ball Let us now consider the ℓ_1-ball $K = \{x \in \mathbb{R}^n : \|x\|_1 \leq 1\}$. Projection to this set can also be computed using an explicit formula, but it is significantly more complicated than for the ℓ_2-ball. We write the formula here without the proof. It can be derived from the generalisation of KKT conditions for subgradients. For $a \in \mathbb{R}$, we use the notation $a^+ = \max\{a, 0\}$ for the positive part of a, and $\operatorname{sign}(a) \in \{0, \pm 1\}$ to denote the sign of a.

If $\|x\|_1 \leq 1$, then $\Pi_K(x) = x$. Otherwise, define the parameter $\lambda \in \mathbb{R}_+$ as the unique value such that

$$\sum_{i=1}^{n} (|x_i| - \lambda)^+ = 1 \, . \tag{5.2}$$

Such a unique value exists since the left hand side is a continuous strictly monotone decreasing function of λ on the interval $(-\infty, \|x\|_\infty]$. If $\|x\|_1 > 1$ then we must have $\lambda > 0$. To determine the value of λ as in (5.2), let us reorder the indicies for a decreasing order $|x_1| \geq |x_2| \geq \ldots \geq |x_n|$. For $k = 1, 2, \ldots$, compute $q_k = (\sum_{i=1}^{k} |x_i| - 1)/k$, and stop with the first k such that $x_k \geq q_k \geq x_{k+1}$, or set $k = n$ if $x_n \geq q_n$. We let $\lambda = q_k$ for this value.

For the value of λ satisfying (5.2), the projection will be $y = \Pi_K(x)$ such that

$$y_i = \operatorname{sign}(y_i)(|x_i| - \lambda)^+ .$$

Note that if $x_i < -\lambda$, then $y_i = x_i + \lambda$, if $x_i \in [-\lambda, \lambda]$ then $y_i = 0$, and if $x_i > \lambda$ then $y_i = x_i - \lambda$.

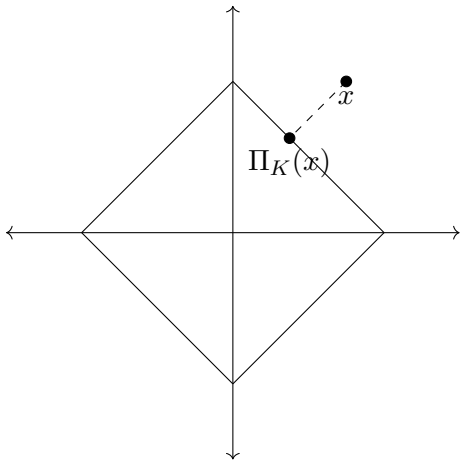


Figure 5.2: Projection to the ℓ_1-ball

5.1.2 Basic analysis of the projected gradient method

Using Proposition 5.1 and Lemma 5.2, the analyses of the unconstrained cases in Chapter 4 can be extended with some modifications. Let us revisit the basic analysis in Section 4.1. Using that $\nabla f(x^{(t)}) = (x^{(t)} - y^{(t+1)})/\eta$ (with $y^{(t+1)}$ instead of $x^{(t+1)}$), (4.2) becomes

$$f(x^{(t)}) - p^* \leq \left\langle \nabla f(x^{(t)}), x^{(t)} - x^* \right\rangle = \frac{1}{\eta} \left\langle x^{(t)} - y^{(t+1)}, x^{(t)} - x^* \right\rangle , \tag{5.3}$$

and (4.3) becomes

$$\begin{aligned} f(x^{(t)}) - p^* &\leq \frac{1}{2\eta} \left(\left\| y^{(t+1)} - x^{(t)} \right\|^2 + \left\| x^{(t)} - x^* \right\|^2 - \left\| y^{(t+1)} - x^* \right\|^2 \right) \\ &= \frac{\eta}{2} \left\| \nabla f(x^{(t)}) \right\|^2 + \frac{1}{2\eta} \left(\left\| x^{(t)} - x^* \right\|^2 - \left\| y^{(t+1)} - x^* \right\|^2 \right) . \end{aligned} \tag{5.4}$$

We now use Lemma 5.2 for $y = y^{(t+1)}$ and $x = x^*$, which gives

$$\left\| x^{(t+1)} - x^* \right\|^2 + \left\| y^{(t+1)} - x^{(t+1)} \right\|^2 \leq \left\| y^{(t+1)} - x^* \right\|^2 .$$

Therefore, the above can be further bounded as

$$f(x^{(t)}) - p^* \leq \frac{\eta}{2} \left\| \nabla f(x^{(t)}) \right\|^2 + \frac{1}{2\eta} \left(\left\| x^{(t)} - x^* \right\|^2 - \left\| x^{(t+1)} - x^* \right\|^2 \right) - \frac{1}{2\eta} \left\| y^{(t+1)} - x^{(t+1)} \right\|^2 , \tag{5.5}$$

Averaging these inequalities, we obtain

$$\frac{1}{T} \sum_{t=0}^{T-1} \left(f(x^{(t)}) - p^* \right) \leq \frac{\eta}{2T} \sum_{t=0}^{T-1} \left\| \nabla f(x^{(t)}) \right\|^2 + \frac{1}{2T\eta} \left\| x^{(0)} - x^* \right\|^2 - \frac{1}{2T\eta} \sum_{t=0}^{T-1} \left\| y^{(t+1)} - x^{(t+1)} \right\|^2 . \tag{5.6}$$

This implies that the bound (4.4) remains valid even in the projected gradient setting!

We can immediately apply this to the setting when the function is Lipschitz-continuous inside K, i.e., $|f(x) - f(y)| \leq L\|x - y\|$ for any $x, y \in K$ holds for some $L > 0$. As in Theorem 4.2, this is equivalent to $\|\nabla f(x)\| \leq L$ for any $x \in K$.

Using (5.6) and ignoring the last term, the proof is identical to that of Theorem 4.3.

Theorem 5.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and $K \subseteq \mathbf{dom}\, f$ a nonempty closed convex set. Assume f is differentiable and has Lipschitz-parameter L on K. Assume that a global minimum $x^* = \arg\min_{x \in K} f(x)$ exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$. Then, for any $\varepsilon > 0$, projected gradient descent finds an ε-approximate solution within $T \geq R^2L^2/\varepsilon^2$ iterations, using step-size $\eta = R/(L\sqrt{T})$.*

5.1.3 Projected gradient method for M-smooth functions

Let us now assume that f is M-smooth on K, that is, $\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|$ holds for every $x, y \in K$. We revisit the analysis of Section 4.3. In place of Lemma 4.6, one can show the following weaker bound that has an additional term. We will prove this as a class exercise.

Lemma 5.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an M-smooth function. Using the step-size $\eta = 1/M$, the subsequent iterates satisfy*

$$f(x^{(t+1)}) \leq f(x^{(t)}) - \frac{1}{2M}\|\nabla f(x^{(t)})\|^2 + \frac{M}{2}\|y^{(t+1)} - x^{(t+1)}\|^2 . \tag{5.7}$$

Note that, in contrast to unconstrained gradient descent, M-smoothness does not guarantee descent for step-size $\eta = 1/M$. That is, $f(x^{(t+1)}) \geq f(x^{(t)})$ may still be possible due to the second term.

Using this lemma together with (5.6), we obtain the generalisation of Theorem 4.7. The proof is an immediate extension, noting that the additional term $\frac{M}{2T}\sum_{t=0}^{T-1}\|y^{(t+1)} - x^{(t+1)}\|^2$ is graciously cancelled out in (5.6).

Theorem 5.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an M-smooth function and $K \subseteq \mathbf{dom}\, f$ a nonempty closed convex set. Assume that a global minimum $x^* = \arg\min_{x \in K} f(x)$ exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)} \in K$. Then, for any $\varepsilon > 0$, projected gradient descent finds an ε-approximate solution $x^{(t)}$ within $T \geq R^2M/\varepsilon$ iterations, using step-size $\eta = 1/M$.*

5.1.4 Projected gradient for well-conditioned functions

Let us now revisit the case of well-conditioned functions, i.e., functions that are both M-smooth and strongly m-convex for some $0 < m < M$; we let $\kappa = M/m$. The analogue of Theorem 4.12 remains true:

Theorem 5.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $K \subseteq \mathbf{dom}\, f$ a nonempty closed convex set. Let us be given a starting point $x^{(0)} \in K$, and assume f is m-strongly convex and M-smooth on the sublevel set $S \cap K$. Assume that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$ and the global optimum $x^* = \arg\min_{x \in K} f(x)$. Then, projected gradient descent with step-size $\eta = 1/M$ obtains an ε-approximate solution $x^{(T)}$ within*

$$T \leq (\kappa - 1)\log\left(\frac{MR^2}{2\varepsilon}\right)$$

iterations.

However, the proof of Theorem 4.12 does not immediately extend. We do not prove this here, just point out two critical points. First, Lemma 5.6 replaces Lemma 4.6, leading to a weaker bound. Second, the stopping criterion implied by Proposition 4.11 is, while valid, insufficient. Since we have a constrained optimisation problem, $\nabla f(x^*)$ is not necessarily 0, and near-optimal solutions do not necessarily have a small gradient.

5.2 Conditional gradient method

The main drawback of the projected gradient method is that we need to be able to compute the projection map $\Pi_K(x)$. This may be difficult unless the feasible set K is of a very simple form.

A different approach is the *conditional gradient method*, also called the *Frank–Wolfe algorithm*. The algorithm remains inside K throughout, without the need for projections. We will assume that K is a *nonempty compact convex* set, that is, we also assume K is bounded.

The Frank–Wolfe algorithm also requires a starting solution $x^{(0)} \in K$. At each iteration $t = 0, 1, 2, \ldots, T-1$, we determine a search direction $\Delta^{(t)}$ (which will be typically different from the gradient), step-size η_t, and update

$$x^{(t+1)} = x^{(t)} + \eta_t \Delta^{(t)} .$$

In contrast to the previous methods, we typically use a varying step-size η_t that depends on the current iteration. In order to decrease the function value, we need to move in a *decreasing direction*, that is,

$$\left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle < 0.$$

Recall from Theorem 2.10 that $x^* \in K$ is a global minimum of f over K if and only if

$$\langle \nabla f(x^*), x - x^* \rangle \geq 0 \text{ for all } x \in K .$$

Thus, if $x^{(t)}$ is not optimal, then there is a vector $s \in K$ such that

$$\left\langle \nabla f(x^{(t)}), s - x^{(k)} \right\rangle < 0 .$$

Direction finding subroutine The Frank–Wolfe algorithm finds a search direction $\Delta^{(t)} = s^{(t)} - x^{(t)}$, where $s^{(t)}$ is obtained as an optimal solution to the problem

$$\begin{aligned} &\min \left\langle \nabla f(x^{(t)}), y \right\rangle \\ &\text{s. t. } y \in K . \end{aligned} \tag{5.8}$$

By the assumption that K is compact, this problem admits an optimal solution. In case when $x^{(t)}$ is an optimal solution to this problem, we conclude that $x^{(t)}$ is an optimal solution to the original problem, as the optimality conditions are satisfied for $x^* = x^{(t)}$. Otherwise, for the optimal solution $s^{(t)} \in K$, we have $\left\langle \nabla f(x^{(t)}), s^{(t)} - x^{(t)} \right\rangle < 0$. Thus, we can use $s^{(t)} - x^{(t)}$ as the search direction.

Problem (5.8) is itself a constrained optimisation problem. However, the objective function is *linear*, which makes the problem typically simpler than the original problem (5.1). The efficiency of the Frank–Wolfe method crucially relies on the efficiency of the direction finding subroutine. If the feasible region K is given by linear constraints, the direction finding subproblem amounts to solving a linear program. Of course, solving an LP at every iteration may not be viable for practical purposes.

Example 5.9. As another example, consider the case when K is the ℓ_p-ball $B_p(0,1) = \{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$ for some $p \in [1, \infty]$. The solution easily follows from Hölder's inequality, a generalisation of the Cauchy–Schwarz inequality:

Theorem 5.10 (Hölder)**.** *Let $p, q \in [1, \infty]$ such that $1/p + 1/q = 1$. For $x, y \in \mathbb{R}^n$, we have*

$$\sum_{i=1}^{n} |x_i y_i| \leq \|x\|_p \cdot \|y\|_q .$$

Further, equality holds if and only if $|x|^p$ and $|y|^q$ are linearly dependent, that is, $|x|^p = \alpha |y|^q$ for some $\alpha \in \mathbb{R}$.

(Here, $|x|^p$ refers to the vector in $\mathbb{R}^n$ with coordinates $|x_i|^p$.) Such p and q satisfying $1/p + 1/q = 1$ are called *Hölder conjugates*. Note that the case $p = q = 2$ recovers the Cauchy–Schwarz inequality. Another fundamental case is $p = 1$, $q = \infty$.

Thus, (5.8) for $K = B_p(0,1)$ amounts to minimising $\langle c, y\rangle$ subject to $\|y\|_p \le 1$, where $c = \nabla f(x^{(t)})$; this is equivalent to maximising $-\langle c, y\rangle$. Hölder's inequality gives the upper bound

$$-\langle c, y\rangle \le \sum_{i=1}^{n} |c_i y_i| \le \|c\|_q \cdot \|y\|_p \le \|c\|_q \,,$$

where q is the Hölder conjugate of p. The first inequality is tight if $c_i y_i \le 0$ for all $i = 1, 2, \ldots, n$, that is, the two vectors have opposite signs on every component. Equality holds in the second inequality if and only if $|y|^p$ and $|c|^q$ are linearly dependent. Using that $q/p = q - 1$ for conjugate pairs, this gives $|y_i| = \alpha |c_i|^{q-1}$ for all $i = 1, 2, \ldots, n$ and $\alpha \in \mathbb{R}^n$. Finally, the third inequality is tight if and only if $\|y\|_p = 1$. This yields the choice $\alpha = 1/\||c|^{q-1}\|_q$. Putting this all together, the optimal solution can be computed as

$$y_i = -\operatorname{sign}(c_i) \frac{|c_i|^{q-1}}{\left(\sum_{i=1}^{n} |c_i|^{q(q-1)}\right)^{1/q}} \,.$$

Description of the algorithm The standard version of the algorithm uses step-size $\eta_t = 2/(t+2)$ in the t-th iteration.

FRANK–WOLFE ALGORITHM
Input: A convex function $f : \mathbb{R}^n \to \mathbb{R}$, a compact convex set $K \subseteq \mathbf{dom}\, f$, a starting point $x^{(0)} \in K$, and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in K$
Determine the number of iterations T and the step-size $\eta > 0$ based on ε and other parameters.
For $t = 0, 1, 2 \ldots, T-1$ do
  Call the DIRECTION FINDING SUBROUTINE to compute
    $s^{(t)} := \arg\min_{y \in K} \left\langle \nabla f(x^{(t)}), y \right\rangle$;
  $\Delta^{(t)} := s^{(t)} - x^{(t)}$; $\eta_t := \frac{2}{t+2}$;
  $x^{(t+1)} = x^{(t)} + \eta_t \Delta^{(t)}$;
Return $x^{(\text{out})} = x^{(T)}$

5.2.1 Convergence analysis

Let us first verify that the algorithm indeed remains inside K throughout.

Lemma 5.11. *For any step-size $\eta_t \in [0,1]$, we have $x^{(t+1)} \in K$.*

Proof. We can rewrite the update formula as $x^{(t+1)} = (1-\eta_t)x^{(t)} + \eta_t s^{(t)}$. Thus, $x^{(t+1)}$ is on the line segment $[x^{(t)}, s^{(t)}]$, which is entirely inside the convex set K. □

Our next lemma shows that the search direction $\Delta^{(t)}$ also provides a bound on the optimality gap. value.

Lemma 5.12. *Let x^* be the optimal solution to $\min_{x \in K} f(x)$ and $p^* = f(x^*)$. In each iteration of the Frank–Wolfe algorithm,*

$$f(x^{(t)}) - p^* \le -\left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle \,.$$

Proof. Using convexity, we see that

$$\begin{aligned} p^* = f(x^*) &\ge f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x^* - x^{(t)} \right\rangle \\ &\ge f(x^{(t)}) + \min_{y \in K} \left\langle \nabla f(x^{(t)}), y - x^{(t)} \right\rangle \\ &= f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), s^{(t)} - x^{(t)} \right\rangle \\ &= f(x^{(t)}) + \left\langle \nabla f(x^{t}), \Delta^{(t)} \right\rangle \,, \end{aligned}$$

giving the desired bound. □

The above two statements are true for any step-size and for any convex differentiable f. We now give the running time bound for M-smooth functions. Let D denote the diameter of K, that is,

$$D = \max\{\|x - y\| : x, y \in K\}.$$

This is closely related to the quantity R previously used; clearly, $R \leq D$, and in most cases we we would use the same estimates on R and D.

Lemma 5.13. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be M-smooth. In each iteration of the Frank–Wolfe algorithm,*

$$f(x^{(t+1)}) \leq f(x^{(t)}) + \eta_t \left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle + \frac{\eta_t^2 M D^2}{2}.$$

Proof. Recall that $x^{(t+1)} - x^{(t)} = \eta_t \Delta^{(t)}$. According to M-smoothness,

$$\begin{aligned} f(x^{(t+1)}) &= f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x^{(t+1)} - x^{(t)} \right\rangle + D_f(x^{(t+1)}, x^{(t)}) \\ &\leq f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x^{(t+1)} - x^{(t)} \right\rangle + \frac{M}{2} \left\| x^{(t+1)} - x^{(t)} \right\|^2 \\ &= f(x^{(t)}) + \eta_t \left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle + \frac{\eta_t^2 M}{2} \left\| \Delta^{(t)} \right\|^2 \\ &\leq f(x^{(t)}) + \eta_t \left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle + \frac{\eta_t^2 M D^2}{2}. \end{aligned}$$

The last inequality uses that $\Delta^{(t)}$ is the distance between two points in K, and hence, at most D. □

Using Lemma 5.12 and Lemma 5.13, we can derive the overall convergence bound as follows:

Theorem 5.14. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an M-smooth function and $K \subseteq \mathbf{dom}\, f$ a nonempty compact convex set. Assume that the diameter of K is bounded by D. Then, for any $\varepsilon > 0$, the Frank–Wolfe algorithm finds an ε-approximate solution $x^{(t)}$ within*

$$T \geq \frac{2MD^2}{\varepsilon} - 2$$

iterations, using step-size $\eta_t = 2/(t+2)$.

Proof. Let $h_t = f(x^{(t)}) - p^*$ denote the optimality gap. Thus, we can write Lemma 5.12 as $\left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle \leq -h_t$, and Lemma 5.13 implies

$$h_{t+1} \leq h_t + \eta_t \left\langle \nabla f(x^{(t)}), \Delta^{(t)} \right\rangle + \frac{\eta_t^2 M D^2}{2}.$$

Putting these together, we get

$$h_{t+1} \leq (1 - \eta_t)\, h_t + \frac{\eta_t^2 M D^2}{2}. \tag{5.9}$$

The statement of the theorem can be equivalently written as

$$h_t \leq \frac{2MD^2}{t+2}. \tag{5.10}$$

We prove this by induction for any $t \geq 1$ using (5.9). The base case $t = 1$ follows by applying (5.9) for $t = 0$, and noting that $\eta_0 = 1$.

Assume (5.10) holds for t; we now prove it to $t+1$. From (5.9), we get

$$\begin{aligned} h_{t+1} &\leq \left(1 - \frac{2}{t+2}\right) h_t + \frac{2MD^2}{(t+2)^2} \\ &\leq \frac{t}{t+2} \cdot \frac{2MD^2}{t+2} + \frac{2MD^2}{(t+2)^2} \\ &= \frac{t+1}{(t+2)^2} \cdot 2MD^2 \\ &< \frac{2MD^2}{t+3}. \end{aligned}$$

completing the proof. □

Chapter 6

Subgradient and stochastic gradient methods

Algorithms in the previous two chapters were based on two crucial assumptions: *(a)* our convex function $f(x)$ is differentiable, and *(b)* we can easily compute the gradient $\nabla f(x)$. Both assumptions may be violated in important practical scenarios. We may easily encounter non-differentiable functions such as $f(x) = \max\{0, \langle a, x\rangle + b\}$; and the objective function may correspond to error minimisation over a huge dataset in which case computing the gradient can require significant time.

In this chapter, we study methods that work without these assumptions: subgradient methods in Section 6.1 and stochastic gradient descent in Section 6.2. For simplicity of exposition, we mainly focus on the simplest unconstrained settings. In Section 6.1.3, we present the *alternating projections method* and reveal its connection to subgradient descent. In Section 6.3, we present *support vector machines* where the two phenomena naturally occur together.

6.1 Subgradient methods

A common example of a non-differentiable convex function is the univariate $f(x) = |x|$. More generally, functions $f : \mathbb{R}^n \to \mathbb{R}$ of the form $f(x) = \max_{i\in I} \langle a_i, x\rangle + b_i$, i.e. the pointwise maximum of a finite set of affine functions will be non-differentiable at points where the maximum is taken for multiple indices.

These functions are differentiable in "most" points. This can be stated more formally. Let us start with a univariate convex function $f : \mathbb{R} \to \mathbb{R}$. Then, one can show (see class exercises) that left and right derivatives $f'_-(x)$ and $f'_+(x)$ exist at every $x \in \mathbb{R}$, and these are monotone increasing functions. Further, f is non-differentiable at x if and only if $f'_-(x) < f'_+(x)$. This immediately shows that there can only be countable many points where f is not differentiable. A multivariate convex function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable *almost everywhere*, meaning the set of points where f is not differentiable has measure 0.

Nevertheless, points where the function is not differentiable can be critical for our optimisation problem. The concept of *subgradients* can be motivated from the first order characterisation of convexity (Theorem 2.6).

Definition 6.1. *For a function $f : \mathbb{R}^n \to \mathbb{R}$, the vector $g \in \mathbb{R}^n$ is a* subgradient *at $x \in \mathbf{dom}\,(f)$ if*

$$f(y) \geq f(x) + \langle g, y - x\rangle \quad \forall y \in \mathbf{dom}\,(f)\,.$$

We let $\partial f(x) \subseteq \mathbb{R}^n$ denote the set of subgradients at x; this set is also called the subdifferential *at x.*

Note that the definition does not assume convexity of f. For a univariate convex function, $\partial f(x) = [f'_-(x), f'_+(x)]$. For $f(x) = |x|$, we have $\partial f(x) = \{-1\}$ if $x < 0$, $\partial f(x) = \{+1\}$ if $x > 0$, and $\partial f(0) = [-1, 1]$ (see Figure 6.1).

Let us now state the relationship between subgradients, differentiability, and convexity.

Theorem 6.2. *For a function $f : \mathbb{R}^n \to \mathbb{R}$, the following hold:*

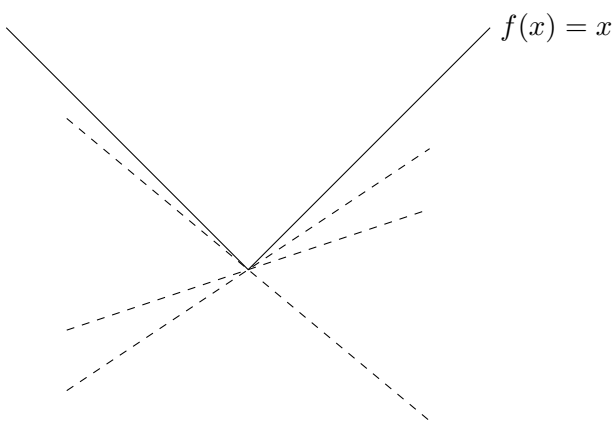


Figure 6.1: Subgradients of $f(x) = x$ at $x = 0$

(a) If f is differentiable, then for every $x \in \mathbf{dom}\,(f)$, either $\partial f(x) = \{\nabla f(x)\}$, or $\partial f(x) = \emptyset$.

(b) f is convex if and only if $\partial f(x) \neq \emptyset$ for every $x \in \mathbf{dom}\,(f)$.

(c) A differentiable function f is convex if and only if $\partial f(x) = \{\nabla f(x)\}$ for every $x \in \mathbf{dom}\, f$.

Proof. **[Non examinable]**
Part (a): Assume for a contradiction that a subgradient $g \in \partial f(x)$, $g \neq \nabla f(x)$ exists; let $h = \nabla f(x) - g$.

We start by showing that $\langle \nabla f(x), g\rangle = \|g\|^2$. Assume first that $\|g\|^2 - \langle \nabla f(x), g\rangle > 0$. Recall from the definition of differentiability (Definition 1.12) that if f is differentiable at x then for every $\delta > 0$ there exists $\varepsilon > 0$ such that $|f(z) - f(x) - \langle \nabla f(x), z - x\rangle| < \delta\|x - z\|$ holds for every $z \in \mathbb{R}^n$ with $\|x - z\| \leq \varepsilon$. Let us select a value $\delta > 0$ such that

$$\delta\|g\| < \|g\|^2 - \langle \nabla f(x), g\rangle \ ,$$

and pick $\varepsilon > 0$ for this choice. Let $\alpha > 0$ such that $\alpha\|g\| \leq \varepsilon$. Then, for $z = x + \alpha g$ we obtain a contradicion from

$$f(x) + \alpha \langle \nabla f(x), g\rangle + \delta\alpha\|g\| > f(z) \geq f(x) + \alpha\|g\|^2 > f(x) + \alpha \langle \nabla f(x), g\rangle + \delta\alpha\|g\| \,,$$

where the first inequality holds by the choice of α and ε; the second by the definition of subgradients, and the third uses the choice of δ. A similar argument works for the case when $\|g\|^2 - \langle \nabla f(x), g\rangle < 0$, using $\alpha < 0$.

Thus, we have shown that $\langle \nabla f(x), g\rangle = \|g\|^2$, and therefore $\langle g, h\rangle = 0$. Below, we will show that $\langle \nabla f(x), h\rangle = 0$. Recalling that $\nabla f(x) = g + h$, we get $\|h\|^2 = \langle \nabla f(x) - g, h\rangle = \langle \nabla f(x), h\rangle - \langle g, h\rangle = 0$, thus, $h = 0$. But this will contradict the choice $g \neq \nabla f(x)$.

Assume for a contradiction $\langle \nabla f(x), h\rangle \neq 0$; let us first assume $\langle \nabla f(x), h\rangle > 0$. For a choice $\delta > 0$ such that $\delta\|g\| < \langle \nabla f(x), h\rangle$, let us select the corresponding $\varepsilon > 0$, and let $z = x - \alpha h$ such that $\alpha\|h\| < \varepsilon$, $\alpha > 0$. Then, we get a contradition from

$$f(x) > f(x) - \alpha \langle \nabla f(x), h\rangle + \delta\alpha\|g\| > f(z) \geq f(x) - \alpha \langle g, h\rangle = f(x) \,.$$

Here, the first inequality is by the choide of δ; the second by the choice of α and ε; the third by the definition of subgradients, and the final equality uses $\langle g, h\rangle = 0$ we have shown above. The case $\langle \nabla f(x), h\rangle < 0$ follows similarly, using $\alpha < 0$.

Part (b): Recall that convexity of f is equivalent to the epigraph being convex; this is the set $K = \{(x, t) \in \mathbb{R}^{n+1} : x \in \mathbf{dom}\, f, f(x) \leq t\}$. For any $x \in \mathbf{dom}\, f$, $(x, f(x))$ is on the boundary of K. According to Theorem 1.5, there exist a supporting hyperplane of K at $(x, f(x))$. This can be written as a vector $(a, \alpha) \in \mathbb{R}^{n+1}$, $(a, \alpha) \neq 0$ such that

$$\langle (a, \alpha), (z, t)\rangle \geq \langle (a, \alpha), (x, f(x))\rangle \quad \forall (z, t) \in K \,,$$

That is, for any point (z,t) such that $z \in \mathbf{dom}\, f$, $t \geq f(z)$, we get

$$\langle a, z\rangle + \alpha t \geq \langle a, x\rangle + \alpha f(x)\,. \tag{6.1}$$

Note that $\alpha < 0$ is impossible, since the inequality must remain true by arbitrary increasing t. Further, $\alpha \neq 0$, as otherwise we must have $\langle a, z\rangle \geq \langle a, x\rangle$ for all $z \in \mathbf{dom}\, f$, which implies $a = 0$, and hence $(a, \alpha) = 0$, a contradiction.

Consequently, $\alpha > 0$. For $g = -a/\alpha$, $t = f(z)$, we can rewrite (6.1) as

$$f(z) \geq f(x) + \langle g, z - x\rangle \quad \forall z \in \mathbf{dom}\, f\,,$$

that is, $g \in \partial f(x)$ is a subgradient.

Part (c): This is immediate from the first two parts. □

6.1.1 The subgradient descent algorithm

The subgradient descent method is an immediate extension of gradient descent: instead of using gradients, in each iteration we pick a subgradient at the current point. We now present the algorithm using iteration-dependent step-sizes η_t.

SUBGRADIENT DESCENT
Input: A convex function $f : \mathbb{R}^n \to \mathbb{R}$, a starting point $x^{(0)} \in \mathbf{dom}\, f$, and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in \mathbf{dom}\, f$
Determine the number of iterations T based on ε and other parameters.
For $t = 0, 1, 2 \ldots, T-1$ do
  Select a subgradient $g^{(t)} \in \partial f(x^{(t)})$;
  Determine the step-size $\eta_t > 0$;
  $x^{(t+1)} = x^{(t)} - \eta_t g^{(t)}$;
Return $x^{(\text{out})} = \arg\min_t f(x^{(t)})$

The basic analysis in Section 4.1 directly extends to replacing gradients by subgradients. In particular, (4.3) turns into

$$f(x^{(t)}) - p^* \leq \frac{\eta_t}{2}\left\|g^{(t)}\right\|^2 + \frac{1}{2\eta_t}\left(\left\|x^{(t)} - x^*\right\|^2 - \left\|x^{(t+1)} - x^*\right\|^2\right)\,. \tag{6.2}$$

Using this, the analysis for Lipschitz-continuous functions directly extend. The following generalisation of Theorem 4.2 holds; we omit the proof.

Theorem 6.3. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz-continuous with parameter L if and only if $\|g\| \leq L$ for every $x \in \mathbf{dom}\, f$ and every subgradient $g \in \partial f(x)$.*

The same argument used for Theorem 4.3 remains valid with $g^{(t)}$ in place of $\nabla f(x^{(t)})$ and leads to the following.

Theorem 6.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an L-Lipschitz-continuous convex function. Assume that a global minimum x^* exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$. Then, for any $\varepsilon > 0$, subgradient descent finds an ε-approximate solution within $T \geq R^2L^2/\varepsilon^2$ iterations, using step-size $\eta = R/(L\sqrt{T})$.*

However, we note that there is no direct analogue of M-smoothness and Theorem 4.7 for subgradient methods: analogouos requirements on subgradients would already imply differentiability.

6.1.2 The Polyak-step-size

[Non examinable]

We now consider a natural subgradient descent method with varying step-size η_t; the same is of course also applicable for gradient descent. The drawback of the fixed step-size $\eta = R/(L\sqrt{T})$ is that it depends on knowing (or guessing) the parameters R and L, and also committing to a number of iterations T in advance.

Let us now consider a scenario where the optimum value $p^* = f(x^*)$ is known. This assumption can be natural in certain cases, as we will see an example in the next section. Knowing p^* gives an obvious stopping criterion: terminate once an iterate with $f(x^{(t)}) \leq p^* + \varepsilon$ is found.

The *Polyak-step-size* at iteration t does not require knowledge of any parameters other than p^*. It is defined as

$$\eta_t := \frac{f(x^{(t)}) - p^*}{\|g^{(t)}\|^2}\,. \tag{6.3}$$

The motivation comes from (6.2) that can be rewritten as

$$\left\|x^{(t+1)} - x^*\right\|^2 \leq \left\|x^{(t)} - x^*\right\|^2 + {\eta_t}^2 \left\|g^{(t)}\right\|^2 - 2\eta_t(f(x^{(t)}) - p^*)\,. \tag{6.4}$$

In order to minimise the distance $\left\|x^{(t+1)} - x^*\right\|$, a natural idea is to minimise the value of the right hand side. This is a convex function in η_t, and the minimum is taken precisely by choosing η_t according to (6.3). With this choice, we obtain the bound

$$\left\|x^{(t+1)} - x^*\right\|^2 \leq \left\|x^{(t)} - x^*\right\|^2 - \frac{\left(f(x^{(t)}) - p^*\right)^2}{\left\|g^{(t)}\right\|^2}\,. \tag{6.5}$$

Adding together these inequalities for $t = 0, 1, 2, \ldots, T-1$, we get

$$\left\|x^{(T)} - x^*\right\|^2 \leq \left\|x^{(0)} - x^*\right\|^2 - \sum_{i=1}^{T-1} \frac{\left(f(x^{(t)}) - p^*\right)^2}{\left\|g^{(t)}\right\|^2}\,. \tag{6.6}$$

We use this bound to prove the same convergence bound for Lipschitz-continuous functions as in Theorem 6.4.

Theorem 6.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an L-Lipschitz-continuous convex function. Assume that a global minimum x^* exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$, and that the optimum value p^* is known. Then, for any $\varepsilon > 0$, subgradient descent with Polyak-step-sizes finds an ε-approximate solution within $T \geq R^2L^2/\varepsilon^2$ iterations.*

Proof. Rearranging (6.6), and using that $\left\|g^{(t)}\right\| \leq L$ by Theorem 6.3, we see that

$$\frac{1}{L^2}\sum_{i=1}^{T-1}\left(f(x^{(t)}) - p^*\right)^2 \leq \left\|x^{(0)} - x^*\right\|^2 - \left\|x^{(T)} - x^*\right\|^2.$$

using the definition of R and the nonnegativity of $\left\|x^{(T)} - x^*\right\|$ we get

$$\frac{1}{T}\sum_{i=1}^{T-1}\left(f(x^{(t)}) - p^*\right)^2 \leq \frac{R^2L^2}{T}\,.$$

If the right hand side is at most ε^2, then the smallest $f(x^{(t)})$ value among the first T iterates is bounded by $p^* + \varepsilon$. Hence, we need to select $T \geq R^2L^2/\varepsilon^2$. □

We note that there are variants of the Polyak-step-size (not covered here) that do not assume knowledge of p^*.

6.1.3 Alternating projections method

[Non examinable]

We now consider the classical *alternating projections method* that turn out to be a special case of subgradient descent with Polyak-step-sizes.

Consider a convex feasibility problem where we are given m closed convex sets $C_1, C_2, \ldots, C_m \subseteq \mathbb{R}^n$, and our goal is to find a feasible point in the intersection

$$K = C_1 \cap C_2 \cap \ldots \cap C_m \,.$$

We assume that all sets C_i are 'simple' in the sense that a fast method for computing the projection mapping $\Pi_{C_i}(y)$ for any point $y \in \mathbb{R}^n$ is available. (Recall the projection mapping from Section 1.1.1). Throughout, we use $\mathrm{dist}(x, C_i) = \|x - \Pi_{C_i}(x)\|$ to denote the distance between x and C_i.

Despite the simplicity of the C_i's, finding a point in the intersection may be significantly harder. As a classical example, assume all C_i's are half-spaces of the form $C_i = \{x \in \mathbb{R}^n : \langle a_i, x\rangle \leq b_i\}$. Then, finding a point in the intersection is the same as the linear programming feasibility problem.

For simplicity of exposition, we only consider here the case $m = 2$, i.e., the intersection of two sets $C_1 \cap C_2$. The alternating projections method starts with an initial point $x^{(0)}$. W.l.o.g. assume $x^{(0)} \notin C_1$; then, we determine $x^{(1)} = \Pi_{C_1}(x^{(0)})$. Now, $x^{(1)} \in C_1$, and if also $x^{(2)} \in C_2$, then we are done. Otherwise, we next project to C_2: $x^{(2)} = \Pi_{C_2}(x^{(1)})$. We continue with alternating projections until we arrive at an iterate $x^{(t)} \in C_1 \cap C_2$, or a required accuracy is reached: we are sufficiently close to both sets. The algorithm is illustrated on Figure 6.1.3 for two circles.

In this context, we say that $x^{(t)}$ is a ε-approximate solution if the distance of $x^{(t)}$ from both C_1 and C_2 is at most ε. We note that this *does not* mean that $x^{(t)}$ is also within distance ε from the intersection $C_1 \cap C_2$ (see exercises).

ALTERNATING PROJECTIONS
Input: Two closed convex sets $C_1, C_2 \subseteq \mathbb{R}^n$, a starting solution $x^{(0)} \in \mathbb{R}^n$ and accuracy requirement $\varepsilon > 0$.
Output: A solution $x^{(\mathrm{out})} \in \mathbf{dom}\, f$ that is within distance ε from both C_1 and C_2
$t = 0$;
While $\max\{\mathrm{dist}(x^{(t)}, C_1), \mathrm{dist}(x^{(t)}, C_2)\} > \varepsilon$ do
  If $\mathrm{dist}(x^{(t)}, C_1) > \mathrm{dist}(x^{(t)}, C_2)$ then $x^{(t+1)} = \Pi_{C_1}(x^{(t)})$;
  Else $x^{(t+1)} = \Pi_{C_2}(x^{(t)})$;
  $t = t + 1$;
Return $x^{(\mathrm{out})} = x^{(t)}$

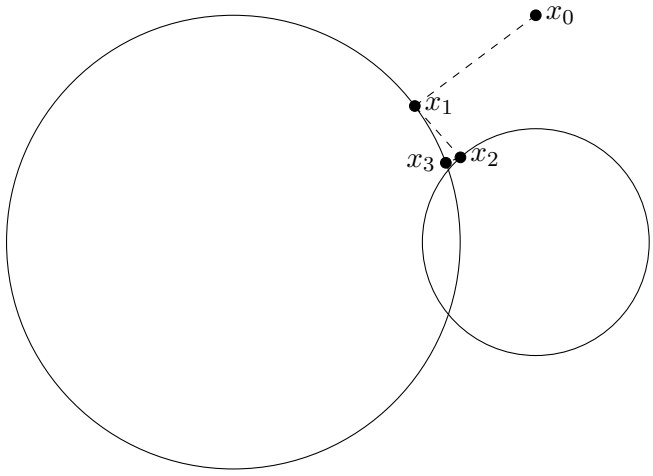


Figure 6.2: Alternating projection for two circles

We now show that the alternating projections method can be interpreted as subgradient descent for minimising the function $f : \mathbb{R}^n \to \mathbb{R}$ defined as

$$f(x) = \max\{\operatorname{dist}(x, C_1), \operatorname{dist}(x, C_2)\} .$$

This is a convex function, since $\operatorname{dist}(x, C_i)$ is a convex function for any closed convex set C_i (see exercises), and the maximum of two convex functions is convex. However, taking the maximum of two differentiable convex functions is not anymore differentiable.

The minimum value of $f(x)$ is 0 if and only if $C_1 \cap C_2 \neq \emptyset$, and every point in the intersection is optimal. If the two sets are disjoint, then the optimum value is $\operatorname{dist}(C_1, C_2)/2$.

Let us assume that the intersection is nonempty, in which case we know $p^* = 0$. Thus, we can use subgradient descent with Polyak-step-sizes. There is a natural choice of subgradients, as shown in the next lemma.

Lemma 6.6. *Assume that in the current iteration, $f(x^{(t)}) = \operatorname{dist}(x^{(t)}, C_i) > 0$ for $i \in \{1, 2\}$. Then,*

$$g^{(t)} := \frac{x^{(t)} - \Pi_{C_i}(x^{(t)})}{\|x^{(t)} - \Pi_{C_i}(x^{(t)})\|}$$

is a subgradient in $\partial g(x^{(t)})$ and $\|g^{(t)}\| = 1$.

The proof can be derived by showing that $g^{(t)} = \nabla h(x^{(t)})$ for $h(x) = \operatorname{dist}(x, C_i)$. We now give a direct proof using the definition of subgradients and Proposition 5.1.

Proof. The statement $\|g^{(t)}\| = 1$ is immediate. We need to verify that for any $y \in \mathbb{R}^n$,

$$f(y) \geq \left\langle g^{(t)}, y - x^{(t)} \right\rangle .$$

We can lower bound $f(y) \geq \operatorname{dist}(y, C_i) = \|y - \Pi_{C_i}(y)\|$. Rearranging, it suffices to show

$$\|y - \Pi_{C_i}(y)\| \cdot \left\|x^{(t)} - \Pi_{C_i}(x^{(t)})\right\| \geq \left\langle x^{(t)} - \Pi_{C_i}(x^{(t)}), y - x^{(t)} \right\rangle .$$

Let us add $\left\langle x^{(t)} - \Pi_{C_i}(x^{(t)}), x^{(t)} - \Pi_{C_i}(x^{(t)}) - y + \Pi_{C_i}(y) \right\rangle$ to both sides. Then, this is equivalent to

$$\left\|x^{(t)} - \Pi_{C_i}(x^{(t)})\right\|^2 + \|y - \Pi_{C_i}(y)\| \cdot \left\|x^{(t)} - \Pi_{C_i}(x^{(t)})\right\| - \left\langle x^{(t)} - \Pi_{C_i}(x^{(t)}), y - \Pi_{C_i}(y) \right\rangle \geq \left\langle x^{(t)} - \Pi_{C_i}(x^{(t)}), \Pi_{C_i}(y) - \Pi_{C_i}(x^{(t)}) \right\rangle .$$

The left hand side is nonnegative by the Cauchy–Schwarz inequality, and the right hand side is nonpositive by Proposition 5.1, noting that $\Pi_{C_i}(y) \in C_i$. □

Let us now consider the subgradient descent update for an iterate $x^{(t)}$ where

$$f(x^{(t)}) = \max\{\operatorname{dist}(x^{(t)}, C_1), \operatorname{dist}(x^{(t)}, C_2)\} = \operatorname{dist}(x^{(t)}, C_i)$$

with the subgradient $g^{(t)}$ as in Lemma 6.6 and the Polyak-step-size, recalling that $p^* = 0$ by our assumption $C_1 \cap C_2 \neq \emptyset$. We have $f(x^{(t)}) - p^* = \operatorname{dist}(x^{(t)}, C_i) - 0 = \|x^{(t)} - \Pi_{C_i}(x^{(t)})\|$ and $\|g^{(t)}\| = 1$. Thus,

$$\begin{aligned} x^{(t+1)} &= x^{(t)} - \eta_t g^{(t)} = x^{(t)} - \|x^{(t)} - \Pi_{C_i}(x^{(t)})\| \frac{x^{(t)} - \Pi_{C_i}(x^{(t)})}{\|x^{(t)} - \Pi_{C_i}(x^{(t)})\|} \\ &= \Pi_{C_i}(x^{(t)}) , \end{aligned}$$

confirming that the subgradient descent steps follow the same sequence of iterations as alternating projection.

We can therefore use Theorem 6.5 to bound the number of iterations of the alternating projections algorithm, with constant $L = 1$. Hence if $C_1 \cap C_2 \neq \emptyset$, then within R^2/ε^2 iterations we can obtain a solution $x^{(t)}$ with $\max\{\operatorname{dist}(x^{(t)}, C_1), \operatorname{dist}(x^{(t)}, C_2)\} \leq \varepsilon$.

6.2 Stochastic gradient descent

Regression problems play a central role in machine learning. The general scheme (that includes linear and logistic regression, and many more) involves a dataset of m data points with k dimensional feature vectors $a_i \in \mathbb{R}^k$ and dependent variables $b_j \in \mathbb{R}$. We have a parametric model that is described by a parameter vector $x \in \mathbb{R}^n$. The model is represented by a function $F : \mathbb{R}^k \times \mathbb{R}^n \to \mathbb{R}$ such that $F(a, x)$ is the outcome for the feature vector $a \in \mathbb{R}^k$ with parameters $x \in \mathbb{R}^n$. Further, we have a loss function $L : \mathbb{R}^2 \to \mathbb{R}$ such that $L(b, t) \geq 0$ is the penalty for the outcome t when the true dependent variable is b; we have $L(b, t) = 0$ if $b = t$.

The objective function is then defined as $f : \mathbb{R}^n \to \mathbb{R}$ as

$$f(x) = \frac{1}{m} \sum_{j=1}^{m} L(b_j, F(a_j, x)) + R(x)\,, \tag{6.7}$$

where $R(x) : \mathbb{R}^n \to \mathbb{R}$ is a convex regulariser function. The first term is the total loss (taking 0 if $F(a_j, x) = b_j$ for each data point), and the second term can express additional constraints on the parameters 0, such as the penalties in the Ridge and Lasso regressions.

To simplify the notation, let $f_j(x) = L(b_j, F(a_j, x)) + R(x)$, that is,

$$f(x) = \frac{1}{m} \sum_{j=1}^{m} f_j(x)\,.$$

Even if $f(x)$ is differentiable, computing the gradient involves computing m gradients $\nabla f_j(x)$ that can be prohibitive for a large value of m.

Stochastic gradient descent (SGD) is a *randomised* algorithm that, at each iteration, picks an index $1 \leq j \leq m$ uniformly at random, and moves opposite the direction of

$$g^{(t)} = \nabla f_j(x^{(t)})\,.$$

STOCHASTIC GRADIENT DESCENT
Input: A differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$, a starting point $x^{(0)} \in \mathbf{dom}\, f$, and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in \mathbf{dom}\, f$
Determine the number of iterations T and the step-size $\eta > 0$ based on ε and other parameters.
For $t = 0, 1, 2 \ldots, T-1$ do
  Select an index $j \in \{1, \ldots, m\}$ uniformly at random ;
  $g^{(t)} = \nabla f_j(x^{(t)})$;
  $x^{(t+1)} = x^{(t)} - \eta g^{(t)}$;
Return $x^{(\text{out})} = \arg\min_t f(x^{(t)})$

The algorithm itself is very similar to the standard gradient descent method. However, we have to be more careful with the analysis, since all $x^{(t)}$ and $g^{(t)}$ vectors are random variables that depend on the sequence of random choices. Overall, the algorithm makes T^m random decisions, which is a finite (if astronomical) value. Let Ω denote the finite set of possible sequences $\{x^{(t)} : t = 1, \ldots, T\}$ we may encounter. (Having a finite domain simplifies the probabilistic arguments.) Before the analysis, we recall some basic concepts from probability theory.

Review on conditional expectations Consider a *finite probability space* $(\Omega, \mathbb{P})$, where Ω is a finite set, and $\mathbb{P} : 2^\Omega \to [0, 1]$ is a probability function. An *event* is any subset $A \subseteq \Omega$. For events A and B, the *conditional probability* $\mathbb{P}(A|\, B)$ is defined as

$$\mathbb{P}(A|\, B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}\,.$$

Let $Y : \Omega \to \mathbb{R}^n$ be a (multi-valued) random variable. We let $Y(\Omega) \subseteq \mathbb{R}^n$ denote the value range of Ω, and we let $Y = y$ denote the event $\{\omega \in \Omega : Y(\omega) = y\}$ Then, the *expected value* of Y is

$$\mathbb{E}[Y] = \sum_{y \in Y(\Omega)} y \cdot \mathbb{P}(y = Y),$$

whereas the *conditional expectation* of Y over an event B is

$$\mathbb{E}[Y \,|\, B] = \sum_{y \in Y(\Omega)} y \cdot \mathbb{P}(y = Y \,|\, B).$$

The (conditional) expectation is well-known to be linear. For random variables $Y_1, \ldots, Y_m$ and coefficients $\lambda_1, \lambda_2, \ldots, \lambda_m$, we have

$$\mathbb{E}\left[\sum_{j=1}^{m} \lambda_j Y_j \,\middle|\, B\right] = \sum_{j=1}^{m} \lambda_j \mathbb{E}[Y_j \,|\, B].$$

6.2.1 Analysis of the stochastic gradient method

First, consider the conditional expectation of $g^{(t)}$, given a value $x^{(t)} = z$ (recall that $x^{(t)}$ is itself a random variable). By linearity of conditional expectations,

$$\mathbb{E}\left[g^{(t)} \,\middle|\, x^{(t)} = z\right] = \sum_{j=1}^{m} \frac{1}{m} \nabla f_j(z) = \nabla f(z).$$

Hence, the expected value of the direction $g^{(t)}$ equals the gradient of the current iterate. The difficulty arises since the sampled vector $g^{(t)}$ may not be a subgradient, thus, already the first step of the basic analysis, (4.2) may fail: we don't necessarily have

$$f(x^{(t)}) - p^* \leq \left\langle g^{(t)}, x^{(t)} - x^* \right\rangle. \tag{6.8}$$

Nevertheless, we show that this holds in expectation. Again by linearity,

$$\begin{aligned}\mathbb{E}\left[\left\langle g^{(t)}, x^{(t)} - x^* \right\rangle \,\middle|\, x^{(t)} = z\right] &= \mathbb{E}\left[\left\langle g^{(t)}, z - x^* \right\rangle \,\middle|\, x^{(t)} = z\right] = \left\langle \mathbb{E}\left[g^{(t)} \,\middle|\, x^{(t)} = z\right], z - x^* \right\rangle \\ &= \langle \nabla f(z), z - x^* \rangle.\end{aligned}$$

From here, we can derive the key equality

$$\mathbb{E}\left[\left\langle g^{(t)}, x^{(t)} - x^* \right\rangle\right] = \mathbb{E}\left[\langle \nabla f(x_t), x_t - x^* \rangle\right], \tag{6.9}$$

by noting that both these expressions equal

$$\sum_{z \in x^{(t)}(\Omega)} \langle \nabla f(z), z - x^* \rangle \cdot \mathbb{P}[x^{(t)} = z].$$

From these arguments, we can recover the analogue of (4.2) *in expectation*:

$$\mathbb{E}\left[f(x^{(t)})\right] - p^* \leq \mathbb{E}\left[\left\langle g^{(t)}, x^{(t)} - x^* \right\rangle\right]. \tag{6.10}$$

Using this, we can recover all convergence bounds from Chapter 4 *in expectation*. We only state here the bound analogous to Theorem 4.3. However, we need to modify the Lipschitz-condition that assumes $\|\nabla f(x)\| \leq L$ for all $x \in \mathbf{dom}\, f$. Instead, we need to assume that $\mathbb{E}\left[\|g^{(t)}\|^2\right] \leq L^2$.

Theorem 6.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function given in the form $f = \frac{1}{m}\sum_{i=1}^m f_i$. Assume that a global minimum x^* exists, and that $\|x^{(0)} - x^*\| \leq R$ holds for the initial point $x^{(0)}$. Further, assume that $\mathbb{E}\left[\|g^{(t)}\|^2\right] \leq L^2$ in the stochastic gradient descent method. Then, for any $\varepsilon > 0$, in $T \geq R^2L^2/\varepsilon^2$ iterations, using step-size $\eta = R/(L\sqrt{T})$, stochastic gradient descent finds a sequence of solutions such that*

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{E}\left[f(x^{(t)})\right] \leq p^* + \varepsilon\,.$$

Similar extensions can be given for the M-smooth and well-conditioned settings, as well as for conditional stochastic gradient descent and stochastic subgradient descent.

6.2.2 Mini-batch stochastic gradient descent

We emphasise the probabilistic nature of the guarantee in Theorem 6.7: an actual run of the algorithm may return a worse solution. A common approach to mitigate this is *mini-batch SGD*: take $k > 1$ samples $g^{(t,\ell)}$, $\ell = 1, \ldots, k$, and use

$$g^{(t)} = \frac{1}{k}\sum_{\ell=1}^{k} g^{(t,\ell)}\,.$$

As usual, the variance of $g^{(t)}$ (conditioned on $x^{(t)}$) is reduced by a factor k if estimating from k samples.

The k different gradients $\nabla f_j(x^{(t)})$ can be computed in parallel. Hence, mini-batch SGD can be implemented without increase in the computation time using a parallel architecture. Note that k is typically still much smaller than m.

6.3 Support vector machines

We now review *Support vector machines (SVM)*, a classical machine learning method that amounts to an optimisation problem of the form (6.7) with a non-differentiable objective.

We consider a *binary classification* problem as in logistic regression (see Section 2.3.3). To simplify the formalism, the target variable will be $b_j \in \{+1, -1\}$ instead of 0 and 1. The feature vectors are $a_j \in \mathbb{R}^n$, $j = 1, 2, \ldots, m$, including the bias term 1 (thus, there are $n-1$ features). Recall that logistic regression returns a vector $w \in \mathbb{R}^n$ and probabilities

$$p_w(a_j) = \frac{1}{1 + e^{-\langle a_j, w\rangle}}\,.$$

Probability 1 would correspond to a certain $+1$ and probability 0 to a certain -1 answer (however, these values cannot be taken). To turn logistic regression to a binary answer, we need a *decision boundary*. For a fixed threshold $\gamma \in (0, 1)$, say $\gamma = 0.5$, we return $+1$ if $p_w(a_j) \geq \gamma$ and -1 if $p_w(a_j) < \gamma$. The inequality $p_w(a_j) \geq \gamma$ can be equivalently written as

$$\langle a_j, w\rangle \geq \log\frac{\gamma}{1-\gamma}\,,$$

which corresponds to a linear half-space. The larger $p_w(a_j) > \gamma$, the further the point lies from the boundary. The loss function in logistic regression is smaller the further we are on the (correct) side of the decision boundary.

Support vector machines (SVM) give an alternative approach to finding a linear classification boundary. Let $P = \{j \in [m] : b_j = 1\}$ and $N = \{j \in [m] : b_j = -1\}$ denote the parts of the dataset with positive and negative target values. To motivate SVM, first assume there is a perfect linear separation between the sets P and N. Such a separation would amount to a vector $w \in \mathbb{R}^n$ and $\delta \in \mathbb{R}$ such that

$$\langle a_j, w\rangle > \delta \quad \forall j \in P \quad \text{and} \quad \langle a_j, w\rangle < \delta \quad \forall j \in N\,.$$

Recall that the dataset includes a bias term $a_{j1} = 1$; for this reason, we can assume without loss of generality that $\delta = 0$, by subtracting δ from w_1. More concisely, we can write this as

$$b_j \langle a_j, w \rangle > 0 \quad \forall j = 1, 2, \ldots, m\,.$$

Hence, finding a perfect separation corresponds to a (strict) linear feasibility problem. If feasible, it has infinitely many different solutions. The *hard-margin support vector machine* selects a separation where the margin (the distance from the hyperplane on both sides) is as large as possible. This can be obtained by the following nonlinear optimisation problem:

$$\begin{aligned} &\max\ M \\ &\text{subject to}\ \ \|w\| = 1, \\ &\qquad b_j \langle a_j, w \rangle \geq M \quad \forall j = 1, 2, \ldots, m\,. \end{aligned} \tag{6.11}$$

The normalisation $\|w\| = 1$ ensures that for each data point j, the expression on the left hand side measures the distance from the hyperplane. We now turn this into a convex quadratic optimisation problem; the transformation amounts to replacing w by w/M. This second form is

$$\begin{aligned} &\min\ \|w\|^2 \\ &\text{subject to}\ \ b_j \langle a_j, w \rangle \geq 1 \quad \forall j = 1, 2, \ldots, m\,. \end{aligned} \tag{6.12}$$

Note that in case no perfect separation exists, this problem is infeasible. Even in case a separation exists, the solution could be distorted by a few outliers. To get a more robust separation, we allow for (but penalise) violation to the inequalities; this leads to the *soft-margin support vector machine*. If not specified otherwise, SVM will refer to this variant.

Penalisation is done using the *hinge loss function*

$$L(b_j, t) = \max\{0, 1 - b_j t\}\,.$$

Thus, $L(b_j, \langle a_j, w \rangle) > 0$ if the corresponding inequality is violated, and in that case the loss function measures the violation. In contrast to the logistic loss function, as long as $b_j \langle a_j, w \rangle \geq 1$, the loss is zero. Hence, there is no additional reward for being outside this margin.

We are ready to formulate the soft-margin SVM optimisation problem. For a parameter $\lambda > 0$, we have

$$\min\ \frac{1}{m} \sum_{j=1}^{m} \max\{0, 1 - b_j \langle a_j, w \rangle\} + \lambda \|w\|^2\,. \tag{6.13}$$

This is an unconstrained optimisation problem with a nondifferentiable objective function. Note that the distance of a_j from the boundary hyperplane $\langle w, x \rangle = 0$ is $\langle w, a_j \rangle / \|w\|$. Hence, the classifier has a *margin* $1/\|w\|$: points further than $1/\|w\|$ have penalty 0, and points within this margin (or on the wrong side) will be penalised; see Figure 6.3.

The parameter λ represents a 'tolerance' for errors: $\lambda = 0$ means that our only priority is avoiding errors; the objective value 0 is attained for any perfect separation. In contrast, $\lambda \to \infty$ corresponds to classifiers with larger margins but more violations.

The standard approach to solve SVM is stochastic subgradient descent. The loss function $L(b_j, \langle a_j, w \rangle)$ is $\|a_j\|$-Lipschitz; hence, the first term is $\frac{1}{m} \sum_{j=1}^{m} \|a_j\|$-Lipschitz in expectation. The term $\lambda \|w\|^2$ does not have a bounded Lipschitz-constant on $\mathbb{R}^n$, but has Lipschitz constant $2\lambda R$ if restricted to a domain $\|w\| \leq R$. A common improvement is to replace the hinge loss function by a smooth loss function that leads to better convergence guarantees.

Support vectors Given an optimal solution to the support vector classifier, we can arbitrarily add or remove points on the correct side at distance more than $1/\|w\|$: the optimal solution remains unchanged. The only "critical" instances are those within the margin or on the wrong side: these are called the *support vectors*: these are the observations that will contribute to the separating hyperplane.

The motivation for the term and the distinguished role of support vectors can be understood through the lens of Lagrangian duality; we will discuss this in more details in class.

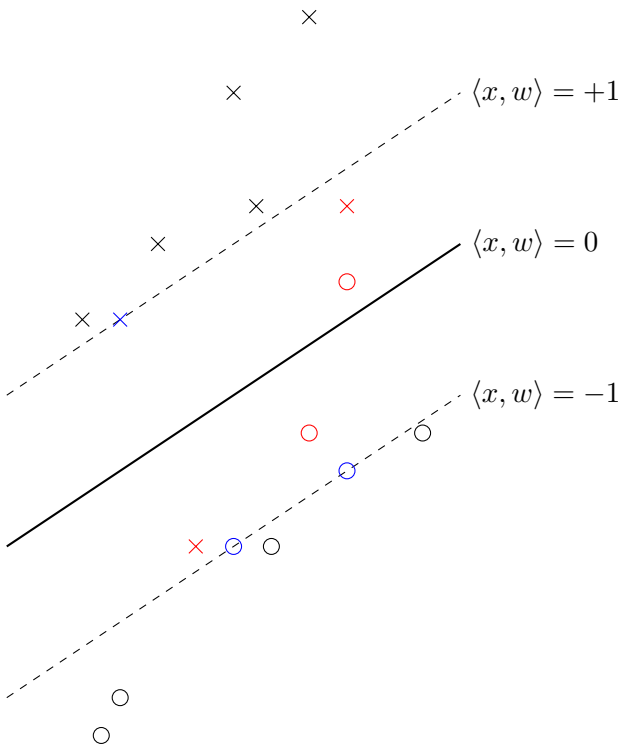


Figure 6.3: Support Vector Machines: The thick line is the classification boundary, the dashed lines show the margins. Crosses denote $+1$ and circles -1 instances. The three support vectors on the margin are shown in blue; the red instances are support vectors on the wrong side of the margin.

Chapter 7

Mirror descent

A salient feature of gradient methods is that the convergence guarantees do not depend on the dimension n of the space, just on parameters such as L or M, and R. However, these parameters may hide implicit dependence on n. Consider a function f that is H-Lipschitz in ℓ_1-norm: $|f(x)-f(y)| \leq H\|x-y\|_1$ for all $x, y \in \mathbf{dom}\, f$, or equivalently (as we will shall see in Theorem 7.5), $\|\nabla f(x)\|_\infty \leq H$ for all $x \in \mathbf{dom}\, f$ for a constant H. Since $\|z\|_2 \leq \sqrt{n}\|z\|_\infty$ for any $z \in \mathbb{R}^n$, such a function is L-Lipschitz in ℓ_2-norm for $L = \sqrt{n}H$. Then, the standard gradient descent method finds a ε-approximate solution in nH^2R^2/ε^2 iterates, a bound that scales linearly with n.

The choice of the norm can thus greatly change the Lipschitz and smoothness properties; however, standard gradient descent is geared for ℓ_2-norms. In Section 7.1, we present mirror descent, a generalisation of gradient descent, that allows for much additional flexibility and can be adapted for different norms. We do not analyse the general method, but focus more closely on a particular instantiation, *exponentiated gradient descent* in Section 7.2.

7.1 The mirror descent framework

As a motivation, let us revisit the proof of Lemma 4.6 that bounds the function value decrement in gradient descent for M-smooth functions. The choice choice $x^{(t+1)} = x^{(t)} - \frac{1}{M}\nabla f(x^{(t)})$ can be justified by bounding

$$\begin{aligned} f(x^{(t+1)}) &= f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x^{(t+1)} - x^{(t)} \right\rangle + D_f(x^{(t+1)}, x^{(t)}) \\ &\leq f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x^{(t+1)} - x^{(t)} \right\rangle + \frac{M}{2}\|x^{(t+1)} - x^{(t)}\|^2 \end{aligned}$$

Denoting $v = x^{(t+1)} - x^{(t)}$, we can observe that $v = -\frac{1}{M}\nabla f(x^{(t)})$ is the minimizer of the convex quadratic expression

$$\min_{v \in \mathbb{R}^n} \left\langle \nabla f(x^{(t)}), v \right\rangle + \frac{M}{2}\|v\|^2 .$$

Noting that for $\Phi(x) = \frac{1}{2}\|x\|^2$, the Bregman-divergence is $D_\Phi(x, y) = \frac{1}{2}\|x-y\|^2$, and setting $\eta = 1/M$, we can also write this as

$$x^{(t+1)} = \arg \min_{x \in \mathbf{dom}\, f} \left\langle \nabla f(x^{(t)}), x - x^{(t)} \right\rangle + \frac{1}{\eta} D_\Phi(x, x^{(t)}) . \tag{7.1}$$

The *mirror-descent algorithm* uses a *mirror map* $\Phi : \mathbb{R}^n \to \mathbb{R}$, a convex function that may be different from the standard choice $\frac{1}{2}\|x\|^2$, and computes the updates using (7.1). We now describe it for constrained convex minimisation, generalising the projected gradient method. When computing the projection step, we pick the next iterate as the point $x \in K$ that minimises the Bregman-divergence $D_\Phi(x, y^{(t+1)})$. The Bregman-divergence plays the role of a distance, even though it is not a metric as it is not symmetric.

MIRROR DESCENT
Input: A convex function $f : \mathbb{R}^n \to \mathbb{R}$, a mirror map $\Phi : \mathbb{R}^n \to \mathbb{R}$ a nonempty closed convex set $K \subseteq \mathbf{dom}\, f$, a starting point $x^{(0)} \in K$, and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in K$
Determine the number of iterations T and the step-size $\eta > 0$ based on ε and other parameters.
For $t = 0, 1, 2 \ldots, T-1$ do
$y^{(t+1)} = \arg\min_{y \in \mathbf{dom}\, f} \left\langle \nabla f(x^{(t)}), y - x^{(t)} \right\rangle + \frac{1}{\eta} D_\Phi(y, x^{(t)})$;
$x^{(t+1)} = \arg\min_{x \in K} D_\Phi(x, y^{(t+1)})$;
Return $x^{(\text{out})} = \arg\min_t f(x^{(t)})$

Note that in case the two functions coincide: $f = \Phi$, then the solution of (7.1) for $\eta = 1$ gives the optimal solution $x^{(t+1)} = x^*$ in a single iteration, since the right hand side expression equals $f(x) - f(x^{(t)})$. The choice of the mirror map Φ needs to balance two requirements:

- First, one should be able to efficiently solve (7.1). Using $\Phi = f$, this problem would be just as hard as the original optimisation problem we are trying to solve.

- Second, $D_\Phi(x, y)$ should be in some way related to $D_f(x, y)$ so that (7.1) is meaningful for determining the next iterate. For example, in the case of M-smooth functions, $D_f(x, y) \leq M D_\Phi(x, y)$ for $\Phi(x) = \frac{1}{2}\|x\|^2$.

A sufficient condition for convergence is given in the next theorem; we do not include the proof. Here, $\|.\|$ denotes any norm instead of the usual ℓ_2-norm.

Theorem 7.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function, $K \subseteq \mathbf{dom}\, f$ a convex set, and let x^* be a minimiser of f in K. Let $\Phi : \mathbb{R}^n \to \mathbb{R}$ be a mirror map, and $\|.\| : \mathbb{R}^n \to \mathbb{R}$ a norm such that the following are satisfied:*

(i) $\mathbf{dom}\, f \subseteq \mathbf{dom}\, \Phi$ *and* $\nabla\Phi : \mathbf{dom}\, \Phi \to \mathbb{R}^n$ *is a bijective map.*

(ii) f is L-Lipschitz with respect to $\|.\|$, that is, $|f(x) - f(y)| \leq L\|x - y\|$ for all $x, y \in \mathbf{dom}\, f$.

(iii) Φ is σ-strongly convex with respect to $\|.\|$ for some $\sigma > 0$, that is, $D_\Phi(x, y) \geq \frac{\sigma}{2}\|x - y\|^2$.

Then, for a suitably chosen step-size, Mirror Descent finds an ε-approximate solution if

$$T \geq C \cdot \frac{L^2 D_\Phi(x^{(0)}, x^*)}{\sigma \varepsilon^2}$$

for some constant $C > 0$.

The first is a technical requirement asserting that every vector $v \in \mathbb{R}^n$ appears as $v = \nabla R(x)$ for some $x \in \mathbf{dom}\, \Phi$. E.g., for $\Phi(x) = \frac{1}{2}\|x\|^2$, this holds because of $\nabla\Phi(x) = x$. For this choice of Φ and for the Euclidean norm $\|.\| = \|.\|_2$, we have $\sigma = 1$ and recover the standard projected gradient method for Lipschitz function as a special case as in Theorem 5.5. Note that $D_\Phi(x, y) = \frac{1}{2}\|x - y\|^2$, and therefore $D_\Phi(x^{(0)}, x^*) \leq \frac{1}{2}R^2$.

In case our function is L-Lipschitz not in ℓ_2-norm but some different norm, one needs to find a suitable mirror map Φ that is strongly convex in the same norm. We stress that a key requirement is that we can efficiently compute the update defining $y^{(t+1)}$ that also depends on the choice of Φ. We will an interesting example in Section 7.2. Before that, let us take a closer look at L-Lipschitzness in different norms.

7.1.1 Dual norms

To characterise L-Lipschitzness in an arbitrary norm $\|.\|$, we introduce the notion of *dual norms*.

Definition 7.2. *Given a norm $\|.\| : \mathbb{R}^n \to \mathbb{R}$, the* dual norm $\|.\|^\star : \mathbb{R}^n \to \mathbb{R}$ *is defined as*

$$\|v\|^\star = \sup\{\langle v, z\rangle : z \in \mathbb{R}^n, \|z\| \leq 1\}\,.$$

One can show that the dual of the dual norm is the original norm. An immediate corollary of Hölder's inequality (Theorem 5.10) provides the duals of ℓ_p-norms:

Lemma 7.3. *Let $p, q \in [1, \infty]$ such that $1/p + 1/q = 1$. Then, $\|v\|_p^\star = \|v\|_q$.*

This in particular tells us that the ℓ_2-norm is its own dual norm, and ℓ_1 and ℓ_∞-norms are dual to each other. The following generalisation of the Cauchy-Schwarz inequality follows from the definition of dual norms.

Theorem 7.4. *For any norm $\|.\|$ and its dual norm $\|.\|^\star$, we have*

$$|\langle u, v\rangle| \leq \|u\| \cdot \|v\|^\star\,.$$

Using this in place of the usual Cauchy-Schwarz inequality, the proof of Theorem 4.2 can be easily extended to obtain the following bound for general norms.

Theorem 7.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and $\|.\| : \mathbb{R}^n \to \mathbb{R}$ any norm. Then, f is Lipschitz-continuous with parameter L in norm $\|.\|$ if and only if $\|\nabla f(x)\|^\star \leq L$ for every $x \in \mathbf{dom}\, f$.*

7.2 Exponentiated gradient descent

We now describe a special instantiation of the mirror descent framework for the following natural setting. Let

$$K = \Delta_n = \{x \in \mathbb{R}^n_+ : \sum x_i = 1\}$$

be the *probability simplex*, i.e., the set of all discrete probability distributions over n elements. Our goal is to compute $\min_{x\in\Delta_n} f(x)$ for a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with $\mathbf{dom}\, f \subseteq \mathbb{R}^n_+$ that is L-Lipschitz on Δ_n in ℓ_1-norm; according to Theorem 7.5, this is equivalent to $\|\nabla f(x)\|_\infty \leq L$ for all $x \in \Delta_n$. A suitable mirror map is the *negative entropy function*:

$$\Phi(x) = \sum_{i=1}^n x_i \log x_i\,,$$

defined on the domain $\mathbf{dom}\,\Phi = \mathbb{R}^n_+$; we use the convention $0 \log 0 = 0$ consistently with $\lim_{t\to 0} t \log t = 0$. We have $\nabla\Phi(x) = \log(x) + \mathbf{1}_n$, where $\log(x) \in \mathbb{R}^n$ denotes the vector with i-th component $\log x_i$, and $\mathbf{1}_n$ is the n-dimensional all ones vector. Thus, $\nabla\Phi$ is a bijection between $\mathbf{dom}\,\Phi$ and $\mathbb{R}^n$, as required in Theorem 7.1. The Bregman-divergence is

$$D_\Phi(x, y) = \sum_{i=1}^n x_i \log\left(\frac{x_i}{y_i}\right) + \sum_{i=1}^n (y_i - x_i)$$

In case $x, y \in \Delta_n$ are probability distributions, the second sum is 0, and therefore $D_\Phi(x, y) = \sum_{i=1}^n x_i \log\left(\frac{x_i}{y_i}\right)$ is the *relative entropy*, also called the *Kullback-Leibler-divergence*, or in short *KL-divergence*. We state (without proof) the strong-convexity property in ℓ_1-norm:

Theorem 7.6 (Pinsker's inequality)**.** *For every two probability distributions $x, y \in \Delta_n$,*

$$D_\Phi(x, y) \geq \frac{1}{2}\|x - y\|_1^2\,.$$

We now explicitly compute the mirror descent updates for mirror descent using negative entropy as the mirror map, also called the *exponentiated gradient descent algorithm*, and derive the convergence bound for this case.

Lemma 7.7. *For a given step-size $\eta > 0$ and using the negative entropy as the mirror map and for $g^{(t)} = \nabla f(x^{(t)})$, the mirror descent update is*

$$y_i^{(t+1)} = x_i^{(t)} \exp\left(-\eta g_i^{(t)}\right) \quad \forall i = 1, 2, \ldots, n \quad \text{and} \quad x^{(t+1)} = \frac{y^{(t+1)}}{\|y^{(t+1)}\|_1}\,.$$

Proof. The update rule can be written as

$$\begin{aligned} y^{(t+1)} &= \arg\min_{y \in \mathbf{dom}\, f} \left\langle g^{(t)}, y - x^{(t)} \right\rangle + \frac{1}{\eta} \sum_{i=1}^{n} y_i \log\left(\frac{y_i}{x_i^{(t)}}\right) - \sum_{i=1}^{n} y_i\,, \\ x^{(t+1)} &= \arg\min_{x \in \Delta_n} D_\Phi(x, y)\,. \end{aligned}$$

here, we simplified by removing the term $\sum_i x_i^{(t)}$ in the first equation as it is always equal to 1. The expression defining $y^{(t+1)}$ is convex in y; setting the gradient to 0 yields

$$\log y_i = -\eta g_i^{(t)} + \log x_i^{(t)} \quad \forall i = 1, 2, \ldots, n\,,$$

giving the claimed expression on y.

Next, we need to find $x \in \Delta_n$ that has minimal Bregman-divergence from $y = y^{(t+1)}$. Note that $y \geq 0$ and consequently $\|y\|_1 = \sum_{i=1}^n y_i$. For the claimed $x = \frac{y}{\|y\|_1}$, we have

$$D_\Phi(x, y) = \|y\|_1 - 1 + \sum_{i=1}^{n} \frac{y_i}{\|y\|_1} \log\left(\frac{1}{\|y\|_1}\right) = \|y\|_1 - 1 - \log \|y\|_1\,.$$

Using the convexity of $h(t) = t \log t$, we show that this lower bound holds for every $x \in \Delta_n$. Indeed, using the coefficients $\lambda_i = y_i / \|y\|_1$ that sum to 1,

$$\begin{aligned} D_\Phi(x, y) &= \|y\|_1 - 1 + \|y\|_1 \sum_{i=1}^{n} \frac{y_i}{\|y\|_1} \cdot \frac{x_i}{y_i} \log\left(\frac{x_i}{y_i}\right) \\ &\geq \|y\|_1 - 1 + \|y\|_1 \left(\sum_{i=1}^{n} \frac{y_i}{\|y\|_1} \cdot \frac{x_i}{y_i}\right) \log\left(\sum_{i=1}^{n} \frac{y_i}{\|y\|_1} \cdot \frac{x_i}{y_i}\right) \\ &= \|y\|_1 - 1 + \log\left(\frac{1}{\|y\|_1}\right) \\ &= \|y\|_1 - 1 - \log \|y\|_1\,, \end{aligned}$$

as required. □

With these formulas, we can write Exponentiated Gradient Descent explicitly. The initial point here is set to $x^{(0)} = \frac{1}{n}\mathbf{1}_n$.

EXPONENTIATED GRADIENT DESCENT
Input: A convex function $f : \mathbb{R}^n \to \mathbb{R}$ and accuracy requirement $\varepsilon > 0$.
Output: A ε-approximate solution $x^{(\text{out})} \in \Delta_n$
Determine the number of iterations T and the step-size $\eta > 0$ based on ε and other parameters.
$x^{(0)} = \frac{1}{n}\mathbf{1}_n$. For $t = 0, 1, 2 \ldots, T-1$ do
  $g^{(t)} = \nabla f(x^{(t)})$;
  For $i = 1, 2 \ldots, n$ do $y_i^{(t+1)} = x_i^{(t)} \exp\left(-\eta g_i^{(t)}\right)$;
  $x^{(t+1)} = \frac{y^{(t+1)}}{\|y^{(t+1)}\|_1}$;
Return $x^{(\text{out})} = \arg\min_t f(x^{(t)})$

7.2.1 Analysis of the algorithm

We first show two lemmas on Bregman-divergences; these hold for general mirror maps Φ.

Lemma 7.8 (Law of cosines for Bregman-divergence)**. [Non examinable]** *For any convex function $\Phi : \mathbb{R}^n \to \mathbb{R}$ and any $x, y, z \in \mathbf{dom}\,\Phi$, we have*

$$\langle \nabla\Phi(y) - \nabla\Phi(z), y - x\rangle = D_\Phi(x, y) + D_\Phi(y, z) - D_\Phi(x, z)$$

The proof follows by substituting the definition of the three Bregman-divergences. For $\Phi(x) = \frac{1}{2}\|x\|^2$ we have $D_\Phi(x, y) = \frac{1}{2}\|x - y\|^2$; the statement is equivalent to the *law of cosines*:

$$\langle y - z, y - x\rangle = \frac{1}{2}\left(\|x - y\|^2 + \|y - z\|^2 - \|x - z\|^2\right) .$$

We also state the next lemma for general Bregman-divergences and for the more general mirror descent algorithm:

Lemma 7.9. [Non examinable] *For any iterates $y^{(t+t)}$, $x^{(t+1)}$ of the mirror descent algorithm using the mirror map Φ and for any $z \in \Delta_n$ we have*

$$D_\Phi(z, x^{(t+1)}) + D_\Phi(x^{(t+1)}, y^{(t+1)}) - D_\Phi(z, y^{(t+1)}) \le 0\,.$$

Proof. According to Lemma 7.8, the expression can be written as

$$\left\langle \nabla\Phi(x^{(t+1)}) - \nabla\Phi(y^{(t+1)}), x^{(t+1)} - z\right\rangle \le 0\,.$$

Recall that $x^{(t+1)}$ is defined as the minimiser of $h(x) = D_\Phi(x, y^{(t+1)})$ over $x \in \Delta_n$. Thus, from the first order optimality condition,

$$\left\langle \nabla h(x^{(t+1)}), x^{(t+1)} - z\right\rangle \le 0\,.$$

The claim follows by observing that $\nabla h(x^{(t+1)}) = \nabla\Phi(x^{(t+1)}) - \nabla\Phi(y^{(t+1)})$. □

We are ready to prove the running time bound.

Theorem 7.10. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function with $\Delta_n \subseteq \mathbf{dom}\, f$. Assume f is differentiable and $\|\nabla f(x)\|_\infty \le L$ for all $x \in \Delta_n$. Let $x^* = \arg\min_{x\in\Delta_n} f(x)$. Then, starting from any $x^0 \in \Delta_n$ that is strictly positive, and for any $\varepsilon > 0$, exponentiated gradient descent finds an ε-approximate solution within*

$$T \ge 4.5\frac{L^2 \log n}{\varepsilon^2}$$

iterations, using step-size

$$\eta = \frac{\sqrt{2\log n}}{L\sqrt{T}}\,.$$

Proof. **[Non examinable]**

We let $p^* = f(x^*)$ denote the optimum value, and start with the usual bound

$$f(x^{(t)}) - f(x^*) \le \left\langle g^{(t)}, x^{(t)} - x^*\right\rangle \tag{7.2}$$

Note that if the initial solution $x^{(0)}$ was nonzero on each coordinates, the update rules keep the same property for all subsequent iterates. From the update rule $y_i^{(t+1)} = x_i^{(t)} \exp\left(-\eta g_i^{(t)}\right)$, and using that $x_i^{(t)}, y_i^{(t+1)} \ne 0$, we get

$$g_i^{(t)} = \frac{1}{\eta}\log\left(\frac{x_i^{(t)}}{y_i^{(t+1)}}\right) . \tag{7.3}$$

Thus,

$$\begin{aligned}\eta(f(x^{(t)}) - f(x^*)) &\le \sum_{i=1}^{n} x_i^{(t)} \log\left(\frac{x_i^{(t)}}{y_i^{(t+1)}}\right) - \sum_{i=1}^{n} x_i^* \log\left(\frac{x_i^{(t)}}{y_i^{(t+1)}}\right) \\ &= D_\Phi(x^{(t)}, y^{(t+1)}) - D_\Phi(x^*, y^{(t+1)}) + D_\Phi(x^*, x^{(t)})\,,\end{aligned} \tag{7.4}$$

where the second line follows from simple calculation. (This can also be obtained from Lemma 7.8).

We will create a telescoping sum using the following bound.

Claim 7.11.

$$D_\Phi(x^{(t)}, y^{(t+1)}) - D_\Phi(x^*, y^{(t+1)}) \le \eta\left\langle g^{(t)}, x^{(t)} - x^{(t+1)}\right\rangle - D_\Phi(x^*, x^{(t+1)}) - D_\Phi(x^{(t+1)}, x^{(t)})\,.$$

Proof. Similarly to the previous argument with $x^{(t+1)}$ in place of x^*, using (7.3) we can write

$$\begin{aligned}\eta\left\langle g^{(t)}, x^{(t)} - x^{(t+1)}\right\rangle &= \sum_{i=1}^{n} x_i^{(t)} \log\left(\frac{x_i^{(t)}}{y_i^{(t+1)}}\right) - \sum_{i=1}^{n} x^{(t+1)} \log\left(\frac{x_i^{(t)}}{y_i^{(t+1)}}\right) \\ &= D_\Phi(x^{(t)}, y^{(t+1)}) + D_\Phi(x^{(t+1)}, x^{(t)}) - D_\Phi(x^{(t+1)}, y^{(t+1)})\,.\end{aligned}$$

Substituting this expression, the claim is equivalent to

$$-D_\Phi(x^*, y^{(t+1)}) \le -D_\Phi(x^{(t+1)}, y^{(t+1)}) - D_\Phi(x^*, x^{(t+1)})\,,$$

which is equivalent to the expression in Lemma 7.9 with $z = x^*$. □

Substituting the inequality in the claim to (7.4), we get

$$\begin{aligned}\eta(f(x^{(t)}) - f(x^*)) &\le D_\Phi(x^{(t)}, y^{(t+1)}) - D_\Phi(x^*, y^{(t+1)}) + D_\Phi(x^*, x^{(t)}) \\ &\le D_\Phi(x^*, x^{(t)}) + \eta\left\langle g^{(t)}, x^{(t)} - x^{(t+1)}\right\rangle - D_\Phi(x^*, x^{(t+1)}) - D_\Phi(x^{(t+1)}, x^{(t)})\,.\end{aligned} \tag{7.5}$$

We can bound the scalar product term using Theorem 7.4 as

$$\eta\left\langle g^{(t)}, x^{(t)} - x^{(t+1)}\right\rangle \le \eta\|g^{(t)}\|_\infty \|x^{(t)} - x^{(t+1)}\|_1 \le \eta L\|x^{(t)} - x^{(t+1)}\|_1\,,$$

and we can lower bound $D_\Phi(x^{(t+1)}, x^{(t)}) \ge \frac{1}{2}\|x^{(t)} - x^{(t+1)}\|_1^2$ from Pinsker's inequality (Theorem 7.6). Therefore,

$$\eta\left\langle g^{(t)}, x^{(t)} - x^{(t+1)}\right\rangle - D_\Phi(x^{(t+1)}, x^{(t)}) \le \eta L\|x^{(t)} - x^{(t+1)}\|_1 - \frac{1}{2}\|x^{(t)} - x^{(t+1)}\|_1^2 \le \frac{(\eta L)^2}{2}\,,$$

where the last inequality uses that the function $h(t) = \eta L t - \frac{1}{2}t^2$ is concave and is maximised at $t = \eta L$. Subsituting back to (7.5), the expression simplifies to

$$\eta(f(x^{(t)}) - f(x^*)) \le D_\Phi(x^*, x^{(t)}) - D_\Phi(x^*, x^{(t+1)}) + \frac{(\eta L)^2}{2}\,. \tag{7.6}$$

We are ready to telescope! Summing this for $t = 0, 1, \ldots, T-1$ and dividing by T/η, we get

$$\frac{1}{T}\sum_{i=0}^{T-1} f(x^{(t)}) - f(x^*) \le \frac{1}{T\eta} D_\Phi(x^*, x^{(0)}) + \frac{\eta L^2}{2}\,.$$

We can bound $D_\Phi(x^*, x^{(0)}) \le \log n$, an upper bound on the KL-divergence using the choice $x^{(0)} = \frac{1}{n}\mathbf{1}_n$. Hence, for a given T, we get the best bound for

$$\eta = \frac{\sqrt{2\log n}}{L\sqrt{T}}\,,$$

giving

$$\frac{1}{T}\sum_{i=0}^{T-1} f(x^{(t)}) - f(x^*) \le \frac{3}{\sqrt{2}} \cdot \frac{L\sqrt{\log n}}{\sqrt{T}}\,.$$

This implies the claim. □

We note that Theorem 7.1 on the general mirror descent method can be proved following the same lines.

Chapter 8

Online convex optimisation

It the previous chapters, we considered full-information optimisation problems with the function and constraints provided. We now make a detour to *online convex optimisation* that captures sequential decisions with unknown future events. We refer the reader to Hazan's book [3] for a comprehensive treatment of this topic.

We start by describing a general model and some illustrative examples. We are given a *time horizon* $T \in \mathbb{N}$ that is known in advance, and we need to make sequential decisions at time-steps $t = 1, 2, \ldots, T$. We are also given a set $K \subseteq \mathbb{R}^n$ that describes the set of *possible decisions*; we will assume that this set is convex. In time-step t, the decision maker needs to select a decision $x^{(t)} \in K$. After this decision is made, the cost-function $f^{(t)} : K \to \mathbb{R}$ is revealed, and the decision maker incurs a cost $f^{(t)}(x^{(t)})$.

A standard way to evaluate a sequence of decisions $X = (x^{(1)}, x^{(2)}, \ldots, x^{(t)})$ is to compare the total cost to minimum total cost of any *fixed* decision $z \in K$. The difference is called the *regret*, and is defined as

$$\text{Regret}(X) = \sum_{t=1}^{T} f^{(t)}(x^{(t)}) - \min_{z \in K} \sum_{t=1}^{T} f^{(t)}(z)\,.$$

We will typically aim to bound the *average regret*, $\text{Regret}(X)/T$. Note that the regret may be negative.

The set of cost functions can be *arbitrary*: one can think of online optimisation as a game between a decision maker and an adversary who sees all previous decisions and can come up with any malicious cost function next.

The possible regret bounds necessarily depend on the magnitude of the functions $f^{(t)}$. As an example, let $K = [0, 1]$ and $T \geq 2$. The player (decision maker) needs to pick some $x^{(1)} \in [0, 1]$. If $x^{(1)} > 0.5$, the adversary can come up with $f^{(1)}(x) = Lx$, and if $x^{(1)} \leq 0.5$, they can select $f^{(1)}(x) = L(1 - x)$ for some large value $L > 0$. Assume all subsequent cost functions are the constant $f^{(t)}(x) = 0$. Thus, the regret of the sequence will be at least $L/2$.

Hence, we can only hope for meaningful regret bounds if our functions (as well as the decision set K) are bounded.

Example 1: Expert advice Assume we need to make a sequence of T decisions with advice from n available experts. However, we do not have any reliable a priori information on the skills and trustworthiness of these experts. At each time-step, we need to pick an expert i and take an action following their advice. Subsequently, we observe the outcome, and associate a cost $g_j^{(t)} \in [0, 1]$ with the advice of each expert j. That is, $g_j^{(t)} = 0$ means that expert j gave the perfect advice and $g_j^{(t)} = 1$ corresponds to the worst possible advice.

Instead of trusting a single expert and ignoring all others, we can associate probabilities with them: let $x_j^{(t)}$ be the probability of following expert j; $x_j^{(t)} \geq 0$ and $\sum_{j=1}^{n} x_j^{(t)} = 1$, that is, $x^{(t)} \in \Delta_n$, the probability simplex. Then, the expected cost at time t is

$$f^{(t)}(x) = \sum_{j=1}^{n} x_j^{(t)} g_j^{(t)} = \left\langle g^{(t)}, x^{(t)} \right\rangle .$$

Consider the overall regret

$$\text{Regret}(X) = \sum_{t=1}^{T} \left\langle g^{(t)}, x^{(t)} \right\rangle - \min_{z \in \Delta_n} \left\langle \sum_{t=1}^{T} g^{(t)}, z \right\rangle .$$

Here, the optimal fixed combination $z \in \Delta_n$ is the solution to a linear program. Recall that a linear program over a bounded polyhedron always has a vertex optimal solution; in this case, it corresponds to a vector e_i that is 1 on expert i and 0 on all other experts. Hence, the regret compares the total expected cost over T time-steps compared to following the advice of the best expert throughout. Note that here, the best expert is defined as the one we can identify in hindsight with the best advice for this particular sequence of decisions.

Example 2: Portfolio selection Assume an investor is distributing their wealth over n assets over T months. They have an initial budget of B_1 and initially invest $q_j^{(1)}$ amount in asset j; they can rebalance their budget B_t at the beginning of every subsequent month. During month t, the price of asset j increases by a factor $r_j^{(t)}$ (or decreases if $r_j^{(t)} < 1$).

Let us define $x_j^{(t)} = q_j^{(j)}/B_t$, i.e., the fraction of the current wealth invested in asset j. Thus, $x^{(t)} \in \Delta_n$. Then, we can see that

$$B_{t+1} = \sum_{j=1}^{n} r_j^{(t)} q_j^{(t)} = B_t \sum_{j=1}^{n} r_j^{(t)} x_j^{(t)} = B_t \left\langle r^{(t)}, x^{(t)} \right\rangle .$$

Defining

$$f^{(t)}(x^{(t)}) = -\log\left(\left\langle r^{(t)}, x^{(t)} \right\rangle\right) ,$$

expresses $\log(B_t/B_{t+1})$, the decrease in the value of the total portfolio (ideally, negative). For this function,

$$\text{Regret}(X) = -\sum_{t=1}^{T} \log \left\langle r^{(t)}, x^{(t)} \right\rangle + \min_{z \in \Delta_n} \sum_{t=1}^{T} \log \left\langle r^{(t)}, z \right\rangle .$$

Here, the optimal z corresponds to the best possible fixed portfolio in hindsight, i.e., the best *constant rebalanced portfolio*.

Example 3: Spam filtering Spam filtering is a prototypical example of online learning against adversarial behaviour. We can think of it as a game between the system administrator and the spammer. Emails arrive sequentially, and the system administrator can dynamically update the filtering criteria on whether they are recognised as genuine emails or sent to the spam folder. The spammer can (possibly) detect this information, and can tweak their emails so that they have a better chance of getting through.

A common approach is to use a *bag of words* model: an email is represented as a vector $u \in \mathbb{R}^d$, where d is the size of the dictionary, and u_j is the frequency of the occurrence of word j. The spam filter is a parametric classification model (such as logistic regression or SVM) that, for a parameter vector $x \in \mathbb{R}^n$ and vector $u \in R^d$ representing an email, outputs the probability of u being spam as $\Gamma(x, u) \in [0, 1]$.

Consider now a sequence of messages $u^{(1)}, u^{(2)}, \ldots, u^{(t)}$; let $b_t = 1$ if the i-th message is a spam and 0 if not. The system administrator has to choose a parameter vector $x^{(t)}$; the choice may be restricted to a convex set K such as an ℓ_p-ball of certain radius for some $p \geq 1$. For a loss function $L : \mathbb{R}^2 \to \mathbb{R}_+$, the cost function at time-step t is

$$f^{(t)}(x^{(t)}) = L(b_t, \Gamma(x^{(t)}, u^{(t)})) .$$

For this model, the regret compares the sequence of online decisions to the best possible performing parameter choice for this set of emails in hindsight.

8.1 Online gradient descent

A remarkable feature of the gradient descent method is that the basic analysis immediately extends to the online setting. Assume we have a sequence of functions $f^{(t)}$, revealed after the iterate $x^{(t)}$ was chosen. As we will see, we can still use (projected) gradient descent, moving in the direction of $-\nabla f^{(t)}(x^{(t)})$. We now present the adaptation of the projected gradient descent method from Section 5.1.

ONLINE GRADIENT DESCENT
Input: A convex set K, the number of rounds T. The convex function $f^{(t)} : K \to \mathbb{R}$ is revealed in round t after $x^{(t)}$ is chosen.
Output: A sequence $(x^{(1)}, x^{(2)}, \ldots, x^{(t)})$
Determine the step-size $\eta > 0$ based on T and other parameters.
Pick the initial $x^{(1)} \in K$.
For $t = 1, 2 \ldots, T$ do
  The cost $f^{(t)}$ is revealed ;
  $y^{(t+1)} = x^{(t)} - \eta \nabla f^{(t)}(x^{(t)})$;
  $x^{(t+1)} = \Pi_K(y^{(t+1)})$;

The crucial observation is that for any $x^* \in K$, we can modify (5.3) to

$$f^{(t)}(x^{(t)}) - f^{(t)}(x^*) \leq \left\langle \nabla f^{(t)}(x^{(t)}), x^{(t)} - x^* \right\rangle = \frac{1}{\eta} \left\langle x^{(t)} - y^{(t+1)}, x^{(t)} - x^* \right\rangle , \tag{8.1}$$

We follow the exact same steps as in Section 5.1.2 to obtain the following inequality in place of (5.6) (after ignoring the last term):

$$\frac{1}{T} \sum_{t=1}^{T} \left(f^{(t)}(x^{(t)}) - f^{(t)}(x^*) \right) \leq \frac{\eta}{2T} \sum_{t=1}^{T} \left\| \nabla f^{(t)}(x^{(t)}) \right\|^2 + \frac{1}{2T\eta} \left\| x^{(0)} - x^* \right\|^2 . \tag{8.2}$$

The left hand side is the regret against x^*; this can be chosen as any point in K. If we can bound the right-hand side as $\leq \varepsilon$ for any x^*, then our online algorithm has average regret $\leq \varepsilon$. Analogously to Theorem 4.3 and Theorem 5.5, we get:

Theorem 8.1. *Let $K \subseteq \mathbb{R}^n$ be a convex set with diameter at most R. Assume that the cost functions $f^{(t)} : K \to \mathbb{R}$ revealed in the online optimisation algorithm are convex and differentiable, and have Lipschitz-parameter L. Then, online gradient descent achieves average regret at most $RL/\sqrt{T}$ over T time-steps, using step-size $\eta = R/(L\sqrt{T})$.*

Given this connection, we can also adapt other gradient methods to the online setting: if our functions are Lipschitz in a different norm, we can use online mirror descent with a suitable mirror map. In what follows, we demonstrate this on the example of exponentiated gradient descent.

8.2 The multiplicative weights update method

Let us now consider the example on expert advice above: in time-steps $t = 1, 2, \ldots, T$, the decision maker associates probabilities $x_j^{(t)}$ to n experts, and subsequently, the costs are revealed: the cost of expert j's advice is $g_j^{(t)} \in [0, 1]$. As noted, this corresponds to online learning with the functions

$$f^{(t)}(x) = \left\langle g^{(t)}, x^{(t)} \right\rangle$$

that are 1-Lipschitz in ℓ_1-norm. Hence, the online version of exponentiated gradient is a natural choice for this setting. In this context, it is called the *Multiplicative Weights Update method* or the *Hedge algorithm*. We initialise with the uniform distribution $x^{(1)} = \frac{1}{n}\mathbf{1}_n$, and use the same updates as in exponentiated gradient descent.

MULTIPLICATIVE WEIGHTS UPDATE
Input: A vector $g^{(t)} \in [0,1]^n$ is revealed in round t after $x^{(t)}$ is chosen.
Output: A sequence $(x^{(1)}, x^{(2)}, \dots, x^{(t)})$
Determine the step-size $\eta > 0$ based on T and other parameters.
Set the initial distribution $x^{(1)} = \frac{1}{n}\mathbf{1}_n$.
For $t = 1, 2 \dots, T$ do
The vector $g^{(t)}$ is revealed ;
For $i = 1, 2 \dots, n$ do $y_i^{(t+1)} = x_i^{(t)} \exp\left(-\eta g_i^{(t)}\right)$;
$x^{(t+1)} = \frac{y^{(t+1)}}{\|y^{(t+1)}\|_1}$;

Theorem 8.2. *Assuming that the cost vectors $g^{(t)}$ are in $[0,1]^n$, the multiplicative weights method achieves average regret at most*

$$\sqrt{\frac{4.5 \log n}{T}}$$

using step-size $\eta = \sqrt{\frac{2 \log n}{T}}$.

The proof directly follows from the analysis in Section 7.2.1. The derivation of (7.6) does not rely on using the same function f in consecutive iterations. We could apply it for $f^{(t)}(x) = \left\langle g^{(t)}, x \right\rangle$ and the bound $\|g^{(t)}\|_\infty \le L = 1$ to obtain

$$\eta \left\langle g^{(t)}, x^{(t)} - x^* \right\rangle \le D_\Phi(x^*, x^{(t)}) - D_\Phi(x^*, x^{(t+1)}) + \frac{\eta^2}{2}\,. \tag{8.3}$$

Moreover this is valid for any choice of $x^* \in K$. Summing up, we obtain the desired regret bound.

Variants of the multiplicative weights update method are prevalent in optimisation and machine learning. We briefly mention their importance in the context of *ensemble learning*. Given a regression or classification problem, we can consider different algorithms as 'experts', and would like to identify their most efficient combination. A particular variant, *Adaboost* can turn 'weak learners', methods that do just marginally better than random guessing, into strong classifiers. This is achieved by iteratively reweighting the dataset based on the success of the previous round, and training a new classifier on the current weighted instance.

8.2.1 The Winnow algorithm

[Non examinable]

We now describe a direct application of the multiplicative weights update method to classification. As in Section 2.3.3 for logistic regression and in Section 6.3 for support vector machines, we consider a *binary classification* problem with m feature vectors $a_j \in \mathbb{R}^n$, $j = 1, 2, \dots, m$, including the bias term 1 as $a_{j1} = 1$, and a target variable will be $b_j \in \{+1, -1\}$. Perfect linear separation between the two classes corresponds to a vector $x \in \mathbb{R}^n$ such that

$$b_j \langle a_j, x \rangle > 0 \quad \forall j = 1, 2, \dots, m\,. \tag{8.4}$$

The *Winnow* algorithm is a simple iterative method to find a perfect separation assuming one exists. It is similar to the classical *Perceptron* algorithm. They maintain a current candidate separator $x^{(t)}$, and update it every time they encounter a data point that is incorrectly classified by $x^{(t)}$.

We can consider different access models to the dataset: we can either scan through the dataset in a fixed order repeatedly, or consider random samples arriving one-by-one. For certain structured datasets, there might be a subroutine available for finding a misclassified data point. This flexible access is an important feature of the Winnow and Perceptron algorithms. We can also think of them as simple examples of *reinforcement learning*, where our method is dynamically updated based on the success or failure.

We now describe the Winnow algorithm for the setting where we are interested in a nonnegative separator vector $x \in \mathbb{R}^n_+$ satisfying (8.4). We will see in the exercises how this can be extended to the general case where x may have arbitrary signs. If we assume $x \in \mathbb{R}^n_+$, we can also normalise such that $x \in \Delta_n$ is a probability distribution. This follows by noting that (8.4) remains true when replacing a feasible x by αx for any $\alpha > 0$, and that $\|x\|_1 \neq 0$ because $x = 0$ is not a feasible solution.

Similary, we can assume that all feature vectors a_j are scaled such that $\|a_j\|_\infty \leq 1$, since scaling does not affect the constraints in the separation problem, and $a_j \neq 0$ since a bias term was included.

WINNOW ALGORITHM
Input: A dataset of m vectors $a_j \in \mathbb{R}^n$, $j = 1, 2, \ldots, m$, $\|a_j\|_\infty = 1$ and $b_j \in \{+1, -1\}$, step-size $\eta > 0$.
Output: A separator $x^{(t)} \in \Delta^n$ such that $b_j \langle a_j, x \rangle > 0$ for all $j = 1, 2, \ldots, m$.
$t = 1$; $x^{(1)} = \frac{1}{n}\mathbf{1}_n$;
Repeat
  Scan through $j = 1, 2, \ldots, m$ until $b_j \left\langle a_j, x^{(t)} \right\rangle \leq 0$ is found
  If such a j is found, then
    For $i = 1, 2 \ldots, n$ do $y_i^{(t+1)} = x_i^{(t)} \exp(\eta b_j a_{ji})$;
    $x^{(t+1)} = \frac{y^{(t+1)}}{\|y^{(t+1)}\|_1}$;
    $t = t + 1$;
  If no such j is found, then terminate by outputting $x^{(t)}$.

The Winnow algorithm can be seen an instantiation of the multiplicative weights method, where the next loss vector $g^{(t)}$ is chosen as $g^{(t)} = -b_j a_j$ for a data point such that $b_j \left\langle a_j, x^{(t)} \right\rangle \leq 0$. Note that $g_j^{(t)} \in [-1, 1]$ instead of $[0, 1]$, but this does not affect the analysis that only uses $\|g^{(t)}\|_\infty \leq 1$.

The analysis reveals that as long as the data points are 'well-separable', i.e., if there exists a separating hyperplane with a large enough margin, then the Winnow algorithm terminates with such a hyperplane in a bounded number of iterations.

Theorem 8.3. *Assume $\|a_j\|_\infty \leq 1$ for all $j = 1, 2, \ldots, m$ and there exists a separator $x^* \in \Delta_n$ such that $b_j \langle a_j, x^* \rangle \geq \varepsilon$ for all $j = 1, 2, \ldots, n$. Then, the Winnow algorithm terminates within*

$$\frac{4.5 \log n}{\varepsilon^2}$$

iterations using step-size $\eta = C\varepsilon$ for some constant $C > 0$.

Proof. As noted above, the Winnow algorithm can be seen as a special case of the multiplicative weights update method. Assume it does not terminate for $T > \frac{4.5 \log n}{\varepsilon^2}$ iterations, using the corresponding step-size η. Then, the sequence of solutions must have average regret $< \varepsilon$ over the first T iterations.

Recalling that $g^{(t)} = -b_j a_j$ for some $1 \leq j \leq m$, we see that

$$\left\langle \sum_{j=1}^{T} g^{(t)}, x^* \right\rangle < -T\varepsilon \,.$$

Hence, one of the first T iterates $x^{(t)}$ must have $\left\langle g^{(t)}, x^{(t)} \right\rangle < 0$. However, we must have $\left\langle g^{(t)}, x^{(t)} \right\rangle \geq 0$ in every iteration by the very choice of $g^{(t)}$, a contradiction. □

Chapter 9

Newton's method

9.1 Root finding of univariate functions

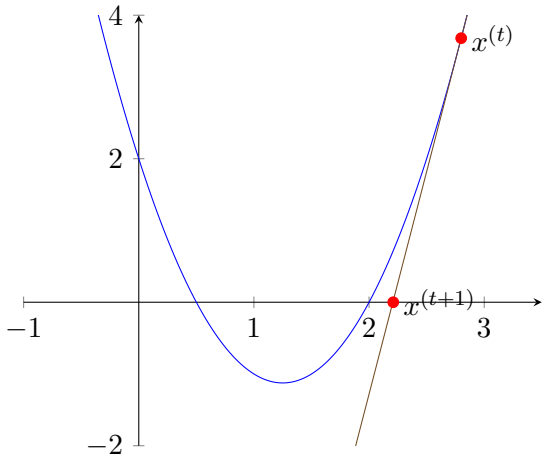
The original version of the *Newton method* or *Newton-Raphson method* is for finding roots of differentiable univariate functions $f : \mathbb{R} \to \mathbb{R}$. Given a starting point $x^{(0)} \in \mathbf{dom}\, f$, we compute iterates

$$x^{(t+1)} = x^{(t)} - \frac{f(x^{(t)})}{f'(x^{(t)})} \quad t = 1, 2, \ldots, T\,.$$

Geometrically, this amounts to following the tangent line of the graph of $f(x)$ at $x^{(t)}$:

$$g(x) = f(x^{(t)}) + f'(x^{(t)})(x - x^{(t)}),$$

and computing the root of $g(x) = 0$. This is illustrated in Figure 9.1.



Clearly, this method may not always converge. E.g., if $f'(x^{(t)}) = 0$, then the update step is undefined, and it can also be numerically unstable if $|f'(x^{(t)})|$ is small.

Example 1: the Babylonian method A special case of Newton's method for numerically approximating squareroots has been used since ancient times. The *Babylonian method* or *Heron's method* is an iterative process for approximating $\sqrt{S}$ for $S > 0$ using elementary arithmetic operations. Starting from $x^{(0)} = S$, the consecutive updates are

$$x^{(t+1)} = \frac{1}{2}\left(x^{(t)} + \frac{S}{x^{(t)}}\right).$$

This is identical to Newton's method for the function

$$f(x) = x^2 - S\,,$$

This function has two roots, $\sqrt{S}$ and $-\sqrt{S}$. If $x^{(t)} > \sqrt{S}$, then $x^{(t+1)} > \sqrt{S}$ follows from the inequality of geometric and arithmetic means. Moreover, $x^{(t+1)} < x^{(t)}$ also holds in this case since $S/x^{(t)} < \sqrt{S}$.

Let us now show that starting with $x^{(0)} > \sqrt{S}$ converges to the positive root $\sqrt{S}$. (For any $a > 0$, either a or S/a is a suitable starting point.) Let us define

$$\delta_t = \frac{x^{(t)}}{\sqrt{S}} - 1 \,.$$

Hence $\delta_0 > 0$, and the above argument shows that $\delta_t > 0$ throughout. Elementary calculation shows

$$\delta_{t+1} = \frac{\delta_t^2}{2(1+\delta_t)} \,.$$

From here, we see that

$$0 < \delta_{t+1} < \min\left\{ \frac{\delta_t^2}{2}, \frac{\delta_t}{2} \right\} . \tag{9.1}$$

The second bound is stronger as long as $\delta_t \geq 1$; in this case, the error δ_t is at least halved in each iteration. The first bound dominates for $\delta_t < 1$.

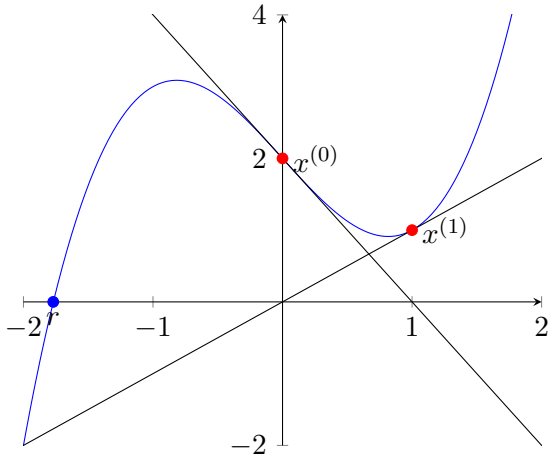
The bound (9.1) does not only show that $x^{(t)} \to \sqrt{S}$, but reveals an interesting phenomenon. In the first $\log_2 \delta_0$ iterations we reach $\delta_t < 1$, at which point we turn to a higher speed and achieve much faster, *quadratic* convergence. If $\delta_t < 2^{-p}$ for $p \geq 0$, then $\delta_{t+1} < 2^{-2p-1}$. Hence, once we have $\delta_t < 1$, we will reach $\delta_T < \varepsilon$ in $\log_2 \log_2(1/\varepsilon)$ iterations!

As we will see, this illustrates a general phenomenon for Newton's method: once we are sufficiently close to a root, the method converges extremely efficiently. At the same time, convergence can be much slower or the method may not converge at all if starting further away from the optimum.

Example 2: cycling behaviour As already noted, Newton's method breaks down if $f'(x^{(t)}) = 0$; e.g., if we start with $x^{(0)} = 0$ for the above example $f(x) = x^2 - S$. Even if this is not the case, convergence may not be guaranteed. Consider the function

$$f(x) = x^3 - 2x + 2 \,,$$

and set the starting point as $x^{(0)} = 0$. Then, $x^{(1)} = 1$, $x^{(2)} = 0$, and the method continues oscillating between these two solutions; see Figure 9.1. Note that this function has a unique root at $r \approx -1.77$, and the first iteration start moving in the wrong direction, away from the root.



9.1.1 Quadratic convergence of root finding

We now formulate a general condition that guarantees that, if starting sufficiently close to a root, then Newton's method converges quadratically.

Theorem 9.1. *Let $f : \mathbb{R} \to \mathbb{R}$ be twice continuously differentiable, and let r be a root, that is, $f(r) = 0$, and $x^{(0)} \in \mathbf{dom}\, f$ a starting point. Let*

$$G = \sup \left\{ \left| \frac{f''(y)}{2f'(z)} \right| : |r - y| \leq |r - z| \leq |r - x^{(0)}| \right\}, \tag{9.2}$$

and assume $G|r - x^{(0)}| \leq 1$. Then,

$$|r - x^{(t+1)}| \leq G(r - x^{(t)})^2$$

holds at every iteration.

Before the proof, let us check the condition on $f(x) = x^2 - S$ analysed above with $r = \sqrt{S}$. We have $f''(y)/(2f'(z)) = 1/(2z)$. For $|r - x^{(0)}| = \alpha$, G is bounded if $\alpha < r$, in which case $G = 1/(2(r - \alpha))$. We also need $G\alpha < 1$, or equivalently, $\alpha < \frac{2}{3}r$. Using the notation above with $x^{(0)} > \alpha$, this is equivalent to $\delta_0 < 2/3$.

Proof. It suffices to show the statement for $t = 0$. Then, we obtain $|r - x^{(1)}| < |r - x^{(0)}|$ from the assumption $G|r - x^{(0)}| < 1$. Let G' be the quanity for $x^{(1)}$ instead of $x^{(0)}$; $G' \leq G$ holds since the constraint on y and z is more restrictive. We also have $G'|r - x^{(1)}| < 1$. Thus, we can show the statement for all values of t by induction.

The boundedness of G already implies $f'(x^{(0)}) \neq 0$; hence, the iteration is well-defined. We use a second order Taylor expansion around $x^{(0)}$ (Theorem 1.15). We get

$$f(r) = f(x^{(0)}) + f'(x^{(0)})(r - x^{(0)}) + \frac{1}{2}f''(\bar{r})(r - x^{(0)})^2 ,$$

where $\bar{r}$ is a value between $x^{(0)}$ and r. Substituting $f(r) = 0$ and $f(x^{(0)}) = f'(x^{(0)})(x^{(0)} - x^{(1)})$ from the update rule, this can be written as

$$f'(x^{(0)})(x^{(0)} - x^{(1)}) + f'(x^{(0)})(r - x^{(0)}) + \frac{1}{2}f''(\bar{r})(r - x^{(0)})^2 = 0$$

which we rearrange as

$$x^{(1)} - r = \frac{f''(\bar{r})}{2f'(x^{(0)})}(r - x^{(0)})^2 .$$

By the definition, G is an upper bound on the absolute value of the fraction, and therefore

$$|x^{(1)} - r| \leq G(r - x^{(0)})^2 ,$$

as required. □

Let us now see how this theorem can be used to bound the number of steps to reach a certain accuracy. Let $\alpha_t = |r - x^{(t)}|$, and assume that for some $\varepsilon > 0$, we would like to bound the number of iterates that guarantee $\alpha_t \leq \varepsilon$. The theorem asserts that if $G\alpha_0 \leq 1$, then $\alpha_{t+1} \leq G\alpha_t^2$. By induction, it is easy to verify that for every $t \geq 1$, we have

$$G\alpha_t \leq (G\alpha_0)^{2^t} .$$

Assume that we have the stronger bound $G\alpha_0 \leq 1/2$ on the starting solution (could use any constant < 1). Then, it suffices to pick t such that

$$\frac{1}{2^{2^t}} \leq G\varepsilon ,$$

that is,

$$t \geq \log_2 \log_2 \left(\frac{1}{G\varepsilon} \right) .$$

9.2 Newton's method for optimisation

Consider now a global optimisation problem. In constrast to the previous chapters, we do not always require convexity. We will however assume that the function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable. We will focus on finding critical points, that is, points in $\mathbf{dom}\, f$ with

$$\nabla f(r) = 0\,.$$

Recall that for convex functions, this is equivalent to global minimisation. For non-convex functions, this could also be a local minimum/maximum or a saddle point.

9.2.1 The univariate case

In the univariate case, we can naturally apply Newton's method for the function $f'(x)$. This yields updates

$$x^{(t+1)} = x^{(t)} - \frac{f'(x^{(t)})}{f''(x^{(t)})} \quad t = 1, 2, \ldots, T\,,$$

assuming the second derivative never vanishes. Observe that this corresponds to a gradient descent iteration with varying step-size $\eta_t = 1/f''(x^{(t)})$. In contrast to gradient descent, the step-size could even be negative.

From here, we can further observe that if $f(x)$ is convex, then the iteration is the optimal solution to the following quadratic minimisation problem:

$$x^{(t+1)} = \arg\min_{x\in\mathbb{R}} f(x^{(t)}) + f'(x^{(t)})(x - x^{(t)}) + \frac{1}{2}f''(x^{(t)})(x - x^{(t)})^2\,. \tag{9.3}$$

In other words, we take the second-order Taylor expansion of f around $x^{(t)}$, and minimise this expression. Notice that in case $f''(x^{(t)}) = 0$, this expression is unbounded. If $f''(x^{(t)}) < 0$, then f is not convex. In this case Newton's method corresponds to finding a maximiser of the expression, also a critical point.

9.2.2 Extension to higher dimensions

Consider now a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. A natural idea is to extend (9.3) to higher dimensions. Consider the second-order Taylor-expansion of f around $x^{(t)}$, namely,

$$\hat{f}_{x^{(t)}}(x) = f(x^{(t)}) + \left\langle \nabla f(x^{(t)}), x - x^{(t)} \right\rangle + \frac{1}{2}(x - x^{(t)})^\top \nabla^2 f(x^{(t)})(x - x^{(t)}) \tag{9.4}$$

We pick the next iterate as the minimiser of this function.

$$x^{(t+1)} = \arg\min_{x\in\mathbb{R}} \hat{f}_{x^{(t)}}(x). \tag{9.5}$$

To simplify the notation in this chapter we denote the Hessian as

$$H_f(x) = \nabla^2 f(x)\,,$$

or simply $H(x)$ if f is clear from the context. We let $H^{-1}(x) = (\nabla^2 f(x))^{-1}$ denote the inverse of the Hessian. Setting $\nabla \hat{f}_{x^{(t)}}(x) = 0$, we obtain the following natural generalisation of Newton's method for higher dimensions:

$$x^{(t+1)} = x^{(t)} - H^{-1}(x^{(t)})\nabla f(x^{(t)})\,. \tag{9.6}$$

This expression is well defined as long as $H(x^{(t)})$ is non-singular. Recall that for convex functions, $H(x^{(t)})$ is positive semidefinite. If f is not convex, then we can still use the iterates (9.6); the goal of the algorithm is then finding a critical point of the expression in (9.5). For brevity, we denote

$$N_f(x) := -H^{-1}(x)\nabla f(x)\,, \tag{9.7}$$

or simply $N(x)$ when clear from the context. With this notation, the updates of Newton's method are

$$x^{(t+1)} = x^{(t)} + N(x^{(t)})\,.$$

Convex quadratic functions Recall from Section 2.1.6 that a quadratic function is convex if and only if it can be written in the form

$$f(x) = \frac{1}{2}x^\top Qx + \langle p, x\rangle + r\,,$$

for a symmetric positive semidefinite function $Q \in \mathbb{R}^{n\times n}$ and $p \in \mathbb{R}^n$. The Newton updates are undefined if Q is positive semidefinite but not positive definite, in which case Q^{-1} does not exist. Let us now assume that $Q \succ 0$, and thus, there exists a global minimiser x^*.

We have $\nabla f(x) = p+Qx$ and $H(x) = Q$ for every $x \in \mathbb{R}^n$, and it is easy to verify that $\hat{f}_{x^{(t)}}(x) = f(x)$ for every $x^{(t)} \in \mathbb{R}^n$. Consequently, already the first iteration of Newton's method finds the exact global minimiser: $x^{(1)} = x^*$.

To verify this directly, we compute $N(x) = -Q^{-1}(Qx+p) = -x-Q^{-1}p$. Therefore, $x^{(1)} = -Q^{-1}p$ regardless of $x^{(0)}$; and this is the solution to $\nabla f(x) = 0$. This argument remains valid even if f is not convex.

9.3 Newton's method as steepest descent in local norm

We now present another perspective on Newton's method that reveals the connection to gradient methods. Recall from Chapter 4 that gradient descent amounts to *steepest descent* in the Euclidean norm: it uses the direction $v = \nabla f(x)/\|\nabla f(x)\|_2$ that maximises

$$\max_{v\in\mathbb{R}^n:\,\|v\|_2\le 1} -\langle \nabla f(x), v\rangle\ .$$

This scalar product corresponds to the directional derivative, that is, the decrement in direction v. We normalise with $\|v\|_2 \le 1$ (which at an optimal solution must hold at equality) to get the decrement rate measured in ℓ_2-norm.

One could instead look at gradient descent for an arbitrary different norm $\|.\|$, simply replacing $\|v\|_2 \le 1$ by $\|v\| \le 1$. Observe that the maximum decrease rate is by definition the dual norm $\|\nabla f(x)\|^\star$.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a strongly convex and twice differentiable function; thus, $H(x) \succ 0$ for every $x \in \mathbf{dom}\, f$. Newton's method can be seen as a steepest descent algorithm that uses a varying norm at each iteration. The *local norm* at x is defined as

$$\|v\|_x := \sqrt{v^\top H(x) v}\,.$$

Recall from the exercises that this is a norm since $H(x)$ is assumed to be positive definite.

Proposition 9.2. *For a strongly convex and twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, the unique optimal solution to*

$$\max_{v\in\mathbb{R}^n:\,\|v\|_x\le 1} -\langle \nabla f(x), v\rangle \tag{9.8}$$

is

$$v^* = \frac{N(x)}{\|N(x)\|_x}\,,$$

the direction used by Newton's method.

Proof. For the Newton step $N(x) = -H^{-1}(x)\nabla f(x)$ we have

$$-\langle \nabla f(x), N(x)\rangle = \nabla f(x)^\top H^{-1}(x)\nabla f(x)\,.$$

Further, (recalling that $H(x)$ and therefore $H^{-1}(x)$ are symmetric), we have

$$\|H^{-1}(x)\nabla f(x)\|_x = \sqrt{\nabla f(x)^\top H^{-1}(x)H(x)H^{-1}(x)\nabla f(x)} = \sqrt{\nabla f(x)^\top H^{-1}(x)\nabla f(x)}\,,$$

thus,

$$-\langle \nabla f(x), N(x)\rangle = \sqrt{\nabla f(x)^\top H^{-1}(x)\nabla f(x)}\,.$$

From the exercises, recall that this equals the dual norm $\|\nabla f(x)\|_x^\star$, which is by definition the optimum value to the system (9.8). □

9.3.1 The Newton decrement

The *Newton decrement* of the function f at $x \in \mathbf{dom}\, f$ is defined as

$$\lambda_f(x) = \sqrt{\nabla f(x)^\top H^{-1}(x) \nabla f(x)}\,,$$

we simply write $\lambda(x)$ when clear from the context. This quantity appeared in the previous proof as $\|H^{-1}(x)\nabla f(x)\|_x$, the length of the Newton-step in the local norm at x. We also noted that it equals the dual norm $\|\nabla f(x)\|_x^\star$.

It can also be used as an estimate on the gap $f(x) - f(x^*)$ based on the following interpretation. Using the second order Taylor approximation (9.4), and the definition of the Newton step (9.5), we have

$$f(x) - \hat{f}_x(x^*) \leq f(x) - \min_{z \in \mathbf{dom}\, f} \hat{f}_x(z) = f(x) - \hat{f}_x(x + N(x))\,,$$

Elementary calculation shows the following:

Lemma 9.3. $f(x) - \hat{f}_x(x + N(x)) = \frac{1}{2}\lambda_f^2(x)$.

Hence, $\frac{1}{2}\lambda_f^2(x^{(t)})$ gives an upper bound on the optimality gap—when evaluating $\hat{f}_{x^{(t)}}(x)$ instead of $f(x)$. A common stopping criteria for Newton's method is to terminate once $\lambda_f(x^{(t)})$ falls below a certain threshold.

9.4 Affine invariance of Newton's method

Let us consider the effect of an affine transformation $g(x) = Ax + b$ on the different optimisation methods. Here $A \in \mathbb{R}^{n \times n}$ is an invertible matrix and $b \in \mathbb{R}^n$. For simplicity, we assume $b = 0$ throughout, that is, $g(x) = Ax$ is a linear transformation, but everything we show extends easily to the case $b \neq 0$.

An affine transformation can be thought of as a change of coordinates and norms. For example, the image of the unit ball $B = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$ will be

$$g(B) = \{Ax \in \mathbb{R}^n : \|x\| \leq 1\} = \{x \in \mathbb{R}^n : \|A^{-1}x\| \leq 1\} = \{x \in \mathbb{R}^n : \|x\|_{A^{-2}} \leq 1\}\,.$$

Hence, the unit ball is transformed to an ellipsoid; the Euclidean norm gets replaced by $\|x\|_{A^{-2}} \leq 1$.

Affine pre-conditioning for gradient descent An affine transformation can be a useful *pre-conditioning* to improve the geometric properties relevant for a certain optimisation method, and thereby achieve faster convergence. This is certainly applicable for gradient descent. For example, consider the quadratic function $f : \mathbb{R}^2 \to \mathbb{R}$

$$f(x_1, x_2) = \frac{1}{2}\left(Kx_1^2 + x_2^2\right)\,,$$

where $K \geq 1$. Clearly, the minimiser is $x^* = 0$. Starting from a point $x^{(0)}$, the standard gradient descent step is

$$x^{(t+1)} = x^{(t)} - \eta \nabla f(x^{(t+1)}) = \left((1 - K\eta)x_1^{(t)}, (1-\eta)x_2^{(t)}\right)\,.$$

Hence, we are only able to use small step-sizes to decrease the objective value. Already with $\eta \geq 3/K$, we will be overshooting, since $|x_1^{(t+1)}| \geq 2|x_1^{(t)}|$. Notice that the function is K-smooth; in accordance with Lemma 4.6, we would choose $\eta = 1/K$. This leads to convergence in a single iteration if $K = 1$, but at a slow pace for large values of K.

On the other hand, if we use the affine transformation $g(x_1, x_2) = (x_1/\sqrt{K}, x_2)$ (i.e., shrinking along the x-axis), then we obtain the function

$$h(x_1, x_2) = f(g(x_1, x_2)) = K\left(\frac{x_1}{\sqrt{K}}\right)^2 + x_2^2 = x_1^2 + x_2^2\,.$$

This is now a much nicer, 2-smooth function, where we can use $\eta = 1$ and achieve immediate convergence.

Affine invariance In contrast, we say that a method is *affine invariant* if such a pre-conditioning does not make a change: replacing the function $f(x)$ by

$$h(x) = f(g(x)) = f(Ax)$$

leads to the same sequence of iterates. Let us make this more precise. For an iterative algorithm, consider a sequence of iterates $x^{(t)}$ obtained by starting from $x^{(0)}$ and using the function $f(x)$. Let us also consider another sequence $y^{(t)}$ obtained by starting from $y^{(0)} = A^{-1}x^{(0)}$, and using the function $h(x)$. Thus, $h(y^{(0)}) = f(x^{(0)})$.

Then, the algorithm is *affine* invariant, if for any invertible matrix $A \in \mathbb{R}^{n\times n}$, and for any starting point $x^{(0)} \in \mathbf{dom}\, f$, we have $y^{(t)} = A^{-1}x^{(t)}$—and consequently, $h(y^{(t)}) = f(x^{(t)})$—throughout.

Using the chain rule of derivatives, we get

$$\nabla h(x) = A^\top \nabla f(Ax) \quad \text{and} \quad H_h(x) = A^\top H_f(Ax)A\,. \tag{9.9}$$

For gradient descent, the updates are

$$\begin{aligned} x^{(t+1)} &= x^{(t)} - \eta\nabla f(x^{(t)}) \\ y^{(t+1)} &= y^{(t)} - \eta\nabla h(x^{(t)}) = y^{(t)} - \eta A^\top \nabla f(Ay^{(t)})\,. \end{aligned}$$

Even if $y^{(t)} = A^{-1}x^{(t)}$, we typically have $y^{(t+1)} \neq A^{-1}x^{(t+1)}$, since $A^{-1}\nabla f(x^{(t)}) \neq A^\top \nabla f(x^{(t)})$ in general.

Affine invariance of Newton's method A remarkable property of Newton's method is affine invariance: the method is unchanged under an affine pre-conditioning. Let us now verify this. Assume $y^{(t)} = A^{-1}x^{(t)}$, and consider the updates

$$\begin{aligned} x^{(t+1)} &= x^{(t)} + N_f(x^{(t)}) = x^{(t)} - H_f^{-1}(x^{(t)})\nabla f(x^{(t)}) \\ y^{(t+1)} &= y^{(t)} + N_h(y^{(t)}) = y^{(t)} - H_h^{-1}(y^{(t)})\nabla h(y^{(t)})\,. \end{aligned}$$

We verify $y^{(t+1)} = A^{-1}x^{(t+1)}$, or equivalently, $Ay^{(t+1)} = x^{(t+1)}$. We use that $H_h^{-1}(y) = \left(A^\top H_f(Ay)A\right)^{-1} = A^{-1}H_f^{-1}(Ay)A^{-\top}$, where $A^{-\top} = (A^\top)^{-1}$. Thus, using (9.9),

$$\begin{aligned} Ay^{(t+1)} &= Ay^{(t)} - AH_h^{-1}(y^{(t)})\nabla h(y^{(t)}) \\ &= x^{(t)} - A\left(A^{-1}H_f^{-1}(Ay^{(t)})A^{-\top}\right)\cdot\left(A^\top\nabla f(Ay^{(t)})\right) \\ &= x^{(t)} - H_f^{-1}(x^{(t)})\nabla f(x^{(t)})\,, \end{aligned}$$

as required.

9.5 Quadratic convergence for optimisation

[Non examinable]

As an analogue of Theorem 9.1, we present a sufficient condition for quadratic convergence in Newton's algorithm for optimisation. We use the spectral norm $\|A\|_2$ of a matrix $A \in \mathbb{R}^{m\times n}$ (see Definition 1.9). We use the following simple properties (see exercises).

Lemma 9.4. *For the spectral norm, we have*

(i) $\|Ax\|_2 \le \|A\|_2 \cdot \|x\|_2$ *for every* $A \in \mathbb{R}^{m\times n}$ *and* $x \in \mathbb{R}^n$.

(ii) $\|AB\|_2 \le \|A\|_2 \cdot \|B\|_2$ *for* $A \in \mathbb{R}^{m\times k}$, $B \in \mathbb{R}^{k\times n}$.

(iii) If $A \in \mathbb{R}^{n\times n}$ *is a symmetric square matrix, then* $\|A\|_2$ *equals the largest absolute value of an eigenvalue of* A.

Below, all norms not indicated otherwise are standard ℓ_2 vector norms.

Theorem 9.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable, and let x^* be a critical point, i.e., $\nabla f(x^*) = 0$. Assume that for some $G \geq 0$ the following condition holds:*

$$\left\|H^{-1}(x)H(y) - I_n\right\|_2 \leq 2G\|x - y\| \quad \forall x, y : \|y - x^*\| \leq \|x - x^*\| \leq \frac{1}{G}\,. \tag{9.10}$$

Then, for any starting point $x^{(0)}$ with $\|x^{(0)} - x^*\| \leq 1/G$, the iterates of Newton's method satisfy

$$\|x^{(t+1)} - x^*\| \leq G\|x^{(t)} - x^*\|^2\,.$$

Let us compare the condition to the one in Theorem 9.1. Assume we want to find a critical point x^* of a univariate function $g : \mathbb{R} \to \mathbb{R}$, and let $f(x) = g'(x)$. Thus, x^* is a root of f. Then, (9.10) can be written as $|f'(y)/f'(x) - 1| \leq 2G|x - y|$ for every x, y such that $|y - x^*| \leq |x - x^*| \leq 1/G$. We rewrite this as $|(f'(y) - f'(x))/f'(x)| \leq 2G|x - y|$, and note that by the mean value theorem for the continuous function $f'(x)$, there exists $w \in [x, y]$ (or $w \in [y, x]$) such that $f''(w) = (f'(y) - f'(x))/(y - x)$. Hence, the condition becomes $|f''(w)/f'(x)| \leq 2G$, the same expression as in (9.2). Note also that $|y - x^*| \leq |x - x^*|$ implies $|w - x^*| \leq |x - x^*|$.

If $f(x) = x^\top Qx + \langle p, x\rangle$ is a quadratic function, then $H(x) = 2Q$ for every $x \in \mathbb{R}^n$, giving $H^{-1}(x)H(y) = I_m$. The condition in the theorem holds with $G = 0$ (and $1/G = \infty$), consistently with the earlier observation that for arbitrary $x^{(0)}$, Newton's method finds the optimal solution in a single iteration.

Proof of Theorem 9.5. It suffices to show the statement for $t = 0$. That implies $\|x^{(1)} - x^*\| \leq G\|x^{(0)} - x^*\|^2 \leq 1/G$, and hence we obtain the statement inductively for every iterate. Condition (9.10) requires $H(x)$ to be nonsingular for every $\|x - x^*\| \leq 1/G$; hence, the Newton iterates are well-defined.

Let us define $\varphi : [0, 1] \to \mathbb{R}^n$ by $\varphi(\tau) = \nabla f(x + \tau(y - x))$. Hence, $\varphi(0) = \nabla f(x)$, and $\varphi(1) = \nabla f(y)$. We have

$$\nabla\varphi(\tau) = H(x + \tau(y - x))(y - x)\,,$$

as the directional second derivative vector. We apply the fundamental theorem of calculus (Theorem 1.13) in each coordinate to get

$$\nabla f(y) - \nabla f(x) = \varphi(1) - \varphi(0) = \int_0^1 \nabla\varphi(\tau)d\tau = \int_0^1 H(x + \tau(y - x))(y - x)d\tau\,. \tag{9.11}$$

(Recall that the integral of an $\mathbb{R} \to \mathbb{R}^n$ function is a vector in $\mathbb{R}^n$ that can be obtained by integrating coordinate-wise.) We use this equation, along with $\nabla f(x^*) = 0$ to bound

$$\begin{aligned}
x^{(1)} - x^* &= x^{(0)} - x^* - H^{-1}(x^{(0)})\nabla f(x^{(0)}) \\
&= x^{(0)} - x^* + H^{-1}(x^{(0)})\left(\nabla f(x^*) - \nabla f(x^{(0)})\right) \\
&= x^{(0)} - x^* + H^{-1}(x^{(0)})\int_0^1 H(x^{(0)} + \tau(x^* - x^{(0)}))(x^* - x^{(0)})d\tau \\
&= \int_0^1 \left(H^{-1}(x^{(0)})H\left(x^{(0)} + \tau(x^* - x^{(0)})\right) - I_n\right)(x^* - x^{(0)})d\tau
\end{aligned} \tag{9.12}$$

We use (9.10) to bound $\|x^{(1)} - x^*\|$. We see that

$$\begin{aligned}
\left\|x^{(1)} - x^*\right\| &\leq \int_0^1 \left\|\left(H^{-1}(x^{(0)})H\left(x^{(0)} + \tau(x^* - x^{(0)})\right) - I_n\right)(x^* - x^{(0)})\right\| d\tau \\
&\leq \int_0^1 \left\|H^{-1}(x^{(0)})H\left(x^{(0)} + \tau(x^* - x^{(0)})\right) - I_n\right\|_2 \cdot \left\|x^* - x^{(0)}\right\| d\tau \\
&\leq \int_0^1 2G\left\|\left(x^{(0)} + \tau(x^* - x^{(0)})\right) - x^{(0)}\right\| \cdot \left\|x^* - x^{(0)}\right\| d\tau \\
&= 2G\left\|x^* - x^{(0)}\right\|^2 \int_0^1 \tau d\tau = G\left\|x^* - x^{(0)}\right\|^2,
\end{aligned} \tag{9.13}$$

completing the proof. □

Corollary 9.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable m-strongly convex function and assume the Hessian $H(x)$ is B-Lipschitz, that is,*

$$\|H(x) - H(y)\|_2 \leq B\|x - y\| \quad \forall x, y \in \mathbf{dom}\, f.$$

If $\|\nabla f(x^{(0)})\| \leq 2m^2/B$ for the starting point $x^{(0)}$, then the iterates of Newton's method satisfy

$$\left\|x^{(t+1)} - x^*\right\| \leq \frac{B}{2m}\left\|x^{(t)} - x^*\right\|^2 .$$

Proof. Recall that m-strong convexity is equivalent to $H(x) \succeq mI_n$ for all $x \in \mathbf{dom}\, f$ (Theorem 4.9). This implies that $H^{-1}(x) \preceq \frac{1}{m}I_n$. (To see this, recall that $A \succeq mI_n$ is equivalent to the smallest eigenvalue being $\geq m$, and $A \preceq \frac{1}{m}I_m$ is equivalent to the largest eigenvalue being $\leq \frac{1}{m}$. Further, the eigenvalues of A^{-1} are the inverses of the eigenvalues of A.)

According to Lemma 9.4(iii), $\|H^{-1}(x)\|_2 \leq \frac{1}{m}$ for any $x \in \mathbf{dom}\, f$. For any $x, y \in \mathbf{dom}\, f$, we see that

$$\|H^{-1}(x)H(y) - I_n\|_2 = \|H^{-1}(x)(H(y) - H(x))\|_2 \leq \|H^{-1}(x)\|_2 \cdot \|H(y) - H(x)\|_2 \leq \frac{B}{m}\|x - y\|,$$

where the first inequality uses Lemma 9.4(ii). This verifies (9.10) for $G = B/(2m)$. We need one more bound that relates the distance to optimality and the gradient:

Claim 9.7. *If f is m-strongly convex and differentiable with minimiser x^*, then*

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

Proof. We again use strong convexity. Proposition 4.11 shows that

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

On the other hand, using $\nabla f(x^*) = 0$ we get

$$f(x) - f(x^*) = \langle \nabla f(x^*), x - x^* \rangle + D_f(x, x^*) \geq \frac{m}{2}\|x - x^*\|^2 .$$

This implies the claim. □

From this claim and the assumption $\|\nabla f(x^{(0)})\| \leq 2m^2/B$, we get $\|x^{(0)} - x^*\| \leq 2m/B$. Hence, Theorem 9.5 is applicable with $G = B/(2m)$. □

9.5.1 Affine invariant conditions on convergence

The limitation of the above analysis is that the conditions in Theorem 9.5 and Corollary 9.6 are *not* affine invariant: they may hold under certain affine preconditionings but not others, even though the sequence of steps in Newton's method remains identical under all such preconditionings.

Below, we formulate the affine invariant conditions on quadratic convergence. To get some intution, observe that Theorem 9.5 bounds the stability of the Hessian matrix $H(x)$: if x and y are near, then $H^{-1}(x)H(y)$ should be close to the identity matrix I_n in spectral norm; in other words, $H(x) \approx H(y)$. To be more precise, let $G < 1/2$; according to Lemma 9.4(iii), the condition in Theorem 9.5 is equivalent to saying that the eigenvalues of $H^{-1}(x)H(y) - I_n$ are between $-2G\|x - y\|$ and $2G\|x - y\|$, or equivalently,

$$-2G\|x - y\|I_n \preceq H^{-1}(x)H(y) - I_n \preceq 2G\|x - y\|I_n .$$

We can rearrange this as

$$(1 - 2G\|x - y\|)\, I_n \preceq H^{-1}(x)H(y) \preceq (1 + 2G\|x - y\|)\, I_n ,$$

that leads to

$$(1 - 2G\|x - y\|)\, H(x) \preceq H(y) \preceq (1 + 2G\|x - y\|)\, H(x) .$$

The affine invariant conditions formulate the analogous requirement, but using the local norm instead of the ℓ_2-norm. Please refer to [6, Section 9.6] for a proof. Instead of distance from optimality, the convergence is measured in the Newton-decrement.

Theorem 9.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable. Assume that*

$$(1 - 3\|x - y\|_x)\, H(x) \preceq H(y) \preceq (1 + 3\|x - y\|_x)\, H(x) \quad \forall x, y \in \mathbf{dom}\, f : \|x - y\|_x \leq 1/6\,.$$

If $\lambda_f(x^{(0)}) \leq 1/6$ for the starting point $x^{(0)}$, then

$$\|\lambda_f(x^{(t+1)})\| \leq 3\|\lambda_f(x^{(t)})\|^2$$

holds in each iteration of Newton's algorithm.

The condition in the theorem holds for the important class of *self-concordant* functions, introduced by Nesterov and Nemirovski. We do not define this class here but mention their key importance in the context of *interior point methods*; see e.g. [6, Chapters 10 and 11].

9.6 The damped Newton method

[Non examinable]

We gave sufficient conditions for quadratic convergence of Newton's method. However, this requires being close enough to the optimum, and from a further starting point, the iterations may not converge at all, as already seen in the example of root finding. This can be addressed by making shorter steps in the Newton direction, namely, using updates

$$x^{(t+1)} = x^{(t)} - \eta_t N(x^{(t)})\,,$$

for a step-size $\eta_t \in [0, 1]$. Recall $N(x^{(t)}) = H_f^{-1}(x^{(t)})\nabla f(x^{(t)})$. The best option would be *exact line search* that determines $\eta_t = \arg\min_\eta f(x^{(t)} - \eta N(x^{(t)}))$. This may be computationally too expensive, as already was the case for gradient methods.

Backtracking line search A fast alternative to exact line search that is able to calibrate the step size is backtracking line search; this is applicable for gradient descent as well as Netwon's method. We use a search direction $v \in \mathbb{R}$; this will be $v = -\nabla f(x^{(t)})$ in gradient descent and $v = N(x^{(t)})$ for Newton's method.

We fix two parameters: $0 < \alpha < \frac{1}{2}$, and $0 < \beta < 1$. Instead of finding the optimal step size, we wish to obtain η such that

$$f(x^{(t)} + \eta v) \leq f(x^{(t)}) + \alpha\eta \left\langle \nabla f(x^{(t)}), v \right\rangle\,. \tag{9.14}$$

From the first-order Taylor approximation around, we see that this must be true for small enough $\eta > 0$. We wish to approximately identify the largest value of η where this holds.

This can be done by starting form $\eta = 1$, and as long as (9.14) is not satisfied, we decrease η by a factor β. This can be formally described as follows.

1. $\eta \leftarrow 1$.
2. *While* $f(x^{(t)} + \eta v) > f(x^{(t)}) + \alpha\eta \langle \nabla f(x), v\rangle$, *update* $\eta \leftarrow \beta\eta$.

Setting the value of β gives a trade-off between speed and accuracy. For larger values of β (that is, close to 1), we might need several calibrating iterations. On the other hand, a smaller value (e.g. $\beta = 0.3$) provides fast convergence, but the resulting t might be a factor $1/\beta$ worse than the best choice for (9.14). The value of α is typically chosen between 0.01 and 0.3.

For Newton's method, i.e., $v = N(x^{(t)})$ the condition (9.14) can be written as

$$f(x^{(t)} + \eta v) \leq f(x^{(t)}) - \alpha\eta\lambda_f^2(x^{(t)})\,. \tag{9.15}$$

Description of the algorithm We are ready to describe the damped Newton method.

DAMPED NEWTON METHOD
Input: A twice differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$, a starting point $x^{(0)} \in \mathbf{dom}\, f$, parameters $\alpha \in (0, 1/3), \beta \in (0, 1)$, and accuracy requirement $\varepsilon > 0$.
Output: A solution $x^{(t)}$ with Newton-decrement $\lambda_f(x^{(t)}) < \varepsilon$.
Repeat
  Compute the Newton direction $N(x^{(t)}) = -H_f^{-1}(x^{(t)})\nabla f(x^{(t)})$ and decrement $\lambda_f^2(x^{(t)}) = \nabla f(x^{(t)})^\top H_f^{-1}(x^{(t)})\nabla f(x^{(t)})$;
  If $\lambda_f(x^{(t)}) < \varepsilon$ then terminate returning $x^{(t)}$.
  Use backtracking line-search with parameters α and β to determine $\eta_t \in (0, 1)$;
  $x^{(t+1)} = x^{(t)} - \eta_t N(x^{(t)})$;

9.6.1 Convergence analysis

We consider the case when f is both m-strongly convex and M-smooth:

$$mI_n \prec H(x) \leq MI_n \quad \forall x \in \mathbf{dom}\, f\,.$$

Further, we require that the Hessian $H(x)$ is B-Lipschitz, that is,

$$\|H(x) - H(y)\|_2 \leq B\|x - y\| \quad \forall x, y \in \mathbf{dom}\, f.$$

As noted above, these conditions are not affine invariant; an affine invariant analysis based on self-concordance is given e.g. in [1, Section 9.6.4].

Let us connect the termination criterion to distance from optimality.

Lemma 9.9. *For a function $f : \mathbb{R}^n \to \mathbb{R}$ as above,*

$$\frac{m}{\sqrt{M}}\|x - x^*\| \leq \lambda_f(x) \leq \frac{M}{\sqrt{m}}\|x - x^*\|$$

Proof. By assumption, $mI_n \preceq H(x) \preceq MI_n$, and therefore $\frac{1}{M}I_n \preceq H^{-1}(x) \preceq \frac{1}{m}I_n$. Hence,

$$\frac{1}{M}\|\nabla f(x)\|^2 \leq \lambda_f^2(x) = \nabla f(x)^\top H^{-1}(x)\nabla f(x) \leq \frac{1}{m}\|\nabla f(x)\|^2$$

Claim 9.7 implies the first inequality in the statement. For the second inequality, we use that by definition of M-smoothness,

$$\|\nabla f(x)\| = \|\nabla f(x) - \nabla f(x^*)\| \leq M\|x - x^*\|\,.$$

□

We define parameters

$$\delta = \frac{2m^2}{B} \quad \text{and} \quad \gamma = \alpha\beta\frac{m^5}{M^2B}\,.$$

(The precise values are not very important; these are constants determined by α, β, m, M and B). The key lemma of the analysis is the following:

Lemma 9.10. *For a function $f : \mathbb{R}^n \to \mathbb{R}$ as above, in every iteration of the damped Newton method,*

(i) If $\|\nabla f(x^{(t)})\| \geq \delta$, then

$$f(x^{(t+1)}) - f(x^{(t)}) \leq -\gamma\,.$$

(ii) If $\|\nabla f(x^{(t)})\| \leq \delta$*, then* $\eta_t = 1$ *and*

$$\left\|x^{(t+1)} - x^*\right\| \leq \frac{B}{2m}\left\|x^{(t)} - x^*\right\|^2 .$$

for all subsequent iterates.

Note that we have already proved the bound in part (ii) in Corollary 9.6. See [1, Section 9.5.3] for the proof of part (i) and for the proof of $\eta_t = 1$ in part (ii). In light of this lemma, the algorithm comprises two phases.

1. In the *damped Newton phase*, step-sizes $\eta_t \leq 1$ are chosen, and the optimality gap $f(x^{(t)}) - f(x^*)$ decreases by γ. Hence, the total number of such iterations is at most $(f(x^{(0)}) - f(x^*))/\gamma$.

2. In the *pure Newton phase*, full steps with $\eta_t = 1$ are used, and we experience quadratic convergence. In $C \log\log(Mm/(B\varepsilon))$ steps for some $C > 0$ we get to an iterate $\left\|x^{(t)} - x^*\right\| \leq \frac{\sqrt{m}}{M}\varepsilon$, and consequently, $\lambda_f(x^{(t)}) \leq \varepsilon$ by Lemma 9.9.

Thus, Lemma 9.10 yields the following running time bound:

Theorem 9.11. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a twice differentiable,* m*-strongly convex,* M*-smooth function, and further assume the Hessian is* B*-Lipschitz. Let* x^* *denote the minimiser of* f *and let* $x^{(0)} \in \mathbf{dom}\, f$ *be a given starting point. Then, the damped Newton method finds a* ε*-accurate solution (with* $\lambda_f(x^{(t)}) < \varepsilon$*) in*

$$\frac{M^2 B}{\alpha\beta m^5}(f(x^{(0)}) - f(x^*)) + C \log\log\left(\frac{Mm}{B\varepsilon}\right)$$

for some $C > 0$.

The log log function is extremely slow growing. In practice, one never needs more than 6 iterations in the pure Newton phase to get a highly accurate solution.

9.6.2 Comparison with gradient descent

The damped Newton phase could also be replaced by gradient descent with comparable convergence guarantees. Newton's algorithm is typically faster even in this phase. However, each iteration of Newton's algorithm requires significantly more computation than a gradient update. We do not only need to compute the Hessian $H(x)$, but also solve the system of linear equations $H(x)q = \nabla f(x)$. A naïve approach with Gaussian elimination would give n^3 arithmetic operations per iteration (up to constant overhead). There are better algorithms known, the theoretical current best improving the exponent of n to roughly 2.37. Still, this is significant extra work compared to a simple gradient step.

Computing the Hessian itself might be a difficult task. *Quasi-Newton methods* relax the requirement of using the exact Hessian, and instead use an estimate based on the previous iterates of the algorithm. We do not discuss these here; a gentle introduction can be found in [2, Chapter 8].

Bibliography

[1] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. https://web.stanford.edu/~boyd/cvxbook/.

[2] B. Gärtner and M. Jaggi. Optimization for machine learning, 2021. https://github.com/epfml/OptML_course.

[3] E. Hazan. Introduction to online convex optimization, 2021. https://arxiv.org/abs/1909.05207.

[4] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013. https://www.statlearning.com/.

[5] G. Lan. *First-order and stochastic optimization methods for machine learning*. Springer Nature, 2020. https://cpn-us-w2.wpmucdn.com/sites.gatech.edu/dist/f/330/files/2019/08/LectureOPTML.pdf.

[6] N. K. Vishnoi. *Algorithms for convex optimization*. Cambridge University Press, 2021. https://convex-optimization.github.io/ACO-v1.pdf.