Lecture Note:

Numerical Optimization

Lecturer: Dr. Hai Yen LE

2020

Contents

Course Description:

The course introduces theory and numerical methods for continuous optimization problems (constrained and unconstrained). The goal is to provide the basics of numerical optimization methods and to enable the students to apply and adapt these methods to optimization problems that arise in multiple areas of science, engineering and business.

Learning outcomes:

- Define and use optimization terminology and concepts, and understand how to classify an optimization problem.
- Apply optimization methods to problems, including developing a model, defining an optimization problem, applying optimization methods, exploring the solution, and interpreting results. Students will demonstrate the ability to choose and justify optimization techniques that are appropriate for solving realistic problems.
- Understand and apply unconstrained optimization theory for continuous problems,

including the necessary and sufficient optimality conditions and algorithms such as: gradient, Newton's method, and quasi-Newton methods.

- Understand and apply constrained optimization theory for continuous problems, including the Karush-Kuhn-Tucker conditions and algorithms such as: gradient projection, sequential quadratic programming, and interior-point methods.

Prerequisites: Linear algebra, Calculus, Programming.

Main references:

1. A. Beck: *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*, SIAM, 2014.

2. J. Nocedal and S.J. Wright: *Numerical Optimization*, 2^{nd} edition, Springer-Verlag, 2006.

3. A. Ruszczynski, *Nonlinear Optimization*, Princeton University Press, 2006.

Nothing in the world takes place without optimization, and there is no doubt that all aspects of the world that have a rational basis can be explained by optimization methods. (Leonhard Euler , 1744).

1

Introduction

What is optimization?

Optimization is an important branch of applied mathematics. Its applications appear in every area of life and science. Optimization can be applied to design cars, airplanes, cellphones, laptop, ..., it can also be used in image reconstruction, portfolio optimization,

signal processing, classification, ...

There are three important elements in an optimization problem: decision variables, objective function, and constraints. Our goal is to find values of the variables that satisfy all the constraints and optimize the objective.

Some application problems

Classification: Given a set of data points $(x_1, y_1), \ldots, (x_m, y_m)$ where the vector $x_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$. The vector x_i, may be, for example measurements of test performed on patients or features extracted from emails, ... And $y_i = 1$ or 0 means a patient diagnosed with a disease or not, an email is 'spam' or 'non-spam', ...

Our goal is to find a linear classifier $\phi(x) = a^T x + b$ such that

$$\phi(x_i) = a^T x_i + b > 0 \text{ if } y_i = 1,$$

and

$$\phi(x_i) = a^T x_i + b < 0 \text{ if } y_i = 0.$$

This problems can be formulated as

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{i=1}^m z_i \\ \text{subject to} \quad & a^T x + b + z_i \geq 0 \quad \text{if } y_i = 1, \\ & a^T x + b - z_i \geq 0 \quad \text{if } y_i = 0, \\ & z \geq 0, \\ & \|a\| = 1. \end{aligned}$$

Matrix completion (The Netflix problem): There are about 10^6 users and 2500 movies on Netflix. The database of Netflix contains a matrix $A \in \mathbb{R}^{m\times n}$ where each entry A_{ij} is the rating of user i to the movie j. Most of the users have only seen a small part of the movies, and they only rates the movies that they have seen. So, many entries of the matrix A are unknown. The goal is to predict which movies a particular user might like. This means that we would like to complete matrix A based on the given entries. This problem is called matrix completion problem. Clearly, without additional assumptions we cannot recover all the entries of A. So, we assume that matrix A is low-rank. This problem is often encountered in the analysis of incomplete data sets exhibiting an underlying factor model with applications in collaborative filtering, computer vision, control.

Suppose that we are presented with a set of triples $(I(i), J(i), S(i))$ for $i = 1, \ldots, k$ and wish to find a matrix with $S(i)$ in the entry corresponding to row $I(i)$ and column $J(i)$ for all i. The matrix completion can be formulated as follows

$$\begin{aligned} \text{Minimize} \quad & \text{rank } A \\ \text{subject to} \quad & A_{I(i),J(i)} = S(i) \quad \forall i = 1, \ldots, k. \end{aligned}$$

2

Convex analysis

This chapter contains some basic notions and results in convex analysis, a part of optimization theory that studies properties of convex set and convex function and their applications in convex optimization problem. Futher results on this topic can be found in [1, 2, 6].

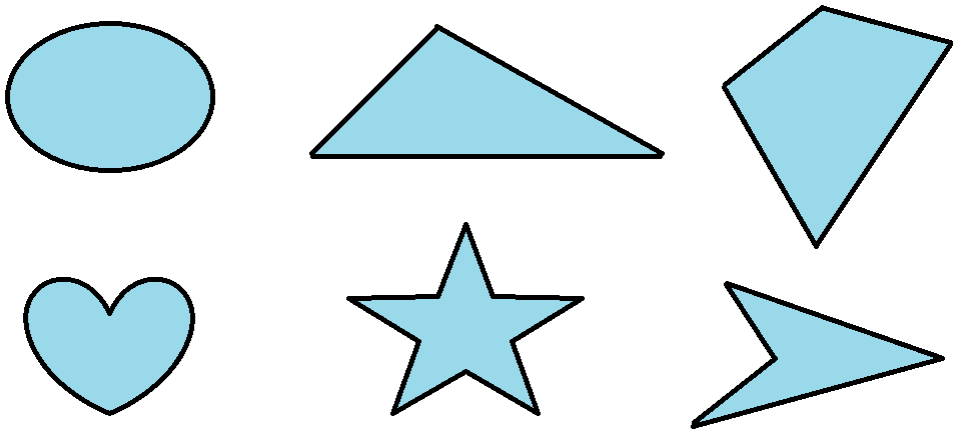


Figure 2.1: Convex and nonconvex sets

1 Convex Sets

Definition and Examples

Definition 2.1. *A set $C \subseteq \mathbb{R}^n$ is said to be convex if it contains all of its line-segments, that is for all $x, y \in C$ and $\lambda \in [0, 1]$*

$$\lambda x + (1 - \lambda)y \in C.$$

The point $\lambda x + (1 - \lambda)y$ is called a convex combination of x and y. Consequently, two arbitrary points in C can be linked by a continuous path. Examples of convex and non-convex sets in $\mathbb{R}^2$ are illustrated in Figure 2.1.

We will now show some basic examples of convex sets.

Example 2.2. *(Hyperplanes and half-spaces) Let $a \in \mathbb{R}^n, b \in \mathbb{R}$ and $a \neq 0$. Then the following sets are convex:*

- *the hyperplane* $H = \{x \in \mathbb{R}^n | \quad a^T x = b\}$,
- *the closed half-space* $H^{\leq} = \{x \in \mathbb{R}^n | \quad a^T x \leq b\}$,
- *the open half-space* $H^{<} = \{x \in \mathbb{R}^n | \quad a^T x < b\}$.

Example 2.3. *(Balls). Let* $c \in \mathbb{R}^n$ *and* $r > 0$*. Let* $\|.\|$ *be an arbitrary norm defined on* $\mathbb{R}^n$*. Then the open ball*

$$B(c, r) = \{x \in \mathbb{R}^n | \quad \|x - c\| < r\}$$

and closed ball

$$B[c, r] = \{x \in \mathbb{R}^n | \quad \|x - c\| \leq r\}$$

are convex.

Note that the above result is true for any norm defined on $\mathbb{R}^n$*. The* l_1, l_2*, and* l_∞ *unit balls are illustrated in Figure 2.2.*

Example 2.4. *(Ellipsoids) Let* $Q \in \mathbb{R}^{n \times n}$ *be a positive semi-definite matrix,* $b \in \mathbb{R}^n$*, and* $c \in \mathbb{R}$*. An ellipsoid* E*, defined by*

$$E := \{x \in \mathbb{R}^n | \quad x^T Q x + 2b^T x + c \leq 0\},$$

is convex.

Example 2.5. *(Simplices) The unit simplex in* $\mathbb{R}^n$ *defined by*

$$\triangle_n := \{x \in \mathbb{R}^n | \quad \sum_{i=1}^{n} x_i = 1, \quad x_i \geq 0 \quad \forall i = 1, \ldots, n\}.$$

is convex.

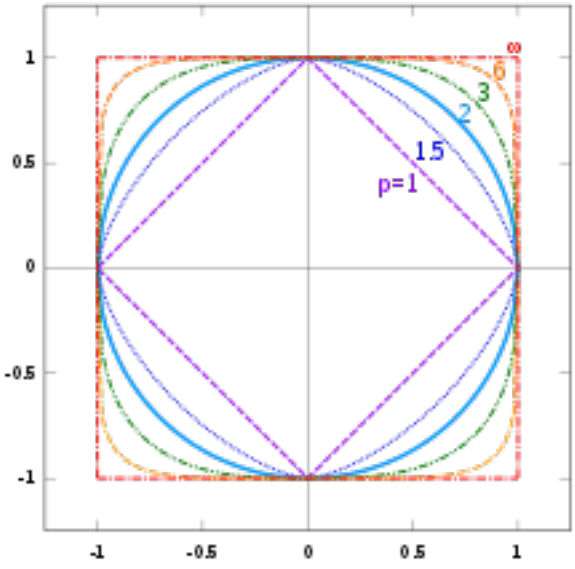


Figure 2.2: Different unit-balls

(Source: https://de.wikipedia.org/wiki/P-Norm)

Example 2.6. *(Convex cones) A set $K \subseteq \mathbb{R}^n$ is a cone if for any $x \in K$ and $\alpha \geq 0$, $\alpha x \in K$.*

A convex cone is a cone which is convex. It is easy to see that the set

$$\{x \in \mathbb{R}^n | \quad \langle s_j, x\rangle = 0 \quad \forall j = 1, ..., m, \quad \langle s_{m+j}, x\rangle \leq 0 \quad \forall j = 1, ..., p\}$$

where the s_j's are given in $\mathbb{R}^n$, is a convex cone in $\mathbb{R}^n$.

Convexity-Preserving Operations on Sets

Proposition 2.7. *Let $\{C_j\}_{j\in J}$ be an arbitrary family (possibly infinite) of convex sets. Then*

$$C := \bigcap_{j\in J} C_j$$

is convex.

Proof. Let $x, y \in C$ and $0 < \lambda < 1$. Since C_j is convex for any $j \in J$, we have $\lambda x + (1-\lambda)y \in C_j$. Therefore, $\lambda x + (1-\lambda)y \in C$. □

Example 2.8. *(Convex polytopes). Consequently, a set defined by linear inequalities, i.e.*

$$P = \{x \in \mathbb{R}^n | \quad Ax \leq b\},$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ is convex. The convexity of P follows from the fact that it is an intersection of half-spaces:

$$P = \bigcap_{i=1}^{m} \{x \in \mathbb{R}^n | \quad a_i x \leq b_i\},$$

where a_i is the i-th row of A.

Convexity is stable under Cartesian product, just as it is under intersection.

Proposition 2.9. *For $i = 1, ..., k$, let $C_i \subseteq \mathbb{R}^{n_i}$ be convex sets. Then $C_1 \times \cdots \times C_k$ is a convex set of $\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k}$.*

Note that the converse is also true; $C_1 \times \cdots \times C_k$ is convex if and only if each C_i is convex.

We recall that $A : \mathbb{R}^n \to \mathbb{R}^m$ is called affine if

$$A(\lambda x + (1-\lambda)y) = \lambda A(x) + (1-\lambda)A(y)$$

for all x and y in $\mathbb{R}^n$ and all $\lambda \in \mathbb{R}$.

Proposition 2.10. *Let $A : \mathbb{R}^n \to \mathbb{R}^m$ be an affine mapping and C a convex set of $\mathbb{R}^n$. The image $A(C)$ of C under A is convex in $\mathbb{R}^m$. If D is a convex set in $\mathbb{R}^m$, the inverse image*

$$A^{-1}(D) := \{x \in \mathbb{R}^n | \quad A(x) \in D\}$$

is convex in $\mathbb{R}^n$.

Proof. First, we will prove that $A(C)$ is convex. Indeed, let $X = A(x)$, $Y = A(y)$ where $x, y \in C$ and $0 < \lambda < 1$. Then

$$\lambda X + (1-\lambda)Y = \lambda A(x) + (1-\lambda)A(y) = A(\lambda x + (1-\lambda)y) \in A(C).$$

Now, let D be a convex set in in $\mathbb{R}^m$. If $x, y \in A^{-1}(D)$ and $0 < \lambda < 1$, then $A(x), A(y) \in D$. Therefore,

$$A(\lambda x + (1-\lambda)y) = \lambda A(x) + (1-\lambda)A(y) \in D.$$

It means that $\lambda x + (1-\lambda)y \in A^{-1}(D)$. □

Proposition 2.11. *For $i = 1, ..., k$, let $C_i \subseteq \mathbb{R}^n$ be convex sets. and let $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$. Then the set*

$$\lambda_1 C_1 + \cdots + \lambda_k C_k = \left\{ \sum_{i=1}^{k} \lambda_i x_i | \quad x_i \in C_i, i = 1, 2, \ldots, k \right\}$$

is convex.

Proof. Let $X = \sum_{i=1}^{k} \lambda_i x_i$ and $Y = \sum_{i=1}^{k} \lambda_i y_i$ where $x_i, y_i \in C_i, i = 1, 2, \ldots, k$. If $0 < a < 1$, we have

$$aX + (1-a)Y = \sum_{i=1}^{k} \lambda_i (a x_i + (1-a) y_i) \in \lambda_1 C_1 + \cdots + \lambda_k C_k.$$

□

Proposition 2.12. *If C is convex, so are its interior $intC$ and its closure clC.*

Proof. First, we will prove that $intC$ is convex. Let $x, y \in intC$ and$0 < \lambda < 1$. There exists $\epsilon_1, \epsilon_2 > 0$ such that

$$B(x, \epsilon_1) \subset C,$$

and

$$B(y, \epsilon_2) \subset C.$$

Set $\epsilon = \min(\epsilon_1, \epsilon_2)$. Take an arbitrary point $z \in B(\lambda x+(1-\lambda)y, \epsilon)$, let $d = z-(\lambda x+(1-\lambda)y)$. We have $\|d\| \leq \epsilon$, hence $x + d \in B(x, \epsilon_1)$ and $y + d \in B(y, \epsilon_2)$; So $x + d, y + d \in C$ and

$$z = \lambda(x + d) + (1 - \lambda)(y + d) \in C.$$

It means that $B(\lambda x + (1 - \lambda)y, \epsilon) \subset C$. We can conclude that $\lambda x + (1 - \lambda)y \in intC$. □

Convex Combinations and Convex Hulls

Definition 2.13. *A convex combination of elements $x_1, \ldots, x_k$ in $\mathbb{R}^n$ is an element of the form*

$$\sum_{i=1}^{k} \lambda_i x_i$$

where $\sum_{i=1}^{k} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i = 1, \ldots, k$.

Proposition 2.14. *A set $C \subset \mathbb{R}^n$ is convex if and only if it contains every convex combination of its elements.*

Proof. ($\Leftarrow$) Assume that C contains every convex combinations of its elements. Clearly, for any $x, y \in C$ and $0 < \lambda < 1$, $\lambda x + (1-\lambda)y \in C$.

($\Rightarrow$) Assume that C is convex. We prove this by induction on k. Clearly, for $k = 2$, for two arbitrary elements x, y of C, any convex combination of x and y belongs to C. If for k arbitray elements $x_1, \ldots, x_k$ and $\lambda_1, \ldots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i = 1, \ldots, k$, we have

$$\sum_{i=1}^{k} \lambda_i x_i \in C.$$

Now for $k+1$ arbitrary elements $x_1, \ldots, x_k, x_{k+1}$ and $\lambda_1, \ldots, \lambda_k, \lambda_{k+1}$ such that $\sum_{i=1}^{k+1} \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i = 1, \ldots, k+1$, we have

$$\sum_{i=1}^{k+1} \lambda_i x_i = \sum_{i=1}^{k} \lambda_i x_i + \lambda_{k+1} x_{k+1} = (1 - \lambda_{k+1}) z + \lambda_{k+1} x_{k+1},$$

where $z = \sum_{i=1}^{k} \frac{\lambda_i}{\sum_{i=1}^{k} \lambda_i} x_i \in C$. Therefore $\sum_{i=1}^{k+1} \lambda_i x_i \in C$.

□

Definition 2.15. *(Convex hulls)*

Let $C \subset \mathbb{R}^n$. The convex hull of C, denoted by $\mathrm{conv}(C)$*, is the set comprising all the convex combinations of vectors from C:*

$$\mathrm{conv}(C) := \left\{ \sum_{i=1}^{k} \lambda_i x_i | \quad x_i \in C, \lambda_i \geq 0 \quad \forall i = 1, \ldots, k, \sum_{i=1}^{k} \lambda_i = 1, k \in \mathbb{N} \right\}.$$

Note that in the definition of the convex hull, the number of vectors k in the convex combination representation can be any positive integer. The convex hull $\mathrm{conv}(C)$ is the "smallest" convex set containing C meaning that if another convex set T contains C, then $\mathrm{conv}(C) \subset T$. This property is stated and proved in the following lemma.

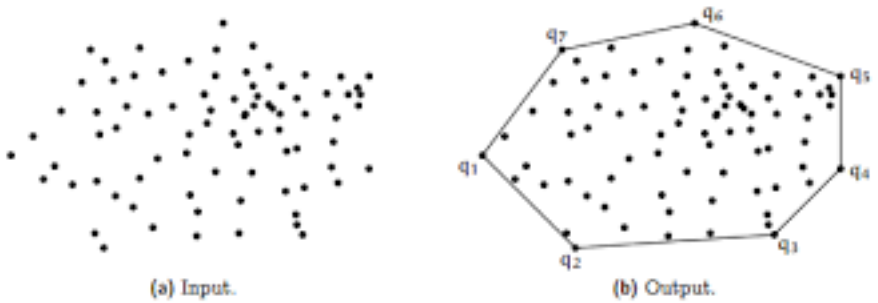


Figure 2.3: Convex hull

(Source: https://medium.com/@harshitsikchi/convex-hulls-explained-baab662c4e94)

Lemma 2.16. *Let $C \subset \mathbb{R}^n$. If $C \subset T$ for some convex set T , then* $\text{conv}(C) \subset T$.

An example of a convex hull of a nonconvex set is given in Figure 2.3.

Theorem 2.17. *(Carathéodory theorem)*

Let $C \subset \mathbb{R}^n$ and let $x \in \text{conv}(C)$. Then there exist $x_1, \ldots, x_{n+1} \in C$ such that $x \in \text{conv}(\{x_1, \ldots, x_{n+1}\})$; that is, there exist $\lambda \in \triangle_{n+1}$ such that

$$x = \sum_{i=1}^{n+1} \lambda_i x_i.$$

Proof. Let x be a convex combination of m points $x_1, \ldots, x_m$ in C with $m > n + 1$, that is

$$x = \sum_{i=1}^{m} \lambda_i x_i,$$

where $\lambda_i \geq 0$, $\sum_{i=1}^{m} \lambda_i = 1$. We will show that m can be reduced by one. It is easy to see that if $\lambda_i = 0$ for some i. So, we assume that $\lambda_i > 0$ for all i. Since $m > n + 1$, there exists

$\gamma_1, \ldots, \gamma_m$ not all equal 0 such that

$$\gamma_1 \begin{bmatrix} x_1 \\ \\ 1 \end{bmatrix} + \cdots + \gamma_m \begin{bmatrix} x_m \\ \\ 1 \end{bmatrix} = 0. \tag{2.1}$$

Hence, $\gamma_1 + \cdots + \gamma_m = 0$ and they not all equal 0. Therefore, there exist i such that $\gamma_i > 0$. Let $\tau = \min\{\frac{\lambda_i}{\gamma_i} : \gamma_i > 0\}$ and $\overline{\lambda_i} = \lambda_i - \tau\gamma_i$ for any i. From (2.1), it implies that

$$\sum_{i=1}^{m} \overline{\lambda}_i = 1,$$

and

$$x = \sum_{i=1}^{m} \overline{\lambda}_i x_i.$$

In addition, we have $\overline{\lambda}_i \geq 0$ and by the definition of τ, there exists i_0 such that $\overline{\lambda}_{i_0} = 0$. So we can delete x_{i_0} in the convex combination. □

Projection onto closed convex sets

Let C be a nonempty closed convex set of $\mathbb{R}^n$. For fixed $x \in \mathbb{R}^n$, we consider the following problem:

$$\inf_{y \in C} \|x - y\|^2, \tag{2.2}$$

i.e. we are interested in those points (if any) of C that are closest to x for the Euclidean distance.

Proposition 2.18. *The minimization problem (2.2) has a unique solution.*

Proof. Set $\inf_{y \in C} \|x - y\|^2 = \mu$. Let $\{y_k\}$ be a sequence in C such that

$$\lim_{k \to \infty} \|x - y_k\| = \mu.$$

There exists k_0 such that for any $k \geq k_0$,

$$\|x - y_k\| < \mu + 1,$$

or equivalently

$$y_k \in B(x, \mu + 1).$$

Hence, the sequence $\{y_k\}$ is bounded. So there exists a subsequence $\{y_{k_j}\}$ of $\{y_k\}$ that is convergent. Let us denote by z the limit of $\{y_{k_j}\}$. We have $z \in C$ (because C is closed) and

$$\|x - z\| = \lim_{j \to \infty} \|x - y_{k_j}\| = \mu.$$

This proves that the problem (2.2) has at least one solution.

Now, assume that z_1 and z_2 are two different solutions of (2.2), that means $z_1, z_2 \in C$ and

$$\|x - z_1\| = \|x - z_2\| = \mu.$$

Let $z = \frac{1}{2}(z_1 + z_2)$. Since C is convex, $z \in C$. And

$$\|x - z\|^2 = \|x - \frac{1}{2}(z_1 + z_2)\|^2 = \frac{1}{2}\|x - z_1\|^2 + \frac{1}{2}\|x - z_2\|^2 - \frac{1}{4}\|z_1 - z_2\|^2 < \mu^2.$$

This contradicts with the definition of μ. □

Definition 2.19. *The projection of x on a nonempty closed convex set C in $\mathbb{R}^n$ is the closest point to x in C:*

$$P_C(x) := \text{argmin}_{y \in C} \|x - y\|^2. \tag{2.3}$$

Example 2.20. • *Let us consider the projection of $x \in \mathbb{R}^n$ onto the standard Euclidean ball defined as $B := \{x \in \mathbb{R}^n | \quad \|x\| \leq 1\}$. We can easily see that*

$$P_B(x) = \frac{x}{\|x\|}.$$

- *The projection of $x \in \mathbb{R}^n$ to a hyperplane $H := \{x \in \mathbb{R}^n | \quad w^T x + b = 0\}$ is given by*

$$P_H(x) = x - \frac{(w^T x + b)w}{\|w\|^2}.$$

Theorem 2.21. *Let $x \in \mathbb{R}^n$. Then $z = P_C(x)$ if and only if $z \in C$ and*

$$\langle x - z, y - z \rangle \leq 0 \quad \forall y \in C. \tag{2.4}$$

Proof. ($\Rightarrow$) Let y be an arbitrary point in C. For $0 \leq \lambda \leq 1$, we have

$$\|x - [\alpha y + (1 - \alpha)z]\|^2 = \|x - z\|^2 - 2\lambda\langle x - z, y - z\rangle + \lambda^2\|y - z\|^2.$$

Since $z = P_C(x)$,

$$\|x - [\alpha y + (1 - \alpha)z]\|^2 \geq \|x - z\|^2.$$

Equivalently, for $0 \leq \lambda \leq 1$

$$-2\lambda\langle x - z, y - z\rangle + \lambda^2\|y - z\|^2 \geq 0.$$

This implies

$$\langle x - z, y - z \rangle \leq 0.$$

($\Leftarrow$) Suppose that (2.4) is satisfied for $z \in C$. By choosing $y = P_C(x)$, we obtain

$$\langle x - z, P_C(x) - z \rangle \leq 0. \tag{2.5}$$

We also have

$$\langle x - P_C(x), y - P_C(x) \rangle \leq 0 \quad \forall y \in C.$$

By taking $y = z$, we obtain

$$\langle x - P_C(x), z - P_C(x)\rangle \leq 0. \tag{2.6}$$

Combining (2.5) and (2.6), we get

$$\|z - P_C(x)\|^2 \leq 0,$$

which means $z = P_C(x)$.

□

Theorem 2.22. *For any $x, y \in \mathbb{R}^n$, we have*

(i) $\langle P_C(x) - P_C(y), x - y\rangle \geq \|P_C(x) - P_C(y)\|^2$,

(ii) $\|P_C(x) - P_C(y)\| \leq \|x - y\|$.

Proof. (i) By Theorem 2.21,

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

and

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

Adding the above inequalities, we get

$$\langle P_C(x) - P_C(y), x - y\rangle \geq \|P_C(x) - P_C(y)\|^2.$$

(ii) Obviously,

$$\begin{aligned} & \|P_C(x) - P_C(y) + y - x\|^2 \geq 0 \\ \Leftrightarrow \quad & \|P_C(x) - P_C(y)\|^2 - 2\langle P_C(x) - P_C(y), x - y\rangle + \|x - y\|^2 \geq 0. \end{aligned}$$

Combining this with part (i), we obtain

$$\|P_C(x) - P_C(y)\|^2 \leq \|x - y\|^2.$$

□

Seperation theorems

Theorem 2.23. *Let $C \subset \mathbb{R}^n$ be nonempty closed convex, and let $x \notin C$. Then there exists $s \in \mathbb{R}^n$ such that*

$$\langle s, x\rangle > \sup_{y\in C}\langle s, y\rangle.$$

Proof. Let $z = P_C(x)$. By Theorem 2.21, for any $y \in C$,

$$\langle x - z, y - z\rangle \leq 0.$$

Set $s = x - z$. Note that $z \neq 0$ because $x \notin C$. So, for $y \in C$,

$$\langle s, y\rangle \leq \langle s, z\rangle = \langle s, x\rangle - \|s\|^2.$$

Therefore,

$$\langle s, x\rangle > \sup_{y\in C}\langle s, y\rangle.$$

□

Corollary 2.24. *(Strict Separation of Convex Sets)*

Let C_1, C_2 be two nonempty closed convex sets with $C_1 \cap C_2 = \emptyset$. If C_2 is bounded, there exists $s \in \mathbb{R}^n$ such that

$$\sup_{y \in C_1} \langle s, y \rangle < \min_{y \in C_2} \langle s, y \rangle.$$

Proof. The set $C = C_1 - C_2$ is closed and $0 \notin C$. Thanks to Theorem 2.23, we can prove that there exists $s \in \mathbb{R}^n$ such that

$$\sup_{y \in C_1} \langle s, y \rangle < \min_{y \in C_2} \langle s, y \rangle.$$

□

Theorem 2.25. *(Proper Separation of Convex Sets)*

If the two nonempty convex sets C_1 and C_2 satisfy $riC_1 \cap riC_2 = \emptyset$, they can be properly separated.

Proof. Proof of this theorem can be found in [1, 2]. □

2 Convex Functions

Definitions and Examples

Definition 2.26. *Let C be a nonempty convex set in $\mathbb{R}^n$. A function $f : C \to \mathbb{R} \cup \{+\infty\}$ is said to be convex when, for all $x, y \in C$ and all $\lambda \in [0, 1]$, there holds*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \tag{2.7}$$

We say that f is **strictly convex** on C if (2.7) holds as a strict inequality if $x \neq y$ and $0 < \lambda < 1$. The function f is said to be **strongly convex** on C if there exists $c > 0$ such that

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{1}{2}c\lambda(1 - \lambda)\|x - y\|^2$$

for all $x, y \in C$ and all $\lambda \in [0, 1]$.

Example 2.27. *1. The affine function $f(x) = a^T x + b$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$ is convex.*

2. The norm function $f(x) = \|x\|$ is convex.

Proposition 2.28. *The function f is strongly convex with modulus c if and only if the function $f - \frac{c}{2}\|.\|^2$ is convex .*

In a more modern definition, a convex function f is considered as defined on the whole of $\mathbb{R}^n$, but possibly taking infinite values:

Definition 2.29. *A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to be convex when, for all $x, y \in \mathbb{R}^n$ and all $\lambda \in [0, 1]$, there holds*

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

The class of such functions is denoted by Conv$\mathbb{R}^n$ .To realize the equivalence between our two definitions, extend an f from Definition 2.26 by

$$f(x) := +\infty \quad \forall x \notin C.$$

thus obtaining a new f, which is now in Conv$\mathbb{R}^n$.

Definition 2.30. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The domain (or effective domain) of f is the nonempty set*

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

Definition 2.31. *Given $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, not identically equal to $+\infty$, the epigraph of f is the nonempty set*

$$\text{epi} f := \{(x, r) \in \mathbb{R}^n \times \mathbb{R}| \quad r \geq f(x)\}.$$

Proposition 2.32. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be not identically equal to $+\infty$,. The following properties are equivalent:*

(i) f is convex;

(ii) its epigraph is a convex set in $\mathbb{R}^n \times \mathbb{R}$;

Proof. ($\Rightarrow$) Assume that f is convex. Take (x_1, r_1), (x_2, r_2) in epi$f$, we have $f(x_1) \leq r_1$ and $f(x_2) \leq r_2$. For $\lambda \in [0, 1]$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \leq \lambda r_1 + (1-\lambda) r_2.$$

Therefore, $(\lambda x_1 + (1-\lambda)x_2, \lambda r_1 + (1-\lambda) r_2)$ belongs to epi$f$. So, epi$f$ is convex.

($\Leftarrow$) Now assume that epi$f$ is convex. Take x_1, x_2 arbitrary in $\mathbb{R}^n$. It is clear that $(x_1, f(x_1))$ and $(x_2, f(x_2)$ are in epi$f$. So, for $\lambda \in [0, 1]$,

$$\lambda(x_1, f(x_1)) + (1-\lambda)(x_2, f(x_2)) \in \text{epi} f.$$

It means that

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2).$$

□

The basic property characterizing a convex function is that the function value of a convex combination of two points x and y is smaller than or equal to the corresponding convex combination of the function values $f(x)$ and $f(y)$. An interesting result is that convexity implies that this property can be generalized to convex combinations of any number of vectors. This is the so-called Jensen's inequality.

Theorem 2.33. *Let $f : C \to \mathbb{R}$ be a convex function where $C \subseteq \mathbb{R}^n$ is a convex set. Then for any $x_1, \ldots, x_k \in C$ and $\lambda \in \triangle_k$, the following inequality holds:*

$$f(\sum_{i=1}^{k} \lambda_i x_i) \leq \sum_{i=1}^{k} \lambda_i f(x_i). \tag{2.8}$$

Proof. We prove this theorem by induction in k. It is true for $k = 2$. Assume that (2.8) is true for k. Then for any $x_1, \ldots, x_{k+1} \in C$ and $\lambda \in \triangle_{k+1}$, by setting

$$y = \sum_{i=1}^{k} \frac{\lambda_i}{\sum_{i=1}^{k} \lambda_i} x_i,$$

we obtain

$$\sum_{i=1}^{k+1} \lambda_i x_i = (1 - \lambda_{k+1}) y + \lambda_{k+1} x_{k+1}.$$

Since f is convex,

$$f(\sum_{i=1}^{k+1} \lambda_i x_i) \leq (1 - \lambda_{k+1}) f(y) + \lambda_{k+1} f(x_{k+1}). \tag{2.9}$$

Moreover, let $\lambda_i' = \frac{\lambda_i}{\sum_{i=1}^{k} \lambda_i}$, then $\lambda_i' \geq 0$ and

$$\sum_{i=1}^{k} \lambda_i' = 1.$$

In the other words, $\lambda' = (\lambda_1', \ldots, \lambda_k') \in \triangle_k$. Note that (2.8) is true for k so

$$f(y) \leq \sum_{i=1}^{k} \lambda_i' f(x_i). \tag{2.10}$$

From (2.9) and (2.10),

$$f(\sum_{i=1}^{k+1} \lambda_i x_i) \leq \sum_{i=1}^{k+1} \lambda_i f(x_i).$$

□

First Order Characterizations of Convex Functions

Convex functions are not necessarily differentiable, but in case they are, we can replace the Jensen's inequality definition with other characterizations which utilize the gradient

of the function. An important characterizing inequality is the gradient inequality, which essentially states that the tangent hyperplanes of convex functions are always underestimates of the function.

Theorem 2.34. *Let $f : C \to \mathbb{R}$ be a continuously differentiable function defined on a convex set $C \subseteq \mathbb{R}^n$. Then f is convex over C if and only if*

$$f(x) + \nabla f(x)^T (y - x) \leq f(y) \quad \forall x, y \in C. \tag{2.11}$$

Proof. ($\Rightarrow$) Assume that f is convex. Let x, y be two arbitrary points in C. If $x = y$, (2.11) is satisfied. If $x \neq y$, then for $\lambda \in [0, 1]$,

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

It is equivalent with

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

Let $\lambda \to 0^+$, it becomes

$$\nabla f(x)^T (y - x) \leq f(y) - f(x).$$

($\Leftarrow$) Now assume that (2.11) is true for all $x, y \in C$. Let $\lambda \in [0, 1]$, we set $u = \lambda x + (1 - \lambda)y$. From (2.11),

$$f(u) + \nabla f(u)^T (x - u) \leq f(x),$$

and

$$f(u) + \nabla f(u)^T (y - u) \leq f(y).$$

Multiplying these inequalities by λ and $(1-\lambda)$, we get

$$f(u) \leq \lambda f(y) + (1-\lambda) f(x).$$

□

Theorem 2.35. *Let $f : C \to \mathbb{R}$ be a continuously differentiable function defined on a convex set $C \subseteq \mathbb{R}^n$. Then f is strictly convex over C if and only if*

$$f(x) + \nabla f(x)^T (y - x) < f(y) \quad \forall x \neq y \in C. \tag{2.12}$$

Theorem 2.36. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be the quadratic function given by $f(x) = x^T Ax + 2b^T x + c$, where $A \in \mathbb{R}^{n \times n}$ is symmetric, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$. Then f is (strictly) convex if and only if $A \succeq 0 (A \succ 0)$.*

Proof. For $f = x^T Ax + 2b^T x + c$ with A is symmetric, we have

$$\nabla f(x) = 2Ax + 2b.$$

By Theorem 2.34, f is convex if and only if

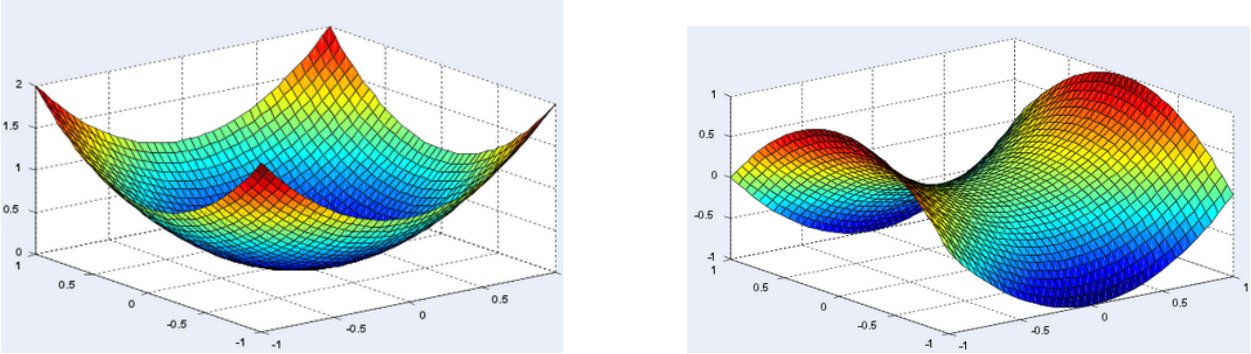
$$x^T Ax + 2b^T x + c + 2(Ax + b)^T (y - x) \leq y^T Ay + 2b^T y + c \quad \forall x, y \in \mathbb{R}^n.$$

This equivalents to

$$(y - x)^T A (y - x) \geq 0 \forall x, y \in \mathbb{R}^n,$$

which means $A \succeq 0$.

By the same arguments, we can proof the second part of this theorem. □



(a) Convex quadratic function

(b) Noncovex quadratic function

Figure 2.4: Source: https://www.quora.com/Are-all-quadratic-programming-problems-convex

Examples of convex and nonconvex quadratic functions are illustrated in Figure 2.4.

The next theorem is about the monotonicity of the gradient of a convex function.

Theorem 2.37. *Suppose that f is a continuously differentiable function over a convex set $C \subseteq \mathbb{R}^n$. Then f is convex over C if and only if*

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0 \quad \forall x, y \in C. \tag{2.13}$$

Second Order Characterization of Convex Functions

Theorem 2.38. *(Second order characterization of convexity)*

Let f be a twice continuously differentiable function over an open convex set $C \subseteq \mathbb{R}^n$. Then f is convex if and only if $\nabla^2 f(x) \succeq 0$ for any $x \in C$.

Theorem 2.39. *(Sufficient second order condition for strict convexity)*

Let f be a twice continuously differentiable function over a convex set $C \subseteq \mathbb{R}^n$, and suppose that $\nabla^2 f(x) \succ 0$ for any $x \in C$. Then f is strictly convex over C.

Example 2.40.

- *The log-sum-exp function $f(x) = \ln(e^{x_1} + e^{x_2} + \cdots + e^{x_n})$ is convex on $\mathbb{R}^n$.*
- *The quadratic-over-linear function $f(x) = \frac{x_1^2}{x_2}$ is convex on $\{(x_1, x_2)| \quad x_2 > 0\}$.*

Operations Preserving Convexity

Theorem 2.41. *(Preservation of convexity under summation and multiplication by nonnegative scalars).*

(i) Let f be a convex function defined over a convex set $C \subseteq \mathbb{R}^n$ and let $\alpha \geq 0$. Then αf is a convex function over C.

(ii) Let $f_1, f_2, \ldots, f_p$ be convex functions over a convex set $C \subseteq \mathbb{R}^n$. Then the sum function $f_1 + f_2 + \ldots + f_p$ is convex over C.

Theorem 2.42. *(Preservation of convexity under linear change of variables) Let $f : C \to \mathbb{R}$ be a convex function defined on a convex set $C \subseteq \mathbb{R}^n$. Let $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$. Then the function g defined by*

$$g(y) = f(Ay + b)$$

is convex over the convex set $D = \{y \in \mathbb{R}^m| \quad Ay + b \in C\}$.

Example 2.43. *(Generalized quadratic-over-linear)*

Let $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^m$,$c \in \mathbb{R}^n$, and $d \in \mathbb{R}$. We assume that $c \neq 0$. We will show that the quadratic-over-linear function

$$g(x) = \frac{\|Ax+b\|^2}{c^T x + d}$$

is convex over $D = \{x \in \mathbb{R}^n : | \quad c^T x + d > 0\}$.

Theorem 2.44. *(Preservation of convexity under composition with a nondecreasing convex function)*

Let $f : C \to \mathbb{R}$ be a convex function over the convex set $C \subseteq \mathbb{R}^n$. Let $g : I \to \mathbb{R}$ be a one-dimensional nondecreasing convex function over the interval $I \subseteq \mathbb{R}$. Assume that the image of C under f is contained in $I : f(C) \subseteq I$. Then the composition of g with f defined by

$$h(x) := g(f(x)),$$

is a convex function over C.

Example 2.45.

- *The function $h(x) = e^{\|x\|^2}$ is convex since it can be represented as $h(x) = g(f(x))$, where $g(t) = e^t$ is a nondecreasing convex function and $f(x) = \|x\|^2$ is a convex function.*

- *The function $h(x) = (\|x\|^2+1)^2$ is a convex function over $\mathbb{R}^n$ since it can be represented as $h(x) = g(f(x))$, where $g(t) = t^2$ and $f(x) = \|x\|^2 + 1$.*

Another important operation that preserves convexity is the pointwise maximum of convex functions.

Theorem 2.46. *(Pointwise maximum of convex functions)*

Let $f_1, \ldots, f_p : C \to \mathbb{R}$ be p convex functions over the convex set $C \subseteq \mathbb{R}^n$. Then the maximum function

$$f(x) := \max_{i=1,\ldots,p} f_i(x)$$

is a convex function over C.

Example 2.47. *(i) $f(x) = \max\{x_1, \ldots, x_n\}$ is convex.*

(ii) Given a vector $x = (x_1, \ldots, x_n)^T$. Let $x\,[i]$ denote the ith largest value in x. The sum of the k largest components

$$h_k(x) = x\,[1] + \cdots + x\,[k]\,,$$

is convex.

Theorem 2.48. *Let $f : C \times D \to \mathbb{R}$ be a convex function defined over the set $C \times D$ where $C \subseteq \mathbb{R}^m$ and $D \subseteq \mathbb{R}^n$ are convex sets. Let*

$$g(x) = \min_{y \in D} f(x, y) \quad \forall x \in C,$$

where we assume that the minimum in the above definition is finite. Then g is convex over C.

Example 2.49. *Let $C \subseteq \mathbb{R}^n$ be a convex set. The distance function defined by*

$$d(x, C) = \min\{\|x - y\| : y \in C\}$$

is convex.

Definition 2.50. *(Level sets). Let $f : C \to \mathbb{R}$ be a convex function over the convex set $C \subseteq \mathbb{R}^n$.. Then the level set of f with level α is given by*

$$L_\alpha f = \{x \in C| \quad f(x) \leq \alpha\}.$$

A fundamental property of convex functions is that their level sets are necessarily convex.

Theorem 2.51. *(Convexity of level sets of convex functions). Let $f : C \to \mathbb{R}$ be a convex function over the convex set $C \subseteq \mathbb{R}^n$. Then for any $\alpha \in \mathbb{R}$ the level set $L_\alpha f$ is convex.*

Continuity and Differentiability of Convex Functions

Theorem 2.52. *(local Lipschitz continuity of convex functions). Let $f : C \to \mathbb{R}$ be a convex function over the convex set $C \subseteq \mathbb{R}^n$. Let $x_0 \in int(C)$. Then there exist $\epsilon > 0$ and $L > 0$ such that $B(x_0, \epsilon) \subseteq C$ and*

$$|f(x) - f(x_0)| \leq L\|x - x_0\|$$

for all $x \in B(x_0, \epsilon)$.

Theorem 2.53. *(existence of directional derivatives for convex functions). Let $f : C \to \mathbb{R}$ be a convex function over the convex set $C \subseteq \mathbb{R}^n$. Let $x \in int(C)$. Then for any $d \neq 0$, the directional derivative $f'(x; d)$ exists.*

Exercises

1. For each of the following sets determine whether they are convex or not.

(i) $C_1 = \{x \in \mathbb{R}^n | \quad \|x\|^2 = 1\}$.

(ii) $C_2 = \{x \in \mathbb{R}^n | \quad \max_{i=1,\ldots,n} x_i \leq 1\}$.

(iii) $C_3 = \{x \in \mathbb{R}^n | \quad \min_{i=1,\ldots,n} x_i \leq 1\}$.

(iv) $C_4 = \{x \in \mathbb{R}^n_{++} | \quad \prod_{i=1}^n x_i \geq 1\}$.

2. Prove that

(i) The function $f(x_1, x_2) = x_1^2 + 2x_1x_2 + 3x_2^2 + 2x_1 - 3x_2 + e^{x_1}$ is convex on $\mathbb{R}^2$.

(ii) The function $f(x_1, x_2, x_3) = e^{x_1} - x_2 + x_3 + e^{2x_2} + x_1$ is convex on $\mathbb{R}^3$.

(iii) The function $f(x_1, x_2) = -\log(x_1x_2)$ is convex on $\mathbb{R}^2_{++}$.

3. Show that the log-sum-exp function $f(x) = \log(\sum_{i=1}^n e^{x_i})$ is not strictly convex over $\mathbb{R}^n$.

4. Show that the following functions are convex over the specified domain C:

(i) $f(x_1, x_2, x_3) = -\sqrt{x_1x_2} + 2x_1^2 + 2x_2^2 + 3x_3^2 - 2x_1x_2 - 2x_2x_3$ over $\mathbb{R}^3_{++}$.

(ii) $f(x) = \|x\|^4$ over $\mathbb{R}^n$.

(iii) $f(x) = \sum_{i=1}^n x_i \log(x_i) - (\sum_{i=1}^n x_i) \log(\sum_i^n x_i)$ over $\mathbb{R}^n_{++}$.

(iv) $f(x) = x^TQx + 1$ over $\mathbb{R}^n$, where $Q \succ 0$ is an $n \times n$ matrix.

(v) $f(x_1, x_2) = (2x_1^2 + 3x_2^2)(\frac{1}{2}x_1^2 + \frac{1}{3}x_2^2$.

5. Let $A \in \mathbb{R}^{m\times n}$, and let $f : \mathbb{R}^n \to \mathbb{R}$ be defined by

$$f(x) = \log \sum_{i=1}^m e^{A_i x},$$

where A_i is the ith row of A. Prove that f is convex over $\mathbb{R}^n$.

6. Show that the function $f(x_1, x_2, x_3) = -e^{(-x_1+x_2-2x_3)^2}$ is not convex over $\mathbb{R}^n$.

7. (i) Show that the function $f(x) = \sqrt{1 + \|x\|^2}$ is strictly convex over $\mathbb{R}^n$ but is not strongly convex over $\mathbb{R}^n$.

   (ii) Show that the quadratic function $f(x) = x^T Ax + 2b^T x + c$ with $A = A^T \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$ is strongly convex if and only if $A \succ 0$, and in that case the strong convexity parameter is $2\lambda_{\min}(A)$.

8. Prove that for any $x_1, \ldots, x_n \geq 0$, the following inequality holds:

$$\frac{1}{n}\sum_{i=1}^{n} x_i \geq \sqrt[n]{\prod_{i=1}^{n} x_i}.$$

   More generally, for any $\lambda \in \triangle_n$ one has

$$\sum_{i=1}^{n} \lambda_i x_i \geq \prod_{i=1}^{n} x_i^{\lambda_i}.$$

9. For any $s, t \geq 0$ and $p, q > 1$ satisfying $\frac{1}{p} + \frac{1}{q} = 1$, prove that

$$st \leq \frac{s^p}{p} + \frac{t^q}{q}.$$

10. (Holder's inequality). For any $x, y \in \mathbb{R}^n$ and $p, q \geq 1$ satisfying $\frac{1}{p} + \frac{1}{q} = 1$, prove that

$$|x^T y| \leq \|x\|_p \|y\|_q.$$

11. (Minkowski's inequality). Let $p \geq 1$. Then for any $x, y \in \mathbb{R}^n$, prove that

$$\|x + y\|_p \leq \|x\|_p + \|y\|_p.$$

3

Unconstrained Optimization

In this chapter, we study the unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function.

This problem is intensively studied in many works ([1, 4, 5] and the references therein). In the first section of this chapter, we present the optimality conditions of (3.1). An important class of unconstrained problem, the least square problems, is studied in Section 3.2. Then in the last sections, the gradient, Newton and quasi-Newton methods are introduced.

Some examples of unconstrained optimization problem:

Example 3.1. *(Linear least-square problem) Suppose that we are given a linear system of the form*

$$Ax = b,$$

where $A \in \mathbb{R}^{m\times n}$ *and* $b \in \mathbb{R}^m$*. Assume that* $m > n$ *and* A *has a full column rank; that is,* $\text{rank}(A) = n$*. Under these assumptions, the system is usually inconsistent (has no solution) and a common approach for finding an approximate solution is to take the optimal solution of the following minimization problem*

$$(LS) \quad \min_{x\in\mathbb{R}^n} \|Ax - b\|^2.$$

Problem (LS) is a problem of minimizing a quadratic function over the entire space.

Example 3.2. *(Non-linear least square problem) Suppose that we are trying to find a curve of form* $y = f(x, \beta)$ *with* $\beta \in \mathbb{R}^n$ *that fits some experimental data:* $(x_1, y_1), \ldots, (x_m, y_m)$*. In order to find the vector of parameters* β *such that the curve fits best the given data in the least squares sense, we solve the unconstrained minimization problem*

$$(NLS) \quad \min_{\beta\in\mathbb{R}^n} \sum_{i=1}^{m} (f(x_i, \beta) - y_i)^2.$$

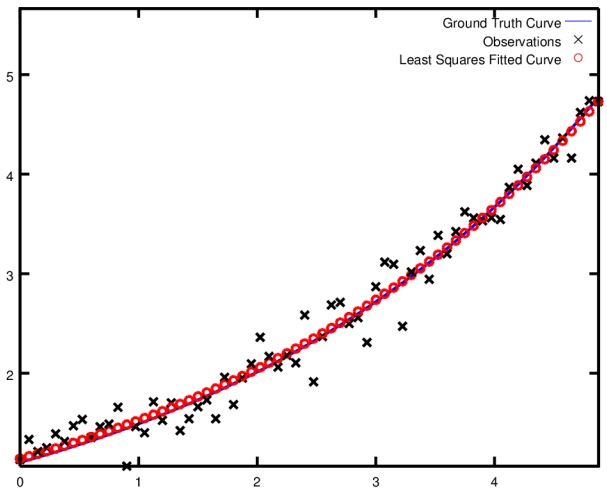


Figure 3.1: Nonlinear least square

(Source: http://ceres-solver.org/nnls_tutorial.html)

This is a nonlinear least-squares problem, a special case of unconstrained optimization. It illustrates that some objective functions can be expensive to evaluate even when the number of variables is small. If the number of given datas is large then the evaluation of $f(x, \beta)$ for a given parameter vector x is a significant computation.

1 Optimality Conditions

Definition 3.3. *(local and global minimum).*

Let $f : C \to \mathbb{R}$ be defined on a set $C \subseteq \mathbb{R}^n$. We consider the problem

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

Then

1. $x^* \in C$ *is called a* ***local minimizer*** *(maximizer) of this problem if there exists* $\epsilon > 0$ *such that* $f(x) \geq f(x^*)$ $(f(x) \leq f(x^*))$ *for any* $x \in C$ *such that* $\|x - x^*\| \leq \epsilon$.

2. $x^* \in C$ *is called a* ***strict local minimizer*** *(maximizer) of this problem if there exists* $\epsilon > 0$ *such that* $f(x) > f(x^*)$ $(f(x) < f(x^*))$ *for any* $x \neq x^* \in C$ *such that* $\|x - x^*\| \leq \epsilon$.

3. $x^* \in C$ *is called a* ***global minimizer*** *(maximizer) of this problem if* $f(x) \geq f(x^*)$ $(f(x) \leq f(x^*))$ *for any* $x \in C$.

4. $x^* \in C$ *is called a* ***strict global minimizer*** *(maximizer) of this problem if* $f(x) > f(x^*)$ $(f(x) < f(x^*))$ *for any* $x \neq x^* \in C$.

The first question is about the existence: whether a function actually has a global minimizer or maximizer on a set or not. Answer to this question, the Weierstrass theorem state that a continuous function attains its minimum and maximum over a compact set.

Theorem 3.4. *(Weierstrass theorem) Let* f *be a continuous function defined on a nonempty and compact set* $C \subseteq \mathbb{R}^n$. *Then there exists a global minimizert of* f *on* C *and a global maximizer of* f *on* C.

When the underlying set is not compact, we can not apply the Weierstrass theorem. Fortunately, some properties of f can guarantee the existence of the solution.

Definition 3.5. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a continuous function defined over* $\mathbb{R}^n$. *The function* f *is called coercive if*

$$\lim_{\|x\| \to \infty} f(x) = \infty.$$

Theorem 3.6. *([1]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous and coercive function and let $C \subseteq \mathbb{R}^n$ be a nonempty closed set. Then f has a global minimum point over C.*

Our main task will usually be to find and study global minimum (or maximum points); however, most of the theoretical results only characterize local minima and maxima which are optimal points with respect to a neighborhood of the point of interest.

Example 3.7.

- *For the constant function $f(x) = 5$, every point x is a global minimizer.*
- *The function $f(x) = (x-3)^4$ has a strict global minimizer at $x = 3$.*

Example 3.8. *Consider the two-dimensional function*

$$f(x, y) = \frac{x + y}{x^2 + y^2 + 1}.$$

Figure 3.2 shows the plot of this function. This function has a global maximizer $(x, y) = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and a global minimizer $(x, y) = (-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$.

Example 3.9. *The function*

$$f(x) = \begin{cases} x^4 cos(1/x) + 2x^4 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

has a global minimizer at $x^* = 0$ and many local minimizers (see Figure 3.3). For the functions that has many local minimizers like this function, it is usually difficult to find the global minimizer , because algorithms can be "trapped" at local minimizers.

The Fermat's theorem states that for a one-dimensional function f defined and differentiable on an interval (a, b), if a point $x^* \in (a, b)$ is a local minimizer or maximizer, then

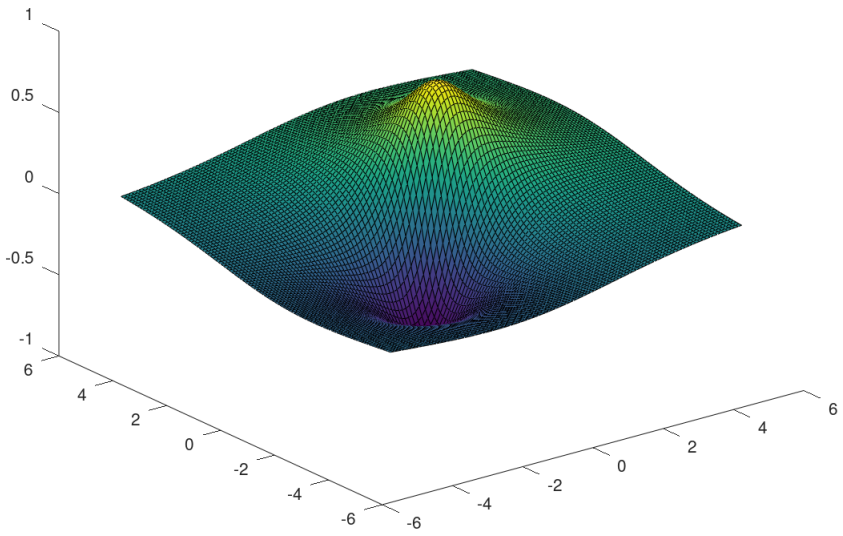


Figure 3.2: Plot of function f in Example 3.8

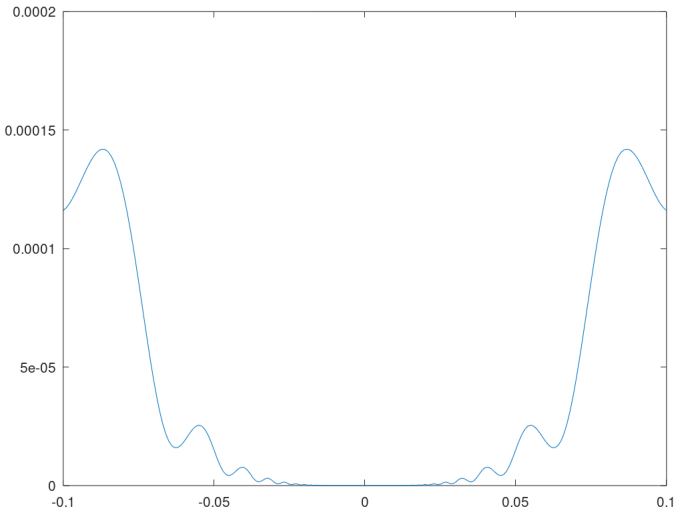


Figure 3.3: Function with many local minimizers

$f'(x^*) = 0$. This result can be extended to the multidimensional case: the gradient is zero at local optimum points. We refer to such an optimality condition as a first order optimality condition, as it is expressed in terms of the first order derivatives. In what follows, we will also discuss second order optimality conditions that use in addition information on the second order (partial) derivatives.

Theorem 3.10. *Assume that $f : C \to \mathbb{R}$ is differentiable at a point $\overline{x} \in int(C)$. If f attains its local minimum at $\overline{x}$ then*

$$\nabla f(\overline{x}) = 0. \tag{3.2}$$

Proof. Since $\overline{x}$ is a local minimizer of f and $\overline{x} \in int(C)$, there exists $\epsilon > 0$ such that for any $y \in B(\overline{x}, \epsilon)$,

$$f(y) \geq f(\overline{x}).$$

For an arbitrary direction $d \in \mathbb{R}^n$, for $t > 0$ small enough, $\overline{x} + td \in B(\overline{x}, \epsilon)$,

$$f(\overline{x} + td) \geq f(\overline{x}),$$

hence

$$\frac{f(\overline{x} + td) - f(\overline{x})}{t} \geq 0.$$

Let $t \to 0^+$, we obtain

$$\langle \nabla f(\overline{x}), d \rangle \geq 0. \tag{3.3}$$

Note that (3.3) is true for any $d \in \mathbb{R}^n$. By applying (3.3) for $-d$,

$$\langle \nabla f(\overline{x}), d \rangle \leq 0.$$

This means

$$\langle \nabla f(\overline{x}), d \rangle = 0 \quad \forall d \in \mathbb{R}^n,$$

or $\nabla f(\overline{x}) = 0$.

□

Definition 3.11. *Assume that $f : C \to \mathbb{R}$ is differentiable at a point $\overline{x} \in int(C)$. Then $\overline{x}$ is called a **stationary point** of f if*

$$\nabla f(\overline{x}) = 0.$$

Theorem 3.12. *([1])*

Let C be a convex set in $\mathbb{R}^n$ and f be a continuously differentiable function which is convex on C. Suppose that $\nabla f(\overline{x}) = 0$ for some $\overline{x} \in C$. Then $\overline{x}$ is a global minimizer of f on C.

When $C = \mathbb{R}^n$, then the condition in Theorem 3.12 is also a necessary condition.

Example 3.13. *Consider the function*

$$f(x) = 3x^4 - 20x^3 + 42x^2 - 36x.$$

We have $f'(x) = 0$ if and only if $x = 1$ or $x = 3$. $x = 3$ is a global minimizer while $x = 1$ is not a local minimizer or maximizer.

Theorem 3.14. *([1])*

Let $f : C \to \mathbb{R}$ be a function defined on an open set $C \subseteq \mathbb{R}^n$. Suppose that f is twice continuously differentiable on C and that $\overline{x}$ is a stationary point. Then the following hold:

(i) If $\overline{x}$ is a local minimum point of f on C, then $\nabla^2 f(\overline{x}) \geq 0$.

(ii) If $\nabla^2 f(\overline{x}) > 0$, then $\overline{x}$ is a strict local minimum point of f on C.

Proof. (i) Since $\overline{x}$ is a stationary point, $\nabla f(\overline{x}) = 0$. If $\overline{x}$ is a local minimum point of f on C, for any $d \in \mathbb{R}^n$ and $t > 0$ small enough, we have

$$0 \leq f(\overline{x} + td) - f(\overline{x}) = \frac{t^2}{2}\langle \nabla^2 f(\overline{x})d, d\rangle + o(t^2),$$

so

$$\langle \nabla^2 f(\overline{x})d, d\rangle \geq 0 \quad \forall d \in \mathbb{R}^n.$$

This means

$$\nabla^2 f(\overline{x}) \succeq 0.$$

(ii) If $\nabla^2 f(\overline{x}) > 0$, there exists $\lambda > 0$ (may be taken as the smallest eigenvalue of $\nabla^2 f(\overline{x})$) such that

$$\langle \nabla^2 f(\overline{x})d, d\rangle \geq \lambda \|d\|^2 \quad \forall d \in \mathbb{R}^n.$$

Since $\nabla f(\overline{x}) = 0$,

$$f(\overline{x} + td) = f(\overline{x}) + \frac{t^2}{2}\langle \nabla^2 f(\overline{x})d, d\rangle + o(t^2)$$

Therefore,

$$\frac{f(\overline{x} + td) - f(\overline{x})}{t^2} \geq \frac{\lambda}{2} + o(1).$$

So, for t small enough, we have $f(x + td) > f(x)$.

□

Definition 3.15. *A stationary point $\overline{x}$ is called a saddle point if it is neither a local minimizer nor a local maximizer.*

Example 3.16. *We consider the function*

$$f(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2 + (x_2^2 - 1)^2.$$

The gradient of f is given by

$$\nabla f(x) = \begin{pmatrix} 4(x_1^2 + x_2^2 - 1)x_1 \\ \\ 4(x_1^2 + x_2^2 - 1)x_2 + 4(x_2^2 - 1)x_2 \end{pmatrix}.$$

To find stationary points, we solve the following equation

$$\begin{aligned} & \nabla f(x) = 0 \\ \Leftrightarrow & \begin{cases} 4(x_1^2 + x_2^2 - 1)x_1 = 0 \\ \\ 4(x_1^2 + x_2^2 - 1)x_2 + 4(x_2^2 - 1)x_2 = 0 \end{cases} \end{aligned}$$

There are 5 stationary points: $(0,0), (1,0), (-1,0), (0,1), (0,-1)$. The Hessian of f is

$$\nabla^2 f(x) = \begin{pmatrix} 4(3x_1^2 + x_2^2 - 1) & 8x_1x_2 \\ \\ 8x_1x_2 & 4(x_1^2 + 6x_2^2 - 1) \end{pmatrix}.$$

We have

$$\nabla^2 f(0,1) = \nabla^2 f(0,-1) = \begin{pmatrix} 0 & 0 \\ \\ 0 & 16 \end{pmatrix} \succeq 0.$$

The Hessian matrices of f at $(0,1)$ and $(0,-1)$ are only positive semidefinite so they may be saddle points. But it is easy to see that the function f is bounded below by 0 and $f(0,1) = f(0,-1) = 0$, so $(0,1)$ and $(0,-1)$ are global minimizers. Since

$$\nabla^2 f(0,0) = \begin{pmatrix} -1 & 0 \\ \\ 0 & -2 \end{pmatrix} \prec 0,$$

then $(0,0)$ *is a local maximizer. And* $\lim_{\|x\|\to\infty} f(x) = \infty$ *so* $(0,0)$ *is not a global maximizer. The points* $(1,0)$ *and* $(-1,0)$ *are saddle points.*

2 Least Squares Problem

Linear least square problem

Given a linear system

$$Ax = b,$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $m > n$. In addition, A has a full column rank, *i.e.* rank$A = n$. In this setting, the system usually has no solution, so instead of solving this system, we can find a point that minimize the square norm of $Ax - b$:

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|^2. \tag{3.4}$$

The problem (3.4) is an unconstrained optimization problem with the objective function

$$f(x) = x^T A^T Ax - 2b^T Ax + \|b\|^2.$$

Since rank$A = n$, we have $\nabla^2 f(x) = A^T A \succ 0$. It means that the objective function f is strictly convex. Hence, the unique stationary point

$$x^* = (A^T A)^{-1} A^T b$$

is the solution of (3.4). The point x^* is called the least square solution of system $Ax = b$.

Example 3.17. *([1]) Consider the inconsistent linear system*

$$\begin{cases} x_1 + 2x_2 &= 0 \\ 2x_1 + x_2 &= 1 \\ x_1 + x_2 &= 1. \end{cases}$$

Let A and b be the coefficients matrix and right-hand-side vector of the system. The least square solution is

$$x^* = (A^T A)^{-1} A^T b = \begin{pmatrix} \frac{8}{11} \\ \frac{-3}{11} \end{pmatrix}.$$

In Octave (Matlab), the least squares solution of an overdetermined linear system $Ax = b$ can be found by using the following command

Octave code

```
A =[1,2;2,1;1,1];
b=[0;1;1];
format rat;
xsol=A\b
```

Another application of linear least square problem is linear fitting: Given m data points (x_i, y_i) for $i = 1, \ldots, m$ where $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$. Suppose that there is a linear relation

$$y_i \approx a^T x_i$$

approximately holds. Our task is to find the parameter $a \in \mathbb{R}^n$ that minimize the distance between y_i and $a^T x_i$ for $i = 1, \ldots, m$. This problem can be formulated as

$$\min_{a \in \mathbb{R}^n} \sum_{i=1}^{m} (y_i - a^T x_i)^2.$$

It can be rewritten as

$$\min_{a\in\mathbb{R}^n} \|Xa - y\|^2,$$

where

$$X = \begin{pmatrix} x_1^T \\ \vdots \\ x_m^T \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}.$$

Example 3.18. *Consider the data points illustrated in Figure 3.4.*

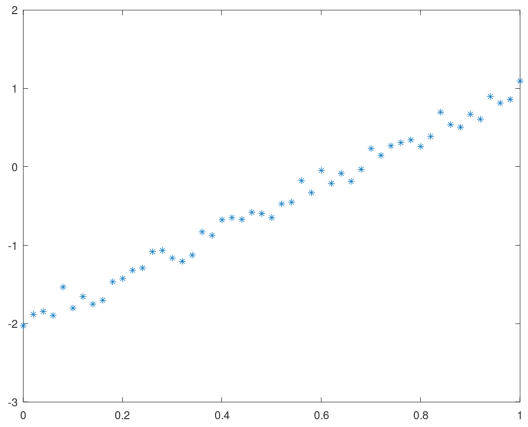


Figure 3.4: Data points

To generate these points, we take the x-coordinate from $[0, 1]$ and $y = 3x - 2 + \epsilon$ where ϵ is a random variable with normal distribution $\mathcal{N}(0, 1)$. Find the lines best fits these points.

Octave code

```
x=(0:0.02:1)';
y=3*x-2+0.1*random("normal", 0,1, [size(x)]);
plot(x,y,"*")
```

```
a=[x,ones(size(x))]\y
figure
plot(x,y,"*");
yapprox=[x,ones(size(x))]*a;
hold on
plot(x,yapprox,"-")
```

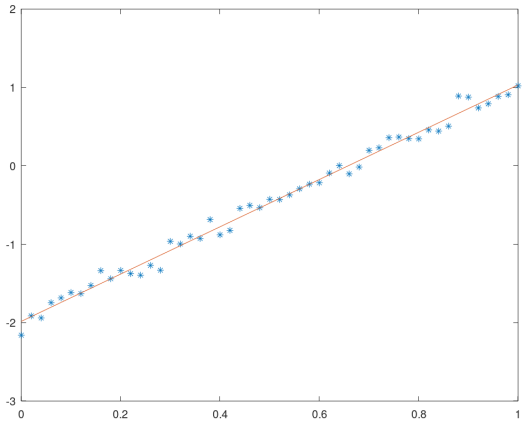


Figure 3.5: Linear fitting

Regularized Least Squares

There are several situations in which the least squares solution does not give rise to a good estimate of the "true" vector x. For example, we consider the linear system $Ax = b$ when $A \in \mathbb{R}^{m \times n}$ is underdetermined: $m < n$. The system usually has many solutions and we don't know which solution should be considered. In these cases, some type of prior information on x should be used in the model. One way to do it is to add a regularization funtion $R(x)$ to

the objective function. The regularized least squares (RLS) problem has the form

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|^2 + \lambda R(x). \tag{3.5}$$

The positive constant λ is the regularization parameter. As λ gets larger, more weight is given to the regularization function.

One application area in which regularization is commonly used is **denoising**. Suppose that a noisy measurement of a signal $x \in \mathbb{R}^n$ is given:

$$b = x + w.$$

Here x is an unknown signal, w is an unknown noise vector, and b is the known measurements vector. The denoising problem is the following: Given b, find a "good" estimate of x. The least squares problem associated with the approximate equations $x \approx b$ is

$$\min_{x\in\mathbb{R}^n} \|x-b\|^2.$$

However, the optimal solution of this problem is obviously $x = b$, which is meaningless. To find a more relevant problem, we will add a regularization term. For that, we need to exploit some a priori information on the signal. For example, we might know in advance that the signal is smooth in some sense. In that case, it is very natural to add a quadratic penalty, which is the sum of the squares of the differences of consecutive components of the vector; that is, the regularization function is

$$R(x) = \sum_{i=1}^{n-1} (x_i - x_{i+1})^2.$$

By letting

$$L = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix},$$

we can write

$$R(x) = \|Lx\|^2.$$

And the regularized least square problem is reformulated as

$$\min_{x \in \mathbb{R}^n} \|x - b\|^2 + \lambda \|Lx\|^2. \tag{3.6}$$

The objective function of (3.6) is

$$f := x^T (I + \lambda L^T L) x - 2b^T x + \|b\|^2.$$

Since this function is strictly convex, the unique solution of (3.6) is

$$x^* = (I + \lambda L^T L)^{-1} b.$$

Example 3.19. *([1])*

Consider the noisy signal constructed by the following code (illustrated by Figure 3.6)

Octave code

```
t=(0:0.01:5)';
x=sin(t)+t.*(cos(t).^2);
b=x+0.05*random("normal", 0,1, [size(x)]);
figure
plot(0:500,b,"LineWidth",2);
```

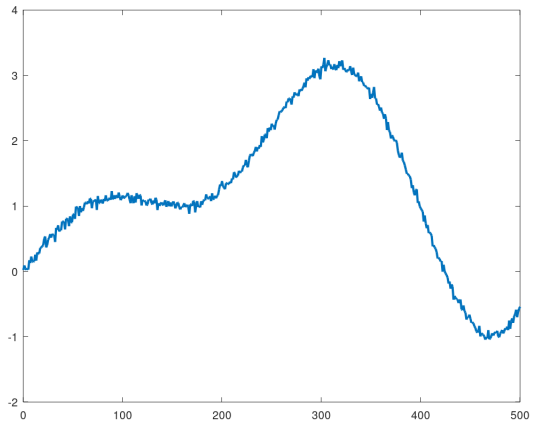


Figure 3.6: Noisy signal

To denoise the signal, we solve the regularized least square problem (3.6) with different values of the regularization parameter $\lambda = 1; 10; 100; 1000$.

Octave code

```
n=length(t);
L=zeros(n-1,n);
for i=1:n-1
  L(i,i)=1;
  L(i,i+1)=-1;
end
x_rls=(eye(n)+1*L'*L)\b;
x_rls=[x_rls,(eye(n)+10*L'*L)\b];
x_rls=[x_rls,(eye(n)+100*L'*L)\b];
x_rls=[x_rls,(eye(n)+1000*L'*L)\b];
figure(2)
for j=1:4
```

```
13    subplot(2,2,j);
14    plot(1:n,x_rls(:,j),"linewidth",2);
15    hold on
16    plot(1:n,x,":r","linewidth",2);
17    hold off
18    title(["\lambda=",num2str(10^(j-1))]);
19  end
```

The signals that we obtained are illustrated in Figure 3.7

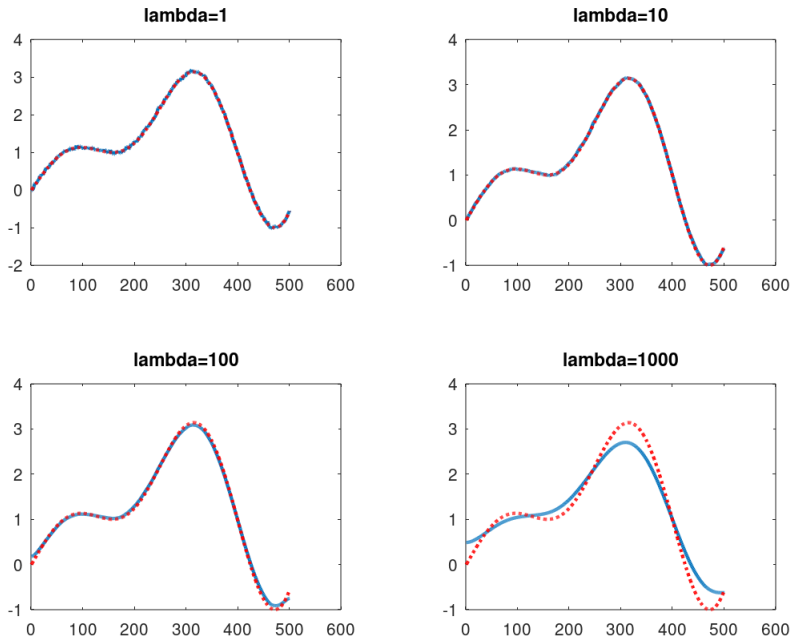


Figure 3.7: Denoising signals

3 The Gradient Method

We consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x). \tag{3.7}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function.

In reality, the analytic form of the function f is unknown. We only have a process to compute the values of f and maybe its derivatives at certain points. We need to design algorithms that identifies a solution reliably and without using too much computer time or storage with these information.

Recall that in Therem 3.10, we prove that if $\overline{x}$ is a local minimum of (3.7) then

$$\nabla f(\overline{x}) = 0.$$

Also in Section 3.1, several examples were presented in which the optimal solution of the problem can be obtained by finding among all the stationary points. But in general, such an approach is not implementable because

(i) it might be a very difficult task to solve the set of (usually nonlinear) equations $\nabla f(x) = 0$;

(ii) even if it is possible to find all the stationary points, it might be that there are infinite number of stationary points and the task of finding the one corresponding to the minimal function value is an optimization problem which by itself might be as difficult as the original problem.

More over, in applications, the function f may be defined algorithmically, that is we only have some procedure to calculate f or ∇f at each point, but we do not have the analytic form of f.

The iterative algorithms that we will consider in this lecture take the form

$$x_{k+1} = x_k + t_k d_k \quad \text{for } k = 0, 1, \ldots,$$

where d_k is the so-called direction and $t_k > 0$ is the stepsize.

There are several questions arised:

- What is the starting point x_0?
- How can we choose the direction d_k and step size t_k.
- Is the sequence $\{x_k\}$ convergent to a solution? In which sense?
- What is the speed of convergence?
- Stopping criterias?

Definition 3.20. *(descent direction). Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^n$. A vector $0 \neq d \in \mathbb{R}^n$ is called a descent direction of f at x if the directional derivative $f'(x; d)$ is negative, meaning that*

$$f'(x; d) = \nabla f(x)^T d < 0.$$

The most important property of descent directions is that taking small enough steps along these directions lead to a decrease of the objective function.

A descent direction method can be described as follow

Algorithm 1: Descent Directions Method

Initialization: Take an arbitrary point $x_0 \in \mathbb{R}^n$.

Step k: For any $k = 0, 1, 2, ...$

- Pick a descent direction d_k .
- Find a stepsize t_k such that $f(x_k + t_k d_k) < f(x_k)$.
- Compute $x_{k+1} = x_k + t_k d_k$.
- If a stopping criterion is satisfied, then STOP.

The process of finding the stepsize t_k is called line search. We look for a minimum of the one-dimensional function $g(t) = f(x_k + td_k)$. There are many choices for stepsize selection rules. We describe here three popular choices:

- **Constant stepsize** $t_k = t$ for any k. The constant stepsize strategy is very simple but it is hard to choose the constant. A large constant might cause the algorithm to be nondecreasing, and a small constant can cause slow convergence of the method.
- **Exact line search:**

  $$t_k \in \text{argmin}_{t \geq 0} f(x_k + td_k).$$

  Finding the exact minimum can be very expensive and is not necessary because the real goal is to find the minimum of f.
- **Backtracking line search** The method requires three parameters: $s > 0, \alpha \in (0, 1), \beta \in (0, 1)$. The choice of t_k is done by the following procedure: starting at

$t = s$, repeat $t := \beta t$ until

$$f(x_k + td_k) < f(x_k) + \alpha t \nabla f(x)^T d_k. \tag{3.8}$$

The condition (3.8) is called the Armijo-Goldstein condition.

In the next lemma, we will show that it is always possible to find t satisfying the Armijo-Goldstein condition.

Lemma 3.21. *([1])*

Let f be a continuously differentiable function over $\mathbb{R}^n$, and let $x \in \mathbb{R}^n$. Suppose that $0 \neq d \in \mathbb{R}^n$ is a descent direction of f at x and let $\alpha \in (0, 1)$. Then there exists $\epsilon > 0$ such that the inequality

$$f(x + td) - f(x) < \alpha t \nabla f(x)^T d$$

holds for all $t \in [0, \epsilon]$.

Gradient method

In the gradient method, the descent direction is choosen by

$$d_k = -\nabla f(x_k).$$

It is clear that if $\nabla f(x_k) \neq 0$ then

$$f'(x_k, -\nabla f(x_k)) = -\|\nabla f(x_k)\|^2 < 0.$$

The gradient method is also called the steepest descent method.

Lemma 3.22. *([1]) Let f be a continuously differentiable function, and let $x \in \mathbb{R}^n$. Assume that $\nabla f(x_k) \neq 0$. Then*

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|} \in \text{argmin}_{\|d\|=1} f'(x, d).$$

Algorithm 2: The Gradient Method

Let ϵ be a positive number.

Initialization: Take $x_0 \in \mathbb{R}^n$ arbitrarily.

Step k: For any $k = 0, 1, \ldots,$

- Pick a stepsize t_k by a line search procedure.
- Compute

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

- If $\nabla f(x_{k+1}) \leq \epsilon$, then STOP.

Implementation

Octave code for the gradient method with constant step-size.

```
function [x,fun_val]=gradientconstant(f,g,x0,t,epsilon)
% Gradient method with constant stepsize
%
% INPUT
%=======================================
% f ......... objective function
% g ......... gradient of the objective function
```

```
% x0......... initial point
% t ......... constant stepsize
% epsilon ... tolerance parameter
% OUTPUT
%=======================================
% x ......... optimal solution (up to a tolerance)
% of min f(x)
% fun_val ... optimal function value
x=x0;
grad=g(x);
iter=0;
while (norm(grad)>epsilon)
  iter=iter+1;
  x=x-t*grad;
  fun_val=f(x);
  grad=g(x);
  fprintf("iter_number = %3d norm_grad = %2.6f fun_val = %2.6f \n",...
  iter,norm(grad),fun_val);
end
```

Octave code for the gradient method with exact line search.

```
function [x,fun_val]=gradientexactlinesearch(f,g,x0,t,epsilon)
% INPUT
% ======================
%  INPUT
%=======================================
```

```
% f ......... objective function
% g ......... gradient of the objective function
% x0......... initial point
% t ......... exact line search stepsize
% epsilon ... tolerance parameter
% OUTPUT
% ========================
% x ....... an optimal solution (up to a tolerance) of
% min f(x)
% fun_val . the optimal function value up to a tolerance
x=x0;
iter=0;
grad=g(x);
while (norm(grad)>epsilon)
iter=iter+1;
x=x-t(x)*grad;
grad=g(x);
fun_val=f(x);
fprintf("iter_number = %3d norm_grad = %2.6f fun_val = %2.6f\n",...
iter,norm(grad),fun_val);
end
```

Octave code for the gradient method with backtracking line search.

```
function [x,fun_val]=gradientbacktracking(f,g,x0,s,alpha,...
beta,epsilon)
% Gradient method with backtracking stepsize rule
```

```
%
% INPUT
%=======================================
% f ......... objective function
% g ......... gradient of the objective function
% x0......... initial point
% s ......... initial choice of stepsize
% alpha ..... tolerance parameter for the stepsize selection
% beta ...... the constant in which the stepsize is multiplied
% at each backtracking step (0<beta<1)
% epsilon ... tolerance parameter for stopping rule
% OUTPUT
%=======================================
% x ......... optimal solution (up to a tolerance)
% of min f(x)
% fun_val ... optimal function value
x=x0;
grad=g(x);
fun_val=f(x);
iter=0;
while (norm(grad)>epsilon)
iter=iter+1;
t=s;
while (fun_val-f(x-t*grad)<alpha*t*norm(grad)^2)
t=beta*t;
end
```

```
x=x-t*grad;
fun_val=f(x);
grad=g(x);
fprintf("iter_number = %3d norm_grad = %2.6f fun_val = %2.6f \n",...
iter,norm(grad),fun_val);
end
```

To test the gradient method we consider the minimization of a quadratic function.

Example 3.23. *([1]) Consider the following optimization problem*

$$\min_{x\in\mathbb{R}^n} x^T Ax + 2b^T x,$$

The gradient of the objective function is

$$\nabla f(x) = 2(Ax + b).$$

Let

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \quad b = (0, 0).$$

For the gradient method with constant stepsize, we take

$$x_0 = (5, 5), \epsilon = 10^{-3}, t_k = 0.1$$

Octave code

```
A=[1,0;0,4];
f=@(x)x'*A*x;
g=@(x)2*A*x;
x_0=[5;5];
```

```
t=0.1;
eps=1e-3;
[x,fun_val]=gradientconstant(f,g,x_0,t,eps);
```

For the exact line search, the stepsize is given by

$$t_k = \text{argmin}_{t\geq 0} f(x_k - t\nabla f(x_k)) = \frac{\|\nabla f(x_k)\|^2}{2\nabla f(x_k)^T A \nabla f(x^k)}.$$

Octave code

```
A=[1,0;0,4];
f=@(x)x'*A*x;
g=@(x)2*A*x;
t=@(x)(norm(g(x),"fro"))^2/(2*g(x)'*A*g(x));
x_0=[5;5];
eps=1e-3;
[x,fun_val]=gradientexactlinesearch(f,g,x_0,t,eps);
```

For the backtracking line search, we take

$$s = 2, \alpha = 0.2, \beta = 0.5.$$

Octave code

```
A=[1,0;0,2];
f=@(x)x'*A*x;
g=@(x)2*A*x;
x_0=[5;5];
s=2; alpha=0.2;  beta=0.5;
eps=1e-3;
[x,fun_val]=gradientbacktracking(f,g,x_0,s,alpha,beta,eps);
```

Example 3.24. *Consider the problem of minimizing Rosenbrock's function*

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Rosenbrock's function is a standard test function in optimization. It has a unique minimum value of 0 *attained at the point* $(1, 1)$.

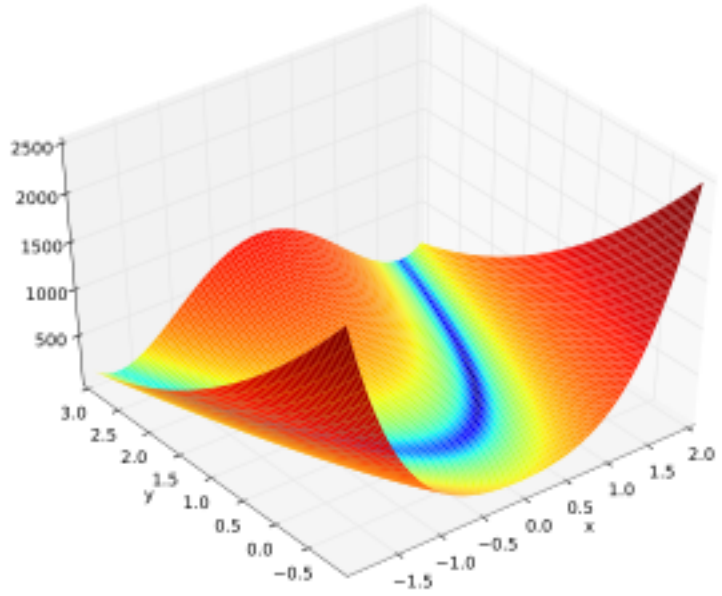


Figure 3.8: Rosenbrock's function (Source: Wikipedia)

Convergence analysis

Assume that the objective function f is continuously differentiable and its gradient ∇f is Lipschitz continuous with constant L on $\mathbb{R}^n$.

Lemma 3.25. *([1]) For any* $x, y \in \mathbb{R}^n$,

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|x - y\|^2.$$

Lemma 3.26. *([1]) Let $\{x_k\}$ be the sequence generated by the gradient method for solving*

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

with one of the following stepsize strategies:

- *constant stepsize $t_k \equiv t \in (0, \frac{1}{2L})$,*
- *exact line search,*
- *backtracking procedure with parameters $s > 0$, $\alpha \in (0,1)$, and $\beta \in (0,1)$.*

Then

$$f(x_k) - f(x_{k+1}) \geq M\|\nabla f(x_k)\|^2, \tag{3.9}$$

where

$$M = \begin{cases} t(1-\frac{tL}{2}) & \text{constant stepsize,} \\ \frac{1}{2L} & \text{exact line search,} \\ \alpha\min\{s, \frac{2(1-\alpha)\beta}{L}\} & \text{backtracking line search.} \end{cases}$$

Theorem 3.27. *([1])*

Assume that f is bounded from below, then

(i) The sequence $\{f(x_k)\}$ is nonincreasing.

(ii) $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$.

Theorem 3.28. *([1]) Let*

$$f^* = \lim_{k\to\infty} f(x_k).$$

Then for any $n = 0, 1, \ldots,$

$$\min_{k=0,1,\ldots,n} \|\nabla f(x_k)\| \leq \sqrt{\frac{f(x_0) - f^*}{M(n+1)}},$$

where

$$M = \begin{cases} t(1 - \frac{tL}{2}) & \text{constant stepsize,} \\ \frac{1}{2L} & \text{exact line search,} \\ \alpha \min\{s, \frac{2(1-\alpha)\beta}{L}\} & \text{backtracking line search.} \end{cases}$$

4 Newton's Method

Newton's method was first published in 1685 in A Treatise of Algebra both Historical and Practical by John Wallis. At first, it is a method for approximating solutions to equations. In optimization, Newton's method is applied to find the roots of the equation $\nabla f(x) = 0$ (stationary point of f).

Consider the unconstrained minimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$. In this section, we assume that the objective function f is twice continuously differentiable.

The main idea of Newton's method is following: Given an iterate x_k, the second-order Taylor expansion of f around x_k is

$$f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k).$$

The next iterate x_{k+1} is choosen as a minimizer of this quadratic approximation in x, that is

$$x_{k+1} \in \text{argmin}_x \{f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)\}. \tag{3.10}$$

We also need to assume that $\nabla^2 f(x_k)$ is positive definite. Then the unique solution of (3.10) is the solution of

$$\nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0.$$

Hence

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

The vector $(\nabla^2 f(x_k))^{-1}\nabla f(x_k)$ is called Newton direction.

Algorithm 3: Newton's Method

Let ϵ be a positive number

Initialization: Take $x_0 \in \mathbb{R}^n$ arbitrarily.

Step k: For any $k = 0, 1, \ldots$

- Compute the Newton direction d_k by solving

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k).$$

- Set $x_{k+1} = x_k + d_k$.

- If $\|\nabla f(x_{k+1})\| \leq \epsilon$, then STOP.

Implementation

Octave code for Newton's method

```
function x=newton(f,g,h,x0,epsilon)

% INPUT
% ==============
% f ..... objective function
% g ..... gradient of the objective function
% h ..... Hessian of the objective function

% x0..... initial point
```

```
% epsilon ..... tolerance parameter
% OUTPUT
% ==============
% x - solution obtained by Newton's method (up to some tolerance)

x=x0;
gval=g(x);
hval=h(x);
iter=0;
while ((norm(gval)>epsilon)&&(iter<10000))
iter=iter+1;
x=x-hval\gval;
fprintf("iter= %2d f(x)=%10.10f\n",iter,f(x))
gval=g(x);
hval=h(x);
end
if (iter==10000)
fprintf("did not converge")
end
```

Octave code for Newton's method with backtracking line search

```
function x=newtonbacktracking(f,g,h,x0,alpha,beta,epsilon)
% Newton's method with backtracking
%
% INPUT
%=======================================
```

```
% f ......... objective function
% g ......... gradient of the objective function
% h ......... hessian of the objective function
% x0......... initial point
% alpha ..... tolerance parameter for the stepsize selection strategy
% beta ...... the proportion in which the stepsize is multiplied
% at each backtracking step (0<beta<1)
% epsilon ... tolerance parameter for stopping rule
% OUTPUT
%=======================================
% x ......... optimal solution (up to a tolerance)
% of min f(x)
% fun_val ... optimal function value
x=x0;
gval=g(x);
hval=h(x);
d=hval\gval;
iter=0;
while ((norm(gval)>epsilon)&&(iter<10000))
iter=iter+1;
t=1;
while(f(x-t*d)>f(x)-alpha*t*gval'*d)
t=beta*t;
end
x=x-t*d;
fprintf("iter= %2d f(x)=%10.10f\n",iter,f(x))
```

```
gval=g(x);
hval=h(x);
d=hval\gval;
end
if (iter==10000)
fprintf("did not converge\n")
end
```

Octave code for Newton-gradient method with backtracking line search

```
function x=hybridnewton(f,g,h,x0,alpha,beta,epsilon)
% Hybrid Newton's method
%
% INPUT
%=======================================
% f ......... objective function
% g ......... gradient of the objective function
% h ......... hessian of the objective function
% x0......... initial point
% alpha ..... tolerance parameter for the stepsize selection strategy
% beta ...... the proportion in which the stepsize is multiplied
% at each backtracking step (0<beta<1)
% epsilon ... tolerance parameter for stopping rule
% OUTPUT
%=======================================
% x ......... optimal solution (up to a tolerance)
% of min f(x)
```

```
% fun_val ... optimal function value
x=x0;
gval=g(x);
hval=h(x);
[L,p]=chol(hval,"lower");
if (p==0)
d=L'\(L\gval);
else
d=gval;
end
iter=0;
while ((norm(gval)>epsilon)&&(iter<10000))
iter=iter+1;
t=1;
while(f(x-t*d)>f(x)-alpha*t*gval'*d)
t=beta*t;
end
x=x-t*d;
fprintf("iter= %2d f(x)=%10.10f\n",iter,f(x))
gval=g(x);
hval=h(x);
[L,p]=chol(hval,"lower");
if (p==0)
d=L'\(L\gval);
else
d=gval;
```

```
end
end
if (iter==10000)
fprintf("did not converge\n")
end
```

Example 3.29. *We test these methods with the optimization problems given in Examples 3.23 and 3.24.*

Convergence analysis

Theorem 3.30. *([1]) Assume that*

- *there exists $m > 0$ for which $\nabla^2 f(x) \succeq mI$ for any $x \in \mathbb{R}^n$,*
- *there exists $L > 0$ for which $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|$ for any $x, y \in \mathbb{R}^n$.*

Let $\{x_k\}$ be the sequence generated by Newton's method, and let x^* be the unique minimizer of f. Then for any k, we have

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

In addition, if $\|x_0 - x^*\| \leq \frac{m}{L}$, then

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

5 Quasi-Newton methods

Quasi-Newton methods are based on Newton's method but it requires only the gradient of the objective function at each iterate. It does not require to compute the Hessian matrix. Instead, the Hessian matrix is approximated using updates specified by gradient evaluations.

The first quasi-Newton method was developed by a physicist, Davidon in 1959. After that, in 1963, Fletcher and Powell proved that this algorithm is much faster and more reliable than the other existing methods. Nowadays, the most popular Quasi-Newton method is the BFGS method that is suggested independently by Broyden, Fletcher, Goldfarb, and Shanno, in 1970.

In the BFGS method, at the iteration k, the direction d_k is choosen as a solution of

$$B_k d_k = -\nabla f(x_k),$$

where B_k is an approximation of the Hessian matrix which is update iteratively at each step. The next iterate is

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

where the step length t_k is chosen by a line search procedure.

B_{k+1} is choosen such that

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

Set $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. We obtain

$$B_{k+1} s_k = y_k.$$

We also want that B_{k+1} is symmetric and positive definite. Let us denote by H_{k+1} the inverse of B_{k+1}. In the BFGS method, at the iterate k, H_{k+1} is choosen such that it is symmetric and positive definite, closet to the current H_k and

$$H_{k+1}y_k = s_k.$$

In the other word, to find H_{k+1}, we solve the following optimization problem

$$\begin{aligned} \min \quad & \|H - H_k\| \\ \text{s.t.} \quad & Hy_k = s_k, \\ & H = H^T, \quad H \succeq 0. \end{aligned}$$

The unique solution of the above problem is

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \tag{3.11}$$

where $\rho_k = \frac{1}{y_k^T s_k}$.

Algorithm 4: BFGS method

Given $\epsilon > 0$.

Initial: Given starting point x_0, inverse Hessian approximation H_0.

Step k:

- Compute

$$d_k = -H_k \nabla f(x_k).$$

- Pick a stepsize t_k by a line search procedure.

- Compute

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

- Set $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ and compute

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T.$$

- If $\nabla f(x_{k+1}) \leq \epsilon$, then STOP.

Exercises

1. Compute the gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ of the Rosenbrock function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

   Show that $(1, 1)^T$ is the only local minimizer of this function, and that the Hessian matrix at that point is positive definite.

2. Find the global minimum and maximum points of the function $f(x, y) = x^2+y^2+2x-3y$ over the unit ball $B[0, 1] = \{(x, y) : x^2 + y^2 \leq 1\}$.

3. Let $a \in \mathbb{R}^n$ be a nonzero vector. Show that the maximum of $a^T x$ over $B[0, 1]$ is attained at $x^* = \frac{a}{\|a\|}$ and that the maximal value is $\|a\|$.

4. Find the global minimum and maximum points of the function $f(x, y) = 2x - 3y$ over the set $S = \{(x, y) : 2x^2 + 5y^2 \leq 1\}$.

5. For each of the following functions, find all the stationary points and classify them according to whether they are saddle points, strict/nonstrict local/global minimum/-maximum points:

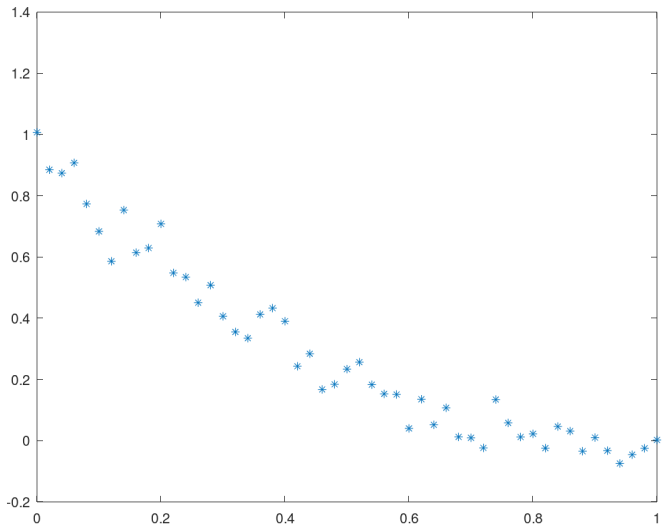
   - $f(x_1, x_2) = (4x_1^2 - x_2)^2$.
   - $f(x_1, x_2, x_3) = x_1^4 - 2x_1^2 + x_2^2 + 2x_2x_3 + 2x_3^2$.
   - $f(x_1, x_2) = 2x_2^3 - 6x_2^2 + 3x_1^2x_2$.
   - $f(x_1, x_2) = x_1^4 + 2x_1^2x_2 + x_2^2 - 4x_1^2 - 8x_1 - 8x_2$.

6. Let f be twice continuously differentiable function over $\mathbb{R}^n$. Suppose that $\nabla^2 f(x) \succ 0$ for any x. Prove that a stationary point of f is necessarily a strict global minimum point.

7. Prove that all isolated local minimizers are strict.

8. Generates points by the following Octave code

```
x=0:0.02:1;
y=x.^2-2*x+1+0.05*random("normal", -1,1, [size(x)]);
plot(x,y, "*")
```



Find the quadratic function $y = ax^2 + bx + c$ that best fits the points in the least squares sense. Indicate what are the parameters a, b , c found by the least squares solution, and plot the points along with the derived quadratic function.

9. Write a function *circle_fit* whose input is an $n \times m$ matrix A; the columns of A are

the m vectors in $\mathbb{R}^n$ to which a circle should be fitted. The call to the function will be of the form $[x, r] = circle\_fit(A)$. The output (x, r) is the optimal solution of

$$\min_{x\in\mathbb{R}^n, r\in\mathbb{R}} \sum_{i=1}^{m} (\|x - a_i\|^2 - r^2)^2. \tag{3.12}$$

Use the code in order to find the best circle fit in this sense of the 5 points

$$a_1 = \begin{pmatrix} 0 \\ \\ 0 \end{pmatrix}; a_2 = \begin{pmatrix} 0.5 \\ \\ 0 \end{pmatrix}; a_3 = \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix}; a_4 = \begin{pmatrix} 1 \\ \\ 1 \end{pmatrix}; a_5 = \begin{pmatrix} 0 \\ \\ 1 \end{pmatrix}.$$

10. Consider the quadratic minimization problem

$$\min\{x^T Ax : x \in \mathbb{R}^5\},$$

where A is the 5×5 Hilbert matrix defined by

$$a_{i,j} = \frac{1}{i + j - 1}.$$

Run the following methods and compare the number of iterations required by each of the methods when the initial vector is $x_0 = (1, 2, 3, 4, 5)^T$ to obtain a solution x with $\nabla f(x) \leq 10^{-4}$

- gradient methodwith backtracking stepsize rule and parameters $\alpha = 0.5$, $\beta = 0.5$, $s = 1$;
- gradient methodwith backtracking stepsize rule and parameters $\alpha = 0.1$, $\beta = 0.5$, $s = 1$;
- gradient method with exact line search.

11. Use the gradient method and Newton's method to minimize the Rosenbrock function with the initial point $x_0 = (1.5, 1.5)$.

4

Constrained Optimization

In this chapter, we consider the constrained optimization problem

$$\begin{aligned} \min \quad & f(x) \\ & x \in C \end{aligned} \tag{4.1}$$

where $C \subseteq \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$.

1 Convex Optimization problem

In this section, we assume that the constrained set C is convex and the objective function f is convex on C.

Linear programming

A linear programming problem is an optimization problem consisting of minimizing a linear objective function subject to linear equalities and inequalities:

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & Ax \leq b \end{aligned}$$

Clearly, a linear programming problem is a convex optimization problem.

Convex Quadratic Problems

Convex quadratic problems are problems consisting of minimizing a convex quadratic function subject to affine constraints:

$$\begin{aligned} \min \quad & x^T Q x + 2b^T x \\ s.t. \quad & Ax \leq b. \end{aligned}$$

Theorem 4.1. *Let $f : C \to \mathbb{R}$ be a convex function defined on the convex set C . Let $x^* \in C$ be a local minimum of f on C . Then x^* is a global minimum of f on C .*

Theorem 4.2. *The solution set X^* of the problem (4.1) is convex. If, in addition, f is strictly convex on C, then (4.1) has at most one solution.*

Theorem 4.3. *Let f be a continuously differentiable function over a closed convex set C, and let x^* be a local minimum of (4.1). Then*

$$\nabla f(x^*)^T (x - x^*) \geq 0 \quad \forall x \in C. \tag{4.2}$$

More over, if f is convex on C then $x^* \in C$ is a global minimum of (4.1) if and only if it statisfies condition (4.2).

2 The Gradient Projection Method

Assume that the constrained set C is closed and convex. In Theorem 4.3, we prove that if x^* is a local minimum of (4.1) then it satisfies condition (4.2), that is

$$\nabla f(x^*)^T(x - x^*) \geq 0 \quad \forall x \in C.$$

Let λ be a positive number. By Theorem 2.21, the above inequality equivalents with

$$x^* = P_C(x^* - \lambda \nabla f(x^*)). \tag{4.3}$$

The gradient projection method is built based on this condition.

Algorithm 5: The Gradient Projection Method

Let ϵ be a positive number.

Initialization: Take $x_0 \in C$ arbitrarily.

Step k: For any $k = 0, 1, \ldots$

- Pick a stepsize t_k by a line search procedure.
- Compute

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

- If $\|x_k - x_{k+1}\| \leq \epsilon$ then STOP.

Implementation

Example 4.4. *([1]) Consider the following optimization problem*

$$\begin{aligned} \min \quad & x^T Ax + 2b^T x \\ \text{such that} \quad & lb \leq x \leq ub. \end{aligned}$$

Octave code for the projection gradient method with constant stepsize:

```
function [x,fun_val]=progradientconstant(f,g,lb,ub,x0,t,epsilon)
% Gradient method with constant stepsize
%
% INPUT
%=======================================
% f ......... objective function
% g ......... gradient of the objective function
% lb ........ lower bound
% ub ........ upper bound
% x0......... initial point
% t ......... constant stepsize
% epsilon ... tolerance parameter
% OUTPUT
%=======================================
% x ......... optimal solution (up to a tolerance)
% of min f(x)
% fun_val ... optimal function value
x=x0;
grad=g(x);
iter=0;er=1;
```

```
while (norm(er)>epsilon)
  iter=iter+1;
  y=x-t*grad;
  for i=1:length(y)
    if y(i)<lb(i)
      y(i)=lb(i);
    end
    if y(i)>ub(i)
      y(i)=ub(i);
    end
  end
  er=y-x;
  x=y;
  fun_val=f(x);
  grad=g(x);
  fprintf("iter_number = %3d norm(xk+1-xk) = %2.6f fun_val = %2.6f \n",...
  iter,norm(er),fun_val);
end
```

Octave code for the projection gradient method with backtracking linesearch:

```
function [x,fun_val]=progradientbacktracking(f,g,lb,ub,x0,s,alpha,...
beta,epsilon)
% Gradient method with backtracking stepsize rule
%
% INPUT
%=======================================
```

```
% f ......... objective function
% g ......... gradient of the objective function
% lb ........ lower bound
% ub ........ upper bound
% x0......... initial point
% s ......... initial choice of stepsize
% alpha ..... tolerance parameter for the stepsize selection
% beta ...... the constant in which the stepsize is multiplied
% at each backtracking step (0<beta<1)
% epsilon ... tolerance parameter for stopping rule
% OUTPUT
%=======================================
% x ......... optimal solution (up to a tolerance)
% of min f(x)
% fun_val ... optimal function value
x=x0;
fun_val=f(x);
grad=g(x);
iter=0;er=1;
while (norm(er)>epsilon)
  iter=iter+1;
  t=s;
  while (fun_val-f(x-t*grad)<alpha*t*norm(grad)^2)
    t=beta*t;
  end
  y=x-t*grad;
```

```
  for i=1:length(y)
    if y(i)<lb(i)
      y(i)=lb(i);
    end
    if y(i)>ub(i)
      y(i)=ub(i);
    end
  end
  er=y-x;
  x=y;
  fun_val=f(x);
  grad=g(x);
  fprintf("iter_number = %3d norm(xk+1-xk)= %2.6f fun_val = %2.6f \n",...
  iter,norm(er),fun_val);
end
```

Let

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \quad b = (0,0),$$
$$lb = (1,1); \quad ub = (5,5).$$

For the gradient method with constant stepsize, we take

$$x_0 = (5,5), \epsilon = 10^{-3}, t_k = 0.1$$

```
A=[1,0;0,4];
f=@(x)x'*A*x;
g=@(x)2*A*x;
```

```
lb=[1;1];
ub=[5;5];
x_0=[5;5];
t=0.1;
eps=1e-3;
[x,fun_val]=progradientconstant(f,g,lb,ub,x_0,t,eps);
```

For the backtracking line search, we take

$$s = 2, \alpha = 0.2, \beta = 0.5.$$

```
A=[1,0;0,4];
f=@(x)x'*A*x;
g=@(x)2*A*x;
lb=[1;1];
ub=[5;5];
x_0=[5;5];
t=0.1;
eps=1e-3;
alpha=0.2;
beta=0.5;
[x,fun_val]=progradientbacktracking(f,g,lb,ub,x_0,t,alpha,beta,eps);
```

Convergence analysis

3 KKT Conditions

KKT Conditions for Linearly Constrained Problems

Theorem 4.5. *([1]) Consider the minimization problem*

$$(P) \quad \begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & a_i^T x \leq b_i \quad \forall i = 1, \ldots, m \end{aligned}$$

where f is a continuously differentiable function on $\mathbb{R}^n$, $a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$ for $i = 1, \ldots, m$. Let x^* be a feasible point of (P).

a. If x^* be a local minimum point of (P), then there exist $\lambda_1, \ldots, \lambda_m \geq 0$ such that

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i a_i = 0 \tag{4.4}$$

and

$$\lambda_i (a_i^T x^* - b_i) = 0, \quad \forall i = 1, 2, ..., m. \tag{4.5}$$

b. In addition, if f is convex, then x^* is an optimal solution of (P) if and only if there exist there exist $\lambda_1, \ldots, \lambda_m \geq 0$ satisfy conditions (4.4) and (4.5).

Theorem 4.6. *([1]) Consider the minimization problem*

$$(Q) \quad \begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & a_i^T x \leq b_i \quad \forall i = 1, \ldots, m \\ & c_j^T x = d_j \quad \forall j = 1, \ldots, p. \end{aligned}$$

where f is a continuously differentiable function on $\mathbb{R}^n$, $a_i, c_j \in \mathbb{R}^n, b_i, d_j \in \mathbb{R}$ for $i = 1, \ldots, m$,$j = 1, \ldots, p$. Let x^* be a feasible solution of (Q).

- *If x^* is a local solution then there exist $\lambda_1, \ldots, \lambda_m \geq 0$, $\mu_1, \ldots, \mu_p$ such that*

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i a_i + \sum_{j=1}^{p} \mu_j c_j = 0 \tag{4.6}$$

and

$$\lambda_i (a_i^T x^* - b_i) = 0, \quad \forall i = 1, 2, ..., m. \tag{4.7}$$

- *In addition, if f is convex then x^* is an optimal solution of (Q) if and only if there exist $\lambda_1, \ldots, \lambda_m \geq 0$, $\mu_1, \ldots, \mu_p$ satisfy conditions (4.6) and (4.7).*

Example 4.7. *([1]) Consider the problem*

$$\begin{aligned} \min \quad & \tfrac{1}{2}(x_1^2 + x_2^2 + x_3^2) \\ \text{s.t.} \quad & x_1 + x_2 + x_3 = 3. \end{aligned}$$

Since the problem is convex, the KKT conditions are necessary and sufficient. The KKT conditions are

$$x_1 + \mu = x_2 + \mu = x_3 + \mu = 0$$

$$x_1 + x_2 + x_3 = 3.$$

We obtained that the unique solution of the KKT system is $x_1 = x_2 = x_3 = 1, \mu = -1$. Hence, the unique optimal solution of the problem is $(x_1, x_2, x_3) = (1, 1, 1)$.

Example 4.8. *([1]) Consider the problem*

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 + 4x_1x_2 \\ \text{s.t.} \quad & x_1 + x_2 = 1, \\ & x_1, x_2 \geq 0. \end{aligned}$$

The objective function is nonconvex. The KKT conditions are

$$2x_1 + 4x_2 + \mu - \lambda_1 = 0$$

$$4x_2 + 4x_1 + \mu - \lambda_2 = 0$$

$$\lambda_1 x_1 = \lambda_2 x_2 = 0$$

$$x_1 + x_2 = 0$$

$$x_1, x_2 \geq 0$$

$$\lambda_1, \lambda_2 \geq 0.$$

Solving this system, we obtain two KKT points $(1, 0)$ and $(0, 1)$. Since the objective function is continuous and the feasible domain is closed and bounded, there exist a solution for our minimization problem. In addtion, $f(1, 0) = 1$, $f(0.1) = 2$, therefore $(1, 0)$ is the global solution of the problem.

Example 4.9. *([1]) Given $y \in \mathbb{R}^n$, we want to find its projection onto an affine space given by $\{x \in \mathbb{R}^n | \quad Ax = b\}$. We assume that A has full row rank. This problem can be formulated as*

$$\begin{aligned} \min \quad & \|x - y\|^2 \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

The objective function is convex and the only constraint is affine, so the KKT conditions are necessary and sufficient. The KKT conditions are

$$2(x - y) + A^T \mu = 0,$$

$$Ax = b.$$

Solving this system, we obtain

$$x = y - A^T(AA^T)^{-1}(Ay - b).$$

The Lagrange function and KKT conditions

Consider the general nonlinear programming problems:

$$(NLP) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_i(x) \leq 0 \quad \forall i = 1, \ldots, m \\ & h_j(x) = 0 \quad \forall j = 1, \ldots, p, \end{array}$$

where $f, g_1, \ldots, g_m, h_1, \ldots, h_p$ are continuously differentiable functions.

The associated Lagrangian function takes the form

$$L(x, \lambda, \mu) = f(x) + \sum_{i1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x).$$

Slater's constraint qualification condition: The function g_i are convex for any $i = 1 \ldots, m$, the function h_i are affine for $j = 1, \ldots, p$ and there exists a feasible point x_s such that $g_i(x_s) < 0$ for all $i = 1, \ldots, m$.

Theorem 4.10. *([5]) Let x^* be a local minimum of the problem (NLP). Assume that the Slater's constraint qualification condition is satisfied. Then there exists multipliers $\lambda_i \geq 0$ for $i = 1, \ldots, m$ and $\mu_j \in \mathbb{R}$ for $j = 1, \ldots, p$ such that*

$$\nabla_x L(x^*, \lambda, \mu) = \nabla f(x^*) + \sum_{i=}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j \nabla h_j(x^*), \tag{4.8}$$

and

$$\lambda_i g_i(x^*) = 0 \quad \forall i = 1, \ldots, m. \tag{4.9}$$

Remark 4.11. *The Slater's constraint qualification condition can be replaced by other constraint qualification conditions such as Robinson's condition and Mangasarian-Fromovitz constraint qualification condition (see [5]. Here, we introduce one of them:*

Linear independence constraint qualification (LICQ): *Let*

$$I(x^*) = \{i| \quad g_i(x^*) = 0\}$$

be the set of active constraints. Suppose that the gradients of the active constraints and the equality constraints

$$\{\nabla g_i(x^*) : i \in I(x^*)\} \cup \{\nabla h_j(x^*) : j = 1, 2, ..., p\}$$

are linearly independent.

Theorem 4.12. *Assume that f is convex, continuous at some feasible point and Slater's condition is satisfied. If x^* is a feasible point and there exist multipliers $\lambda_i \geq 0$ for $i = 1, \ldots, m$ and $\mu_j \in \mathbb{R}$ for $j = 1, \ldots, p$ satisfied conditions (4.8) and (4.9), then x^* is the global minimum of (NLP).*

Example 4.13. *([1]) Consider the problem*

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 = 1. \end{aligned}$$

The objective function is continuous and the feasible domain is closed and bounded, so there exists at least one solution of this problem. In here, there is no inequality constraint, the only equality constraint is

$$h_1(x_1, x_2) = x_1^2 + x_2^2 - 1.$$

For any feasible point, $\nabla h_1(x_1, x_2) \neq 0$. Therefore, the linear independence constraint qualification is satisfied. The KKT conditions are

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= 1 + 2\lambda x_1 = 0, \\
\frac{\partial L}{\partial x_2} &= 1 + 2\lambda x_2 = 0, \\
x_1^2 + x_2^2 &= 1.
\end{aligned}$$

There are two KKT points $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$. The optimal solution of this problem is $(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$.

Example 4.14. *([4]) Consider the convex optimization problem*

$$\begin{aligned}
\min \quad & 4x_1^2 + x_2^2 - x_1 - 2x_2 \\
\text{s.t.} \quad & 2x_1 + x_2 \leq 1 \\
& x_1^2 \leq 1.
\end{aligned}$$

Slater's condition is satisfied at $(0, 0)$. The KKT conditions are

$$\begin{aligned}
\frac{\partial L}{\partial x_1} = 8x_1 - 1 + 2\lambda_1 + 2\lambda_2 x_1 &= 0, \\
\frac{\partial L}{\partial x_2} = 2x_2 - 2 + \lambda_1 &= 0, \\
\lambda_1(2x_1 + x_2 - 1) &= 0, \\
\lambda_2(x_1^2 - 1) &= 0, \\
2x_1 + x_2 &\leq 1, \\
x_1^2 &\leq 1, \\
\lambda_1, \lambda_2 &\geq 0.
\end{aligned}$$

By solving the KKT condition, we obtain Tthe unique optimal solution of the problem $(1/16, 7/8)$.

Example 4.15. *([4]) Consider the constrained least square problem*

$$(CLS) \quad \begin{aligned} \min \quad & \|Ax - b\|^2 \\ \text{s.t.} \quad & \|x\|^2 \leq \alpha, \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$ *is assumed to be of full column rank,* $b \in \mathbb{R}^m$*, and* $\alpha > 0$.

Slater's condition is satisfied at 0. *The KKT conditions are*

$$\begin{aligned} \nabla_x L = 2A^T(Ax - b) + 2\lambda x = 0, \\ \lambda(\|x\|^2 - \alpha) = 0, \\ \|x\|^2 \leq \alpha, \\ \lambda \geq 0. \end{aligned}$$

4 Sequential Quadratic Programming

In this section, we study the sequential quadratic programming (SQP) -one of the most effective methods for solving nonlinearly constrained optimization problem that generates steps by solving quadratic subproblems. SQP methods were first proposed in 1963 by Wilson and then developed by many mathematicians.

Consider the general nonlinear optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad \forall i = 1, \ldots, m, \\ & h_j(x) = 0 \quad \forall j = 1, \ldots, p. \end{aligned} \tag{4.10}$$

The SQP method can be viewed as a generalization of Newton's method for unconstrained optimization in that it finds a step away from the current point by minimizing a quadratic model of the problem. The SQP method replaces the objective function by the quadratic approximation

$$q_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 xx L(x_k, \lambda_k) d$$

and replaces the constraint functions by linear approximations. Many software packages (NPSOL, NLPQL, OPSYC, OPTIMA, MATLAB, and SQP) are based on this approach. Precisely, at the iterate (x_k, λ_k), we approximate (4.10) by the following quadratic optimization problem

$$\begin{aligned} \min_d \quad & f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2_{xx} L(x_k, \lambda_k) d \\ \text{s.t.} \quad & \nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad \forall i = 1, \ldots, m, \\ & \nabla h_j(x_k)^T d + h_j(x_k) = 0 \quad \forall j = 1, \ldots, p. \end{aligned} \tag{4.11}$$

Let d_k be the solution of (4.11) and λ_{k+1} be the corresponding Lagrange multiplier of (4.11). The next iterate is given by

$$(x_{k+1}, \lambda_{k+1}) = (x_k + d_k, \lambda_{k+1}).$$

To produce practical SQP algorithms, we need various ingredients to ensure that subproblems are always feasible, ... For more details about practical SQP algorithms, see [4].

5 Penalty Methods

Consider the constrained optimization problem

$$\min_{x \in C} f(x), \tag{4.12}$$

where C is a closed set in $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$.

Definition 4.16. *Let $P : \mathbb{R}^n \to \mathbb{R}$. P is called a penalty function of (4.12) if it satisfies the following property*

$$P(x) = 0 \quad \forall x \in C, \tag{4.13}$$

$$P(x) > 0 \quad \forall x \notin C. \tag{4.14}$$

Problem (4.12) can be reformulated as an unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} \left[\Phi_\theta(x) := f(x) + \theta P(x) \right], \tag{4.15}$$

where θ is a positive number. θ is called a penalty parameter.

Lemma 4.17. *([5]) If for some $\theta > 0$, (4.15) has a solution x^* which belongs to C, then x^* is also a solution of (4.12).*

Theorem 4.18. *([5]) Assume that (4.12) has a solution. Let $\{\theta_k\}$ be a sequence of positive numbers such that $\lim_{k\to\infty} \theta_k = \infty$ and assume that for any k, (4.15) has a solution x^k. Then every accumulation point of the sequence $\{x_k\}$ is a solution of (4.12).*

Example 4.19. *Consider the following optimization problem*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad \forall i = 1, \ldots, m, \\ & h_j(x) = 0 \quad \forall j = 1, \ldots, p. \end{aligned} \tag{4.16}$$

The function

$$P(x) := \frac{1}{2} \sum_{i=1}^{m} \left[\max(0, g_i(x))\right]^2 + \frac{1}{2} \sum_{j=1}^{p} \left[h_j(x)\right]^2$$

is a penalty function of (4.16) and called quadratic penalty function.

Algorithm 6: Penalty Method

Let $\epsilon > 0$ and $\{\theta_k\}$ be a sequence of positive numbers such that $\lim_{k\to\infty} \theta_k = \infty$.

Initialization: Take $x_0 \in \mathbb{R}^n$.

Step k: For $k = 1, 2, \ldots,$

- Compute

$$x_k \in \operatorname{argmin}_{x\in\mathbb{R}^n} \left[\Phi_{\theta_k}(x) := f(x) + \theta_k P(x)\right].$$

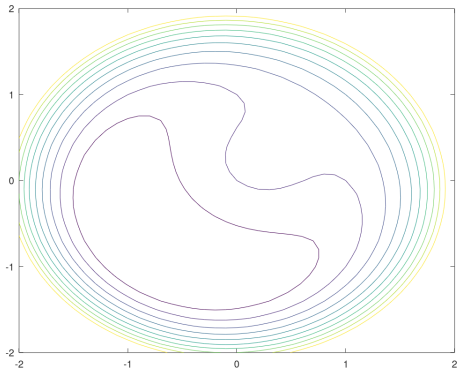
- If $P(x_k) = 0$ or $\|x_k - x_{k-1}\| \leq \epsilon$ then STOP

Example 4.20. *Consider the optimization problem in Example 4.13, that is*

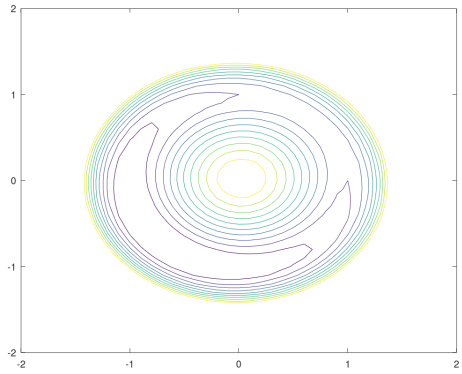
$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 = 1. \end{aligned} \tag{4.17}$$

Recall that the solution of this problem is $(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$. *The quadratic penalty function is*

$$P(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2.$$



(a) $\theta = 1$



(b) $\theta = 10$

Figure 4.1: Contour of the corresponding unconstrained function

The corresponding unconstrained function of this problem is

$$\Phi_\theta(x) := x_1 + x_2 + \theta(x_1^2 + x_2^2 - 1)^2.$$

We plot the contour of this function correspoding to different values of θ in Figure 4.1.

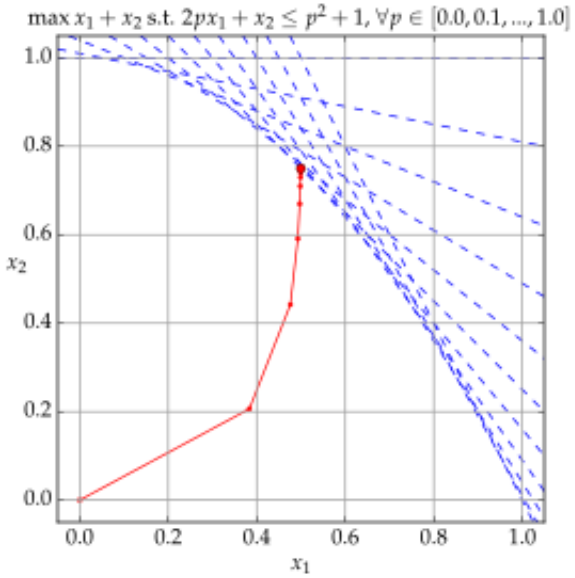


Figure 4.2: Interior point methods

(Source: https://en.wikipedia.org/wiki/Interior-point_method)

6 Interior point methods

Interior-point methods (also known as barrier methods) are a class of algorithms that solve linear and nonlinear convex optimization problems. The interior methods for linear programming was developed by Kamarkar in 1984. Different from the simplex method, it reaches a best solution by traversing the interior of the feasible region.

Consider the following nonlinear optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{4.18}$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}$ are twice continuously differentiable. Assume that there exists a point

x_s such that $g_i(x_s) < 0$ for all $i = 1, \ldots, m$.

By using slack variables, we can transform problem (4.18) to

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) + z_i = 0, \quad i = 1, \ldots, m, \\ & z_i \geq 0, \quad i = 1, \ldots, m. \end{aligned} \tag{4.19}$$

The idea of the interior point methods is to eliminate the inequality constraints $z_i \geq 0$ by using the logarithmic barrier function as follows

$$\begin{aligned} \min \quad & f(x) - \sigma \textstyle\sum_{i=}^{m} \log(z_i) \\ \text{s.t.} \quad & g_i(x) + z_i = 0, \quad i = 1, \ldots, m, \end{aligned} \tag{4.20}$$

The logarithmic barrier function $-\log(z_i)$ prevents z_i from becoming to close to the boundary.

Lemma 4.21. *([5]) Assume that the feasible set of problem 4.18 is bounded. Then for every $\sigma > 0$, problem (4.20) has a solution $(x(\sigma), z(\sigma))$.*

Definition 4.22. *The funtion $\sigma \mapsto x(\sigma)$ is called the central path.*

Theorem 4.23. *([5]) Assume that the feasible set X of problem (4.18) is bounded and the closure of the set*

$$S = \{(x, z) \in X \times \mathbb{R}^m_+ : g(x) = z, z_i > 0, i = 1, \ldots, m\}$$

is the feasible set of problem (4.19). If $\sigma_k \to 0$ when $k \to \infty$, then every accumulation point of the sequence $(x(\sigma_k), z(\sigma_k))$ is a solution of problem (4.19).

Exercises

1. Consider the optimization problem

$$\begin{aligned} \min \quad & x_1 - 4x_2 + x_3 \\ \text{s.t.} \quad & x_1 + 2x_2 + 2x_3 = -2, \\ & x_1^2 + x_2^2 + x_3^2 \leq 1. \end{aligned}$$

   (i) Given a KKT point of problem (P), must it be an optimal solution?

   (ii) Find the optimal solution of the problem using the KKT conditions.

2. Consider the optimization problem

$$\begin{aligned} \min \quad & x_1^4 - 2x_2^2 - x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 + x_2 \leq 0. \end{aligned}$$

   (i) Is the problem convex?

   (ii) Prove that there exists an optimal solution to the problem.

   (iii) Find all the KKT points. For each of the points, determine whether it satisfies the second order necessary conditions.

   (iv) Find the optimal solution of the problem.

3. Consider the optimization problem

$$\begin{aligned} \min \quad & x_1^2 - x_2^2 - x_3^2 \\ \text{s.t.} \quad & x_1^4 + x_2^4 + x_3^4 \leq 1. \end{aligned}$$

   (i) Is the problem convex?

(ii) Find all the KKT points of the problem.

(iii) Find the optimal solution of the problem.

4. Use the KKT conditions in order to find an optimal solution of the each of the following problems:

(i)

$$\begin{aligned} \min \quad & 3x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1 - x_2 + 8 \leq 0, \\ & x_2 \geq 0. \end{aligned}$$

(ii)

$$\begin{aligned} \min \quad & 3x_1^2 + x_2^2 \\ \text{s.t.} \quad & 3x_1^2 + x_2^2 + x_1 + x_2 + 0.1 \leq 0, \\ & x_2 + 10 \geq 0. \end{aligned}$$

(iii)

$$\begin{aligned} \min \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & 4x_1^2 + x_2^2 - 2 \leq 0, \\ & 4x_1 + x_2 + 3 \leq 0. \end{aligned}$$

(iv)

$$\begin{aligned} \min \quad & x_1^3 + x_2^3 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 1. \end{aligned}$$

(v)

$$\begin{aligned} \min \quad & x_1^4 - x_2^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 1, \\ & 2x_2 + 1 \leq 0. \end{aligned}$$

5. (i) Find a formula for the orthogonal projection of a vector $y \in \mathbb{R}^3$ onto the set

$$C = \{x \in \mathbb{R}^3 : x_1^2 + 2x_2^2 + 3x_3^2 \leq 1\}.$$

The formula should depend on a single parameter that is a root of a strictly decreasing one-dimensional function.

(ii) Write a Octave function whose input is a three-dimensional vector and its output is the orthogonal projection of the input onto C.

6. Consider the convex optimization problem

$$(P) \qquad \begin{aligned} \min \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, i = 1, 2, \ldots, m, \end{aligned}$$

where f_0 is a continuously differentiable convex function and $f_1, f_2, \ldots, f_m$ are continuously differentiable strictly convex functions. Let x^* be a feasible solution of (P). Suppose that the following condition is satisfied: there exist $y_i \geq 0$ for $i \in 0 \cap I(x^*)$, which are not all zeros such that

$$y_0 \nabla f_0(x^*) + \sum_{i \in I(x^*)} y_i \nabla f_i(x^*) = 0.$$

Prove that x^* is an optimal solution of (P).

7. Consider the optimization problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & f_i(x) \le 0, \quad i = 1, 2, \ldots, m, \end{aligned}$$

where $c \neq 0$ and $f_1, f_2, \ldots, f_m$ are continuous over $\mathbb{R}^n$. Prove that if x^* is a local minimum of the problem, then $I(x^*) \neq \emptyset$.

8. Consider the QCQP problem

$$(QCQP) \qquad \begin{aligned} \min \quad & x^T A_0 x + 2b_0^T x \\ \text{s.t.} \quad & x^T A_i x + 2b_i^T x + c_i \le 0, \quad i = 1, 2, \ldots, m, \end{aligned}$$

where $A_0, A_1, \ldots, A_m \in \mathbb{R}^{n \times n}$ are symmetric matrices, $b_0, b_1, \ldots, b_m \in \mathbb{R}^n$, and $c_1, c_2, \ldots, c_m \in \mathbb{R}$. Suppose that x^* satisfies the following condition: there exist $\lambda_1, \lambda_2, \ldots, \lambda_m \ge 0$ such that

$$(A_0 + \sum_{i=1}^m \lambda_i A_i) x^* + (b_0 + \sum_{i=1}^m \lambda_i b_i) = 0,$$

$$\lambda_i - \left[(x^*)^T A_i (x^*) + 2b_i^T x^* + c_i \right] = 0, \quad i = 1, 2, \ldots, m,$$

$$(x^*)^T A_i (x^*) + 2b_i^T x^* + c_i \le 0, \quad i = 1, 2, \ldots, m,$$

$$A_0 + \sum_{i=1}^m \lambda_i A_i \succeq 0.$$

Prove that x^* is an optimal solution of (QCQP).

Bibliography

[1] A. Beck: *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*, SIAM, 2014.

[2] J-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I ,Fundamentals*, 2^{nd} edition, Springer-Verlag, 1996.

[3] J-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms II, Advanced Theory and Bundle Methods*, 2^{nd} edition, Springer-Verlag, 1996.

[4] J. Nocedal and S.J. Wright: *Numerical Optimization*, 2^{nd} edition, Springer-Verlag, 2006.

[5] A. Ruszczynski, *Nonlinear Optimization*, Princeton University Press, 2006.

[6] H. Tuy, *Convex Analysis and Global Optimization*, Kluwer, 1998.