Optimization Algorithms for Machine Learning

CONTENTS

1 Introduction **1**

1.1 Optimization and machine learning 2

1.1.1 Regression 3

1.1.2 Support Vector Machines 4

1.1.3 Neural Networks 5

2 Preliminary algebra and convex analysis **7**

2.1 Vectors 7

2.2 Matrices 8

2.3 Functions 9

2.4 Convex sets 11

2.4.1 Operations that preserves convexity of sets 16

2.5 Convex functions 17

2.5.1 First-order conditions for convex functions 18

2.5.2 Second-order conditions for convex functions 18

2.5.3 Operations that preserves convexity of functions 19

3 Unconstrained optimization models **21**

3.1 Optimization algorithms for unconstrained minimization 24

3.1.1 Gradient method 24

3.1.2 Fast Gradient method 27

3.1.3 Newton method 38

1

Introduction

After the interior point era has reached its peak through the discovery of polynomial algorithms for linear/nonlinear convex programming, the continuous increase of the amounts of data in many scientific areas led to a low interest for the second order methods. Moreover, as the new developments in computing technology and ubiquity of digital devices lead to optimization problems of extremely big sizes, larger memory space and computational power resources are required for implementation of any algorithmic scheme. Thus the return to simple algorithms with computationally cheap iterations has been seen as an imperative solution. However, even the most simple first order iterative methods, relying in many cases only on matrix-vector multiplications, have difficulties in execution and must be revisited. Indeed, in the last decades, the size of the problems arising in machine learning, compressed sensing or predictive control became so large that it is necessary to decompose or approximate the big data optimization problems into smaller, more manageable, subproblems. Or in other words, to find efficient first order algorithms that use, at each iteration, instead of the full gradient vector (that involves in some cases a large matrix-vector multiplication) much simpler partial first order information involving cheap vector operations. On the other hand, motivated by applications such as compressed sensing or sparse learning, other strands of recent research revealed the need of finding sparse minimizers of some objective function. The sparsest minimizer of given objective function is the minimizer with the smallest number of non-zeros components, or equivalently with the smallest ℓ_0 quasinorm value. Due to the nonconvexity and nonsmoothness of the ℓ_0 quasinorm function, various convex relaxations, e.g. 1-norm $\|\cdot\|_1$, have been used in literature to obtain a sparse solution of an optimization problem. Actually, since the degrees of approximation or equivalences between ℓ_0 and its relaxations are not fully understood, the design of algorithms for the ℓ_0-regularized problems are sometimes recommended.

Another facet of the dimensional difficulties is the number of constraints in a general optimization problem, which appears in many practical applications and eliminate the possibility of calling simple primal algorithms. A widely known alternative to the primal first order algorithms, when it is difficult to project on the primal feasible set described by conic and convex constraints, is the duality approach. Instead of solving the primal problem and under the assumption that a bounded optimal Lagrange multiplier exists, dual optimization methods based on Lagrangian relaxation are used for solving the corresponding dual problem. Since there are several practical applications with large number of constraints (e.g. classification tasks) where a precise computational runtime estimate of the implemented

This is a Book Title Name of the Author/Editor optimization algorithm is required, there is a need for tight theoretical complexity bounds for the dual first order algorithms. On the other hand, there are certain applications, such as those within big data settings, where the Slater condition (required by most of the dual algorithms) cannot be checked in realistic time and there is a need for simple primal algorithms which work even when there is no bounded optimal Lagrange multipliers.

1.1 Optimization and machine learning

Prediction/Regression. Andrew wants to determine how to rate a given (hyper)market (say ”Lidl”) by inspiring from his peer’s experiences.

	Market	Alex	Paul	Andrew	
x^1	Cora	1.5	2	2	(y_1)
x^2	Carrefour	3	1	1.5	(y_2)
x^t	Lidl	4	2		?

Based on the previous input ratings $\{x^1, x^2\} \subset \mathbb{R}^2$ of Paul and Alex and on the output rating $y \in \mathbb{R}^2$, we aim to predict Andrew’s rating for Lidl store in context of ratings $x^t \in \mathbb{R}^2$.

Classification. The handwritten digits MNIST database (see Fig. 1.1) is a storage with images of decimal digits 0 to 9. The data-set is based on gray-scale images of handwritten digits and, each image is 28 pixel in height and 28 pixel in width. Each pixel has a number associated with it, where 0 represents a dark pixel and, 255 represents a white pixel. Under these circumstances, the dataset is given by $\{x^i, y_i\}$, where the image is vectorized in $x^i \in \mathbb{R}^{784}$. The label y_i for each image represents the handwritten digit, thus $y_i \in \{0, 1, \cdots, 9\}$.

Anomaly detection. Given a set of financial transactions, detect the anomalous money transfers between a fixed number of clients.

Source Client	Destination Client	Amount	Date
A	B	200000	01.01.2020
Z	F	1050	02.01.2020
B	A	200000	07.01.2020
M	F	50	02.01.2020
A	B	700000	01.01.2020
N	W	70	01.01.2020
B	A	700000	07.01.2020

In a normal pool of usual transactions, the transfers between clients A and B become suspicious for fraud due to their back-and-forth closed circuit and to the uncommon large amount. In this context, x^i represents a transaction in the database and, for supervised models, y_i would represents a binary normality label.

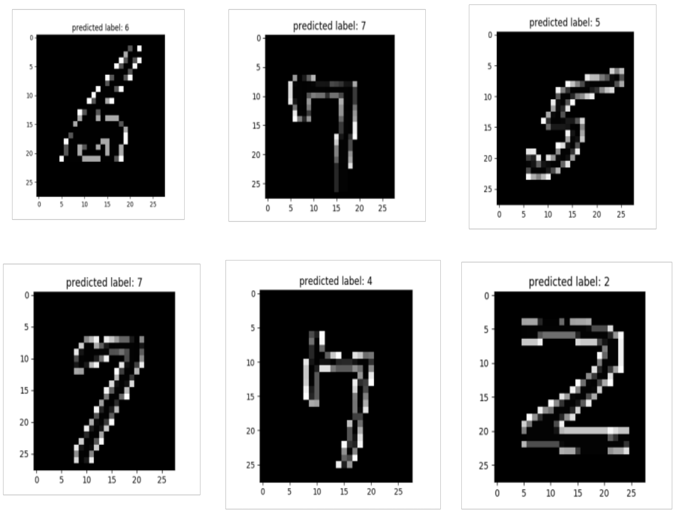


Figure 1.1: Samples from MNIST handwritten digits database. From: https://towardsdatascience.com/support-vector-machine-mnist-digit-classification-with-python-including-my-hand-written-digits-83d6eca7004a

Motivated by previous examples, further we limit ourselves to enumerate only three representative supervised learning models. However, the power of optimization modeling is not limited to supervised learning, but it handles all kind of learning techniques. Let the dataset $\{x^i, y_i\}_{i=1}^m \subset U \times V$ be a m-tuple sampled from an unknown probability distribution $\mathbb{P}$. In general, one desire to find a proper mathematical model $h : U \to V$ which guess the underlying laws of the experiment that generated the dataset, and thus it obeys:

$$h(x^i) \approx y_i \qquad \forall i = 1, \cdots, m$$

The form and complexity of *decision function* h would yields various machine learning models that we briefly expose below.

1.1.1 Regression

We approximate the hypothesis h with a linear function:

$$h(x) := h_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n,$$

where $\theta \in \mathbb{R}^n$ is the parameters vector. Thus, the linear regression models reduces to finding the optimal parameters that minimize the error between decision function output $h(x)$ and labels y:

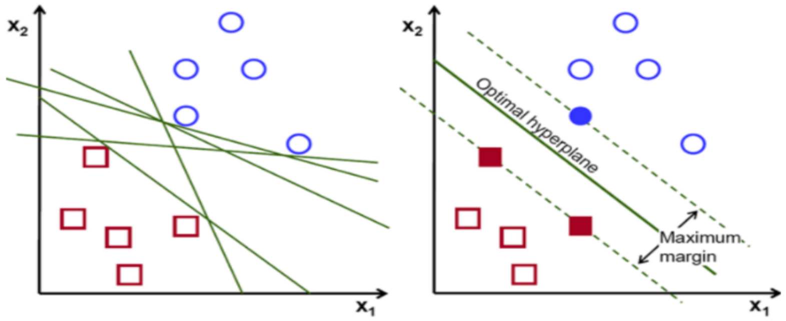
$$\min_{\theta \in \mathbb{R}^n} \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^i) - y_i)^2 \quad \equiv \quad \min_\theta \frac{1}{2m} \|X\theta - y\|_2^2$$

What is the explicit solution of this quadratic optimization model? Is it tractable for large-scale datasets?

1.1.2 Support Vector Machines

Related to binary classification, we present one of the most natural classification technique based on support vectors. Consider the training set $\{x^i, y_i\}_{i=1}^m$ with m examples and the labels $y_i \in \{-1, 1\}$. We assume that the classes are separable. Support Vector Machine model rely on finding a linear hyperplane that is able to distinguish between our classes of objects. Thus, it is aimed to find a linear function $h_{w,b}(x)$ such that:

$$h_{w,b}(x^i) := w^T x^i + b \begin{cases} < 0 & \text{if } y_i = -1 \\ \geq 0 & \text{otherwise.} \end{cases}$$



Obviously there is an infinite number of feasible hyperplanes under separability condition. However, we desire to select a certain hyperplane that achieves the best separation. First, notice that the distance from x^i to a given hyperplane $H = \{z \in \mathbb{R}^n : w^T z + b = 0\}$ is given by:

$$d_i = \frac{y_i(w^T x^i + b)}{\|w\|}$$

Since we seek a hyperplane placed at a maximal distance to any training point (i.e. maximal margin hyperplane), then we have to solve:

$$\max_{w,b} \min_{1 \leq i \leq m} d_i \quad \left(= \min_{1 \leq i \leq m} \frac{y_i(w^T x^i + b)}{\|w\|} \right)$$

It can easily observed that this is further equivalent with:

$$\max_{w,b,r} \frac{r}{\|w\|}$$
$$\text{s.t. } y_i(w^T x^i + b) \geq r.$$

The objective function of this model is nonconvex. Since convex optimization problems are more tractable, we further reduce the above problem to a convex one by observing that multiplying w, b and r by a scaling factor does not change the optimal value. Then we can then assume $r = 1$ and obtain:

$$\begin{aligned} &\max_{w,b} \frac{1}{\|w\|} \\ &\text{s.t.} \;\; y_i(w^T x^i + b) \geq 1. \end{aligned}$$

or in the minimization form:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \;\; y_i(w^T x^i + b) \geq 1. \end{aligned}$$

1.1.3 Neural Networks

2

Preliminary algebra and convex analysis

The present book relies heavily on the concepts and techniques of matrix algebra, mathematical analysis and optimization.

2.1 Vectors

We usually work in the space $\mathbb{R}^n$ composed by column vectors, represented by a lower case letter. For $x, y \in \mathbb{R}^n$ denote the standard Euclidean inner product and the Euclidean norm as:

$$\langle x, y \rangle = x^T y \quad \text{and} \quad \|x\| = (x^T x)^{1/2}.$$

We use the same notation $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ for spaces of different dimension. The orthogonal projection of x onto the set X and the distance from x to the set X is defined as:

$$[x]_X = \arg\min_{z \in X} \|x - z\| \quad \text{and} \quad \text{dist}_X(x) = \min_{z \in X} \|x - z\|.$$

For simplicity, given the integer $N \in \mathbb{N}$, we use notation $[N] = \{1, \dots, N\}$. Let us consider a decomposition of the dimension of the variable in N blocks: $n = \sum_{i=1}^N n_i$. We divide the identity matrix I_n into N block matrices:

$$I_n = \begin{bmatrix} U_1 & \cdots & U_N \end{bmatrix}, \quad U_i \in \mathbb{R}^{n \times n_i}, \quad i \in [N].$$

For any $x \in \mathbb{R}^n$ we denote with x_i the ith block component of vector x of dimension $n_i \geq 1$. Thus, $x = \begin{bmatrix} x_1^T \cdots x_N^T \end{bmatrix}^T \in \mathbb{R}^n$ can be written as:

$$x = \sum_{i=1}^N U_i x_i, \quad x_i \in \mathbb{R}^{n_i}, \quad i \in [N].$$

We also use $x_{ij} \in \mathbb{R}^{n_i + n_j}$ to denote the vector:

$$x_{ij} = \begin{bmatrix} x_i \\ x_j \end{bmatrix} \quad \forall (i, j) \in [N] \times [N].$$

At some point we need to refer to scalar components from the vector x or single columns of I_n and thus, we denote with $x_{(i)}$, the ith scalar component of vector x and $U_{(i)}$ the ith column of matrix I_n, respectively. For the support of vector x, namely the index set of the nonzero components of x, we use notation $I(x)$ and, for the complementary set of the support, $I^c(x)$. The fundamental Cauchy-Schwarz inequality states that for any inner product and the corresponding induced norm the following inequality holds:

$$|\langle x, y\rangle| \leq \|x\|\|y\| \;\; \forall\, x, y \in \mathbb{R}^n,$$

with equality if and only if x and y are linearly dependent. Furthermore, any norm $\|\cdot\|$ on $\mathbb{R}^n$ has a *dual norm*, notation $\|\cdot\|^*$, defined as:

$$\|y\|^* = \max_{\|x\|=1} \langle x, y\rangle.$$

2.2 Matrices

For a square matrix $Q \in \mathbb{R}^{n\times n}$ with entries Q_{ij}, a scalar $\lambda \in \mathbb{C}$ and a non-zero vector x that satisfy the equation $Qx = \lambda x$ is called an *eigenvalue* and *eigenvector* of Q, respectively. The eigenvalue-eigenvector equation may be written equivalently as

$$(\lambda I_n - Q)x = 0, \;\; x \neq 0,$$

i.e. the matrix $\lambda I_n - Q$ is singular, that is,

$$\det(\lambda I_n - Q) = 0.$$

Therefore, the *characteristic polynomial* of Q is defined as:

$$p_Q(\lambda) = \det(\lambda I_n - Q).$$

Clearly the set of roots of $p_Q(\lambda) = 0$ coincides with the set of eigenvalues of Q. The set of all eigenvalues of Q is called the *spectrum* of Q. For a symmetric matrix $Q \in \mathbb{R}^{n\times n}$ the corresponding eigenvalues are real and thus we consider the following order of its spectrum: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, referring to a certain eigenvalue i as $\lambda_i(Q)$. We denote with S^n the vector space of symmetric matrices:

$$S^n = \{Q \in \mathbb{R}^{n\times n} : Q = Q^T\}.$$

A symmetric matrix $Q \in S^n$ is *positive semidefinite*, notation $Q \succeq 0$, if:

$$Q \succeq 0 \;\text{ if }\; x^T Q x \geq 0 \quad \forall x \in \mathbb{R}^n.$$

A symmetric matrix $Q \in S^n$ is *positive definite*, notation $Q \succ 0$, if $x^TQx > 0$ for all $x \in \mathbb{R}^n, x \neq 0$. For two symmetric matrices $P, Q \in S^n$, we say that $Q \succeq P$ if $Q - P \succeq 0$. We denote the set of positive (semi) definite matrices with $(S^n_+)S^n_{++}$. We have the following characterization of a positive semidefinite matrix: the matrix Q is positive semidefinite if and only if all the eigenvalues of Q are non-negative. We have the following expressions for computing the smallest and the largest eigenvalue of a symmetric matrix $Q \in S^n$:

$$\lambda_1 = \min_{x\in\mathbb{R}^n:\, x\neq 0} \frac{x^TQx}{x^Tx} \;\text{ and }\; \lambda_n = \max_{x\in\mathbb{R}^n:\, x\neq 0} \frac{x^TQx}{x^Tx}.$$

We conclude that:

$$\lambda_1 I_n \preceq Q \preceq \lambda_n I_n.$$

Furthermore, the second smallest eigenvalue λ_2 of a symmetric matrix Q can be computed from Courant-Fischer theorem as follows: let a be an eigenvector corresponding to eigenvalue λ_1, then λ_2 is given by the following expression

$$\lambda_2 = \min_{x \in \mathbb{R}^n:\, x \perp a,\, x \neq 0} \frac{x^T Q x}{x^T x}.$$

Given a matrix $Q \in \mathbb{R}^{m \times n}$, we denote its nullspace by $Null(Q)$. We can derive definitions for certain matrix norms from vector norms. Given a vector norm $\| \cdot \|$, we define the corresponding matrix norm as:

$$\|Q\| = \max_{x \neq 0} \frac{\|Qx\|}{\|x\|}.$$

For the Euclidean norm the corresponding matrix norm (spectral norm) is as follows:

$$\|Q\| = \left(\lambda_{\max}(Q^T Q)\right)^{1/2}.$$

The Frobenius norm of a matrix is defined as:

$$\|Q\|_F = (\sum_{i=1}^{m} \sum_{j=1}^{n} Q_{ij}^2)^{1/2}$$

2.3 Functions

It is important to extend a function f to all $\mathbb{R}^n$ by defining its value to be $+\infty$ outside its domain. In the following we assume that all functions are implicitly extended. A scalar function $f : \mathbb{R}^n \to \mathbb{R}$ has the *effective domain* the set

$$\mathrm{dom} f = \{x \in \mathbb{R}^n : f(x) < \infty\}.$$

The function f is said to be *differentiable* at a point $x \in \mathbb{R}^n$ if there exists a vector $g \in \mathbb{R}^n$ such that for all $y \in \mathbb{R}^n$:

$$f(x + y) = f(x) + \langle g, y \rangle + \mathcal{R}(\|y\|),$$

where the remainder satisfies $\lim_{y \to 0} \frac{\mathcal{R}(\|y\|)}{\|y\|} = 0$ and $\mathcal{R}(0) = 0$. The vector g is called the derivative or the gradient of f at the point x and is written as $\nabla f(x)$. In other words a function is differentiable at a point x if it admits a first-order linear approximation at x. It is clear that the gradient is uniquely determined and we define it as a column vector with components:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \cdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

The function f is said to be differentiable on a set $X \subseteq \mathrm{dom} f$ if it is differentiable at all points of X. Based on previous dimension decomposition, the partial gradient is defined as:

$$\nabla_i f(x) = U_i^T \nabla f(x) \quad \text{and} \quad \nabla_{(i)} f(x) = U_{(i)}^T \nabla f(x).$$

The quantity, whenever the limit exists:

$$f'(x;d) = \lim_{t \to +0} \frac{f(x+td) - f(x)}{t}$$

is called the *directional derivative* of f at x along direction d. Note that the directional derivative may exists for a non-differentiable function: e.g. for the function $f(x) = \|x\|$ we have that $f'(0;d) = \|d\|$, but this function is not differentiable at $x = 0$. If the function is differentiable, then:

$$f'(x;d) = \langle \nabla f(x), d \rangle.$$

A scalar function f on $\mathbb{R}^n$ is said to be *twice differentiable* at x if it is differentiable at x and we can find a symmetric matrix $H \in \mathbb{R}^{n \times n}$ such that:

$$f(x+y) = f(x) + \langle \nabla f(x), y \rangle + \frac{1}{2} x^T H x + \mathcal{R}(\|y\|^2) \quad \forall y \in \mathbb{R}^n,$$

where the remainder satisfies $\lim\limits_{y \to 0} \frac{\mathcal{R}(\|y\|^2)}{\|y\|^2} = 0$. The symmetric matrix H is called the *Hessian* and is denoted $\nabla^2 f(x)$. In conclusion, a function is twice differentiable at x if it admits a second-order quadratic approximation in a neighborhood of x. As for the gradient, the Hessian is unique, whenever it exists, and is a symmetric matrix with the components:

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

The function f is said to be twice differentiable on a set $X \subseteq \mathrm{dom} f$ if it is twice differentiable at all points of X. The Hessian can be seen as a derivative of the vector function ∇f:

$$\nabla f(x+y) = \nabla f(x) + \nabla^2 f(x) y + \mathcal{R}(\|y\|).$$

A function that is k times differentiable with the kth derivative continuous is said to belong to the class $\mathcal{C}^k$. For a differentiable function $g : \mathbb{R} \to \mathbb{R}$, we have the classical first-order Taylor approximation with mean value or integral mean value :

$$g(b) - g(a) = g'(\tau)(b-a) = \int_a^b g'(\tau) d\tau,$$

for some τ in the interval $[a,\ b]$. These equalities can be extended to any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ using the previous relations for the function $g(t) = f(x + t(y-x))$:

$$f(y) = f(x) + \langle \nabla f(x + \tau(y-x)), y - x \rangle \ \text{ for some } \tau \in (0,\ 1),$$

$$f(y) = f(x) + \int_0^1 \langle \nabla f(x + \tau(y-x)), y - x \rangle d\tau.$$

The reader should note that, using the rules for differentiability, we used:

$$g'(\tau) = \langle \nabla f(x + \tau(y-x)), y - x \rangle.$$

Some extensions are possible:

$$\nabla f(y) = \nabla f(x) + \int_0^1 \langle \nabla^2 f(x + \tau(y - x)), y - x\rangle d\tau,$$

$$f(y) = f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}(y - x)^T \nabla^2 f(x + \tau(y - x))(y - x), \text{ for some } \tau \in (0,\ 1).$$

A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ has a *Lipschitz continuous gradient* if there exists a constant $L_f > 0$ such that:

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

We usually refer to a function with Lipschitz continuous gradient as: *smooth function*. Using Taylor's approximations given above we obtain the following lemma:

Lemma 2.3.1 *(i) A twice differentiable function f has a Lipschitz continuous gradient if and only if the following inequality holds:*

$$\|\nabla^2 f(x)\| \leq L \ \forall x.$$

(ii) If a differentiable function has a Lipschitz continuous gradient then

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

Interpretation: From Lemma Lipschitz it follows that a differentiable function with a Lipschitz continuous gradient is bounded from above by a special quadratic function having the Hessian $\frac{1}{L} I_n$ (here I_n is the unit matrix in $\mathbb{R}^{n \times n}$):

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

A twice differentiable function has a *Lipschitz continuous Hessian* if there exists a constant $M > 0$ such that

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq M\|x - y\| \ \forall x, y.$$

For this class of functions we have the following characterization:

Lemma 2.3.2 *For a twice differentiable function f which has a Lipschitz continuous Hessian we have:*

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y - x)\| \leq \frac{M}{2}\|y - x\|^2 \ \forall x, y.$$

Moreover, the following inequality also holds:

$$-M\|x - y\| I_n \preceq \nabla^2 f(x) - \nabla^2 f(y) \preceq M\|x - y\| I_n \ \forall x, y.$$

2.4 Convex sets

Definition 2.4.1 *A set S is an affine set if for any two points $x_1, x_2 \in S$ and any $\alpha \in \mathbb{R}$ we have $\alpha x_1 + (1 - \alpha)x_2 \in S$ (i.e. the line generated by any two points from S is included in S).*

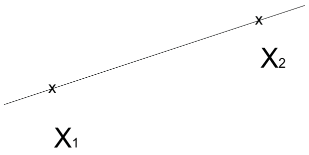


Figure 2.1: Affine set generated by two points x_1 and x_2.

Example 2.4.2 *The solution set of a linear system is an affine set, i.e. the set* $\{x \in \mathbb{R}^n : Ax = b\}$ *is affine.*

The *affine combination* of p points $x_1, \cdots, x_p$ is define as:

$$\sum_{i=1}^{p} \alpha_i x_i, \quad \text{where} \quad \sum_{i=1}^{p} \alpha_i = 1, \alpha_i \in \mathbb{R}$$

The *affine hull* of a set $S \subseteq \mathbb{R}^n$, denoted Aff(S), is the set containing all the possible affine combinations with points from S:

$$\text{Aff}(S) = \{ \sum_{i \in \mathcal{I}, \mathcal{I} \text{ finite}} \alpha_i x_i : x_i \in S, \sum_i \alpha_i = 1, \alpha_i \in \mathbb{R}\}.$$

In other words Aff(S) is the smallest affine set that contains S.

Definition 2.4.3 *The set* S *is called convex if for any two points* $x_1, x_2 \in S$ *and* $\alpha \in [0,\ 1]$ *we have* $\alpha x_1 + (1-\alpha)x_2 \in S$ *(i.e. the segment generated by any two points from* S *is included in S).*

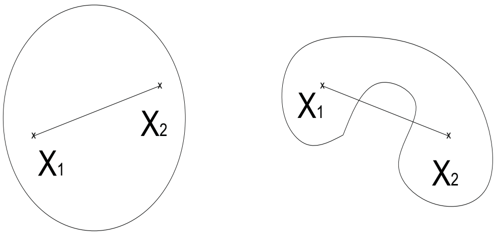


Figure 2.2: Convex set.

It follows immediately that any affine set is convex. Furthermore, the *convex combination* of p points $x_1, \cdots, x_p$ is defined as:

$$\sum_{i=1}^{p} \alpha_i x_i, \quad \text{where} \quad \sum_{i=1}^{p} \alpha_i = 1, \alpha_i \geq 0.$$

The *convex hull* of a set S, denoted Conv(S), is the set containing all possible convex combinations with points from S:

$$\text{Conv}(S) = \{ \sum_{i \in \mathcal{I}, \mathcal{I} \text{ finite}} \alpha_i x_i : x_i \in S, \sum_i \alpha_i = 1, \alpha_i \geq 0\}.$$

Note that the convex hull of a set is the smallest convex set that contains the given set. It follows that if S is convex, the convex hull of S coincides with S.

Theorem 2.4.4 (Caratheodory's Theorem) *If $S \subseteq \mathbb{R}^n$ is a convex set then every element of S is a convex combination of at most $n+1$ points of S.*

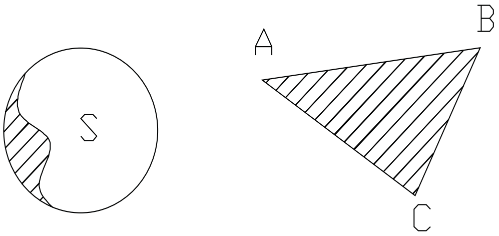


Figure 2.3: Convex hull.

An *hyperplane* is the convex set defined as

$$\{x \in \mathbb{R}^n : a^T x = b\}, \quad a \neq 0, b \in \mathbb{R}.$$

An *halfspace* is the convex set defined as

$$\{x \in \mathbb{R}^n : a^T x \geq b\} \quad \text{or} \quad \{x \in \mathbb{R}^n : a^T x \leq b\},$$

where $a \neq 0$ and $b \in \mathbb{R}$.

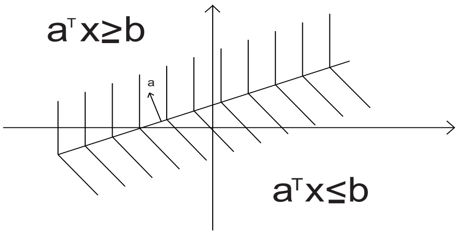


Figure 2.4: Hyperplane.

A *ball* with center $x_0 \in \mathbb{R}^n$ and ray $r > 0$ is a convex set defined as:

$$B(x_0, r) = \{x \in \mathbb{R}^n : \|x - x_0\| \le r\} = \{x \in \mathbb{R}^n : x = x_0 + ru, \|u\| \le 1\}.$$

An *ellipsoid* is the convex set defined as:

$$\{x \in \mathbb{R}^n : (x - x_0)^T Q^{-1}(x - x_0) \le 1\} = \{x_0 + Lu : \|u\| \le 1\},$$

where $Q \succ 0$ and $Q = L^T L$.

A *polyhedron* is the convex set described by a finite set of hyperplanes and/or halfspaces:

$$\{x \in \mathbb{R}^n : a_i^T x \le b_i \;\; i = 1, \cdots, m, c_j^T x = d_j \;\; j = 1, \cdots, p\}$$

A *polygon* is a bounded polyhedron. Another representation of a polyhedron is given in terms of vertices:

$$\{\sum_{i=1}^{n_1} \alpha_i v_i + \sum_{j=1}^{n_2} \beta_j r_j : \sum_i \alpha_i = 1, \alpha_i \ge 0, \beta_j \ge 0 \;\; \forall j\},$$

where v_i are called vertices and r_j are called affine rays.

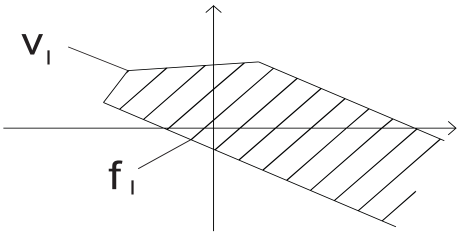


Figure 2.5: Unbounded polygon generated by vertices and affine rays.

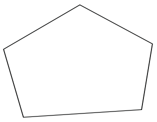


Figure 2.6: Bounded polygon.

Definition 2.4.5 *A set K is called cone if for any $x \in K$ and $\alpha \ge 0, \alpha \in \mathbb{R}$ we have $\alpha x \in K$. We say that K is a convex cone if K is a convex set and a cone.*

The *conic combination* of p points $x_1, \cdots, x_p$ is defined as:

$$\sum_{i=1}^{p} \alpha_i x_i, \quad \text{where} \quad \alpha_i \geq 0$$

The *conic hull* of a set S, denoted Con(S), is the set containing all possible conic combinations with elements from S:

$$\text{Con}(S) = \{ \sum_{i \in \mathcal{I}, \mathcal{I} \text{ finite}} \alpha_i x_i : x_i \in S, \alpha_i \geq 0\}$$

Note that the conic hull of a set is the smallest cone that contains the given set.

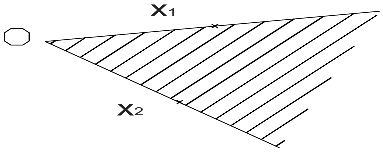


Figure 2.7: Conic hull generated by two points x_1 and x_2.

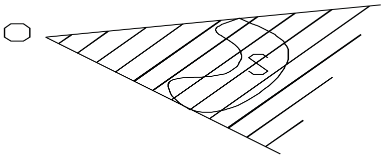


Figure 2.8: Conic hull generated by a set S.

For a given cone K (with an associated scalar product $\langle \cdot, \cdot \rangle$) its *dual cone*, denoted K^*, is defined as:

$$K^* = \{y : \langle x, y \rangle \geq 0, \forall x \in K\}.$$

Note that the dual cone is always a closed set. Using the fact that $\langle x, y \rangle = \|x\| \|y\| \cos \angle(x, y)$ we conclude that the angle between a vector from K and a vector from K^* is less that $\frac{\pi}{2}$. If the cone K satisfies the condition $K = K^*$, then we say that K is a *self-dual cone*.

Example 2.4.6 *The following sets are cones:*

- $\mathbb{R}^n$ *and* $(\mathbb{R}^n)^* = \{0\}$.
- $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x \geq 0\}$ *is called the orthant cone and is self-dual using the usual scalar product* $\langle x, y \rangle = x^T y$, *i.e.* $(\mathbb{R}^n_+)^* = \mathbb{R}^n_+$.

- $\mathcal{L}^n = \{[x^T\ t]^T \in \mathbb{R}^{n+1} : \|x\| \le t\}$ *is called the Lorenz cone or ice-cream cone and it is also self-dual with the scalar product* $\langle [x^T\ t]^T, [y^T\ v]^T\rangle = x^Ty + tv$, *i.e.* $(\mathcal{L}^n)^* = \mathcal{L}^n$.
- $S^n_+ = \{X \in S^n : X \succeq 0\}$ *is the semidefinite cone and is also self-dual with the scalar product* $\langle X, Y\rangle = Trace(XY)$, *i.e.* $(S^n_+)^* = S^n_+$.

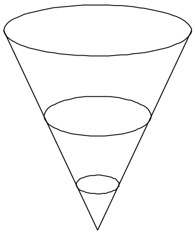


Figure 2.9: Ice cream cone.

2.4.1 Operations that preserves convexity of sets

- intersection of convex sets is convex, i.e. if the family of sets $\{S_i\}_{i\in\mathcal{I}}$ is convex, then $\bigcap_{i\in\mathcal{I}} S_i$ is also convex.
- sum of two convex sets S_1 and S_2 is also convex: $S_1 + S_2 = \{x + y : x \in S_1, y \in S_2\}$. Moreover, $\alpha S = \{\alpha x :, x \in S\}$ is convex if the set S is convex and $\alpha \in \mathbb{R}$.
- translation of a convex set S is also convex, i.e. given an affine function $f(x) = Qx + b$, the image of S through f, $f(S) = \{f(x) : x \in S\}$, is also convex. Similarly, the pre-image: $f^{-1}(S) = \{x : f(x) \in S\}$ is also convex.

Linear Matrix Inequalities (LMI):
It can be easily proved that the set of positive semidefinite matrices S^n_+ is convex. Let us now regard an affine map $G : \mathbb{R}^m \to S^n_+$, $G(x) = A_0 + \sum_{i=1}^m x_iA_i$, with symmetric matrices $A_0, \cdots, A_m \in S^n$. The expression

$$G(x) \succeq 0$$

is called a *linear matrix inequality* (LMI). It defines a convex set $\{x \in \mathbb{R}^m : G(x) \succeq 0\}$, as the pre-image of S^n_+ under the affine map $G(x)$.

Theorem 2.4.7 (Hyperplane separation theorem) *Let S_1 and S_2 be two convex sets such that $S_1 \cap S_2 = \emptyset$. Then, there exists an hyperplane that separates these two sets, i.e. there exists $a \neq 0$ and $b \in \mathbb{R}$ such that $a^Tx \ge b$ for all $x \in S_1$ and $a^Tx \le b$ for all $x \in S_2$.*

Theorem 2.4.8 (Hyperplane support theorem) *Let S be a convex set and $x_0 \in bd(S) = cl(S) - int(S)$. Then there exists a supporting hyperplane for S at x_0, i.e. there exists $a \neq 0$ such that $a^Tx \ge a^Tx_0$ for all $x \in S$.*

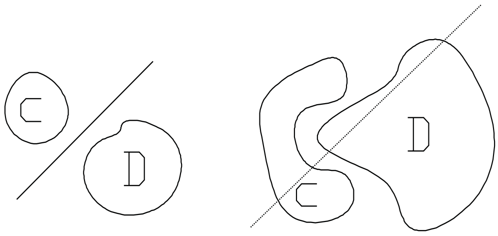


Figure 2.10: Separation theorem.

2.5 Convex functions

Definition 2.5.1 *The function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex if its effective domain $domf$ is a convex set and*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y),$$

for all $x, y \in domf$ and $\alpha \in [0,\ 1]$.

If

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y),$$

for all $x \neq y \in domf$ and $\alpha \in (0,\ 1)$, then f is called a strictly convex function.

If there is a constant $\sigma > 0$ such that

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) - \frac{\sigma}{2}\alpha(1-\alpha)\|x-y\|^2,$$

for all $x, y \in domf$ and $\alpha \in [0,\ 1]$, then f is called a strongly convex function.

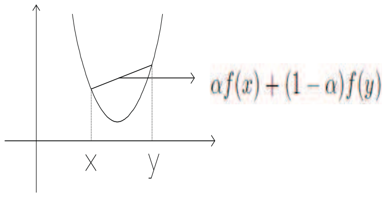


Figure 2.11: Convex function.

The Jensen inequality tells us that f is a convex function if and only if

$$f(\sum_{i=1}^{p} \alpha_i x_i) \leq \sum_{i=1}^{p} \alpha_i f(x_i)$$

for all $x_i \in \text{dom}f$ and $\alpha_i \in [0,\ 1]$, $\sum_i \alpha_i = 1$.

The geometrical interpretation of convexity is very simple. For a convex function the function values are below the corresponding chord, that is, the values of convex function at points on the line segment $\alpha x + (1-\alpha)y$ are less than or equal to the height of the chord joining the points $(x, f(x))$ and $(y, f(y))$.

Remark 2.5.2 A function is convex if and only if it is convex when restricted to any line that intersects its domain. Rephrased, f is convex if and only if for all $x \in \mathrm{dom} f$ and for all d, the function $g(\alpha) = f(x + \alpha d)$ is convex on $\{\alpha \in \mathbb{R} : x + \alpha d \in \mathrm{dom} f\}$. This property is very useful in testing whether a function is convex by restricting it to a line.

A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *concave* if $-f$ is convex.

2.5.1 First-order conditions for convex functions

Theorem 2.5.3 *(Convexity for C^1 functions) Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and dom$f$ is a convex set. Then f is convex if and only if*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \forall x, y \in dom f. \tag{2.1}$$

Proof:
"⇒" From the convexity of f we have that for any $x, y \in \mathrm{dom} f$ and for any $\alpha \in [0,\ 1]$:

$$f(x + \alpha(y - x)) - f(x) \leq \alpha(f(y) - f(x))$$

and therefore

$$\nabla f(x)^T (y - x) = \lim_{t \to 0} \frac{f(x + \alpha(y - x)) - f(x)}{\alpha} \leq f(y) - f(x).$$

"⇐" To prove that for $z = x + \alpha(y - x) = (1-\alpha)x + \alpha y$ holds that $f(z) \leq (1 - \alpha)f(x) + \alpha f(y)$ let us use (2.1) twice at z, in order to obtain $f(x) \geq f(z) + \nabla f(z)^T (x - z)$ and $f(y) \geq f(z) + \nabla f(z)^T (y - z)$ which yield, when weighted with $(1-\alpha)$ and α and added to each other

$$(1-\alpha)f(x) + \alpha f(y) \geq f(z) + \nabla f(z)^T \underbrace{[(1-\alpha)(x - z) + \alpha(y - z)]}_{=(1-\alpha)x + \alpha y - z = 0}.$$

The interpretation is simple: the tangents are below the graph for a convex function. A straightforward consequence of this theorem is the following statement: assume that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex, then

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0 \quad \forall x, y \in \mathrm{dom} f.$$

2.5.2 Second-order conditions for convex functions

Theorem 2.5.4 *(Convexity for C^2 Functions) Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable and dom$f$ is convex. Then f is convex if and only if for all $x \in$ dom$f$ the Hessian is positive semidefinite, i.e.*

$$\nabla^2 f(x) \succeq 0 \quad \forall x \in dom f. \tag{2.2}$$

Proof:
To prove (2.1) $\Rightarrow$ (2.2) we use a second order Taylor expansion of f at x in an arbitrary direction d:

$$f(x+td) = f(x) + \nabla f(x)^T dt + \frac{1}{2}t^2 d^T \nabla^2 f(x) d + o(t^2\|d\|^2).$$

From this we obtain

$$d^T \nabla^2 f(x) d = \lim_{t\to 0} \frac{2}{t^2} \underbrace{\left(f(x+td) - f(x) - t\nabla f(x)^T d\right)}_{\geq 0, \text{ because of (2.1).}} \geq 0.$$

Conversely, to prove (2.1) $\Leftarrow$ (2.2) we use the Taylor rest term formula with some $\theta \in [0,\ 1]$:

$$f(y) = f(x) + \nabla f(x)^T (y-x) t + \underbrace{\frac{1}{2}t^2 (y-x)^T \nabla^2 f(x+\theta(y-x))(y-x)}_{\geq 0, \text{ due to (2.2).}}.$$

Example 2.5.5

1. *The function $f(x) = -\log(x)$ is convex on $\mathbb{R}_+$ because $f''(x) = \frac{1}{x^2} > 0$ for all $x > 0$.*
2. *The quadratic function $f(x) = r + q^T x + \frac{1}{2}x^T Q x$ is convex on $\mathbb{R}^n$ if and only if $Q \succeq 0$, because $\forall x \in \mathbb{R}^n :\ \nabla^2 f(x) = Q$. Note that any affine function is convex and concave in the same time.*
3. *The function $f(x,t) = \frac{x^T x}{t}$ is convex on $\mathbb{R}^n \times (0,\ \infty)$ because its Hessian is positive definite. To see this, multiply it from left and right with $v = [z^T\ s]^T \in \mathbb{R}^{n+1}$ which yields $v^T \nabla f(x,t) v = \frac{2}{t^3}\|tz - sx\|^2 \geq 0$ if $t > 0$.*

Theorem 2.5.6 *(Convexity of sublevel sets) The sublevel set $\{x \in domf : f(x) \leq c\}$ of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to any constant $c \in \mathbb{R}$ is convex.*

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

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

Epigraph of a function
: Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. We define its *epigraph* as the following set:

$$\text{epi}f = \{[x^T\ t]^T \in \mathbb{R}^{n+1} : x \in \text{dom}f,\ \ f(x) \leq t\}.$$

Theorem 2.5.7 *(Convexity of epigraph) A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if its epigraph is a convex set.*

2.5.3 Operations that preserves convexity of functions

1. If f_1 and f_2 are convex functions and $\alpha_1, \alpha_2 \geq 0$ then $\alpha_1 f_1 + \alpha_2 f_2$ is also convex
2. If f is convex then $g(x) = f(Ax + b)$ (i.e. the composition of a convex function with an affine function) is also convex

3. Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be such that $f(\cdot, y)$ convex for any $y \in S \subseteq R^m$. Then the new function
$$g(x) = \sup_{y \in S} f(x, y)$$
is also convex.
4. The composition with a monotone convex function: if $f : \mathbb{R}^n \to \mathbb{R}$ is convex and $g : \mathbb{R} \to \mathbb{R}$ is convex and monotonically increasing, then the function $g \circ f : \mathbb{R}^n \to \mathbb{R},\ x \mapsto g(f(x))$ is also convex.

   Proof:
   $$\nabla^2 (g \circ f)(x) = \underbrace{g''(f(x))}_{\geq 0} \underbrace{\nabla f(x) \nabla f(x)^T}_{\succeq 0} + \underbrace{g'(f(x))}_{\geq 0} \underbrace{\nabla^2 f(x)}_{\succeq 0} \succeq 0.$$

Conjugate functions
: Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. We define its *conjugate*, denoted f^*, as the function

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \underbrace{y^T x - f(x)}_{F(x,y)}$$

From previous discussion it follows that f^* is convex regardless the properties of f. Moreover, $\text{dom} f^* = \{y : f^*(y) \text{ finite}\}$. Another straightforward consequence of the definition is *Fenchel inequality*:

$$f(x) + f^*(y) \geq y^T x \ \ \forall x, y.$$

Example 2.5.8 *For the convex quadratic function $f(x) = \frac{1}{2} x^T Q x$, where $Q \succ 0$, we have $f^*(y) = \frac{1}{2} y^T Q^{-1} y$.*

References

[1] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons, New York, 1983.

[2] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, USA, 2004.

[3] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[4] Y. Nesterov. Gradient methods for minimizing composite objective functions. *Mathematical Programming*, 140(1):125–161, 2013.

[5] Yu. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[6] Yu. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.