Note

This lecture note is mostly based on Nocedal and Wright (2006), and some contents are from Bertsekas (2003), Nesterov (2004), and Boyd and Vandenberghe (2004). Topics marked with ** are optional but recommended for interested readers. We use $\| \cdot \|$ to denote the Euclidean norm, unless otherwise specified.
Bibliography


Contents

1 Elements of Analysis and Topology 9
   1.1 Analysis ........................................ 9
   1.1.1 Sequences ................................... 9
   1.1.2 Convergence of a Sequence and a Limit ...... 9
   1.1.3 Subsequence ................................ 9
   1.1.4 Sequence with a Convergence Subsequence and a Limit Point 9
   1.1.5 Cauchy Sequence .............................. 10
   1.1.6 Boundedness ................................ 10
   1.1.7 Convergence of a Sequence (Finite Dimensions) .... 10
   1.1.8 Convergence of a Scalar Sequence .......... 10
   1.1.9 Properties of Scalar Sequences ............. 11
   1.1.10 Supremum and Infimum ...................... 11
   1.1.11 Limsup and Liminf .......................... 11
   1.2 Rate of Convergence ........................... 13
   1.2.1 Q-Rate of Convergence ....................... 13
   1.2.2 R-Rate of Convergence ....................... 14
   1.2.3 Remarks ..................................... 14
   1.3 Topology and Metric Space** ................. 15
   1.3.1 Topology .................................... 15
   1.3.2 Metric Space ................................ 15
   1.3.3 Open Sets of a Metric Space ............... 15
   1.3.4 Metric-Induced Topology ................... 15
   1.4 Topology of the Euclidean Space .............. 15
   1.4.1 Open Sets .................................. 16
   1.4.2 Closed Sets ................................ 16
   1.4.3 Interior .................................... 16
   1.4.4 Closure ..................................... 17
   1.4.5 Boundary .................................. 17
   1.4.6 Compact Sets ................................. 17
   1.4.7 Neighborhood ................................. 17
2 Continuity and Differentiability 19
   2.1 Continuous Function ........................... 19
   2.1.1 Lipschitz Continuity ....................... 19
   2.2 Derivatives .................................... 19
## CONTENTS

2.2.1 On $\mathbb{R}$ .................................................. 19
2.2.2 On $\mathbb{R}^n$ ................................................. 20

3 Unconstrained Optimization ............................................ 25
3.1 Types of Solutions ............................................. 25
3.2 Local Models of a Function .................................... 26
3.2.1 First- and Second-Order Local Model .................... 26
3.2.2 Alternative Forms of Local Models ....................... 27
3.3 Optimality: Necessary Conditions ................................ 27
3.3.1 First Order Necessary Conditions (FONC) .............. 27
3.3.2 Second Order Necessary Conditions ...................... 28
3.4 Optimality: Sufficient Conditions .............................. 29
3.4.1 Second Order Sufficient Conditions (SOSC) ............ 29
3.5 Complexity of Global Minimization ............................ 30

4 Spectral Properties of Matrices ..................................... 33
4.1 Eigenpairs ...................................................... 33
4.1.1 PageRank ................................................ 34
4.1.2 Properties of Eigenpairs ................................ 35
4.1.3 Eigendecomposition ...................................... 35
4.2 Matrix Norms .................................................. 35
4.2.1 Vector-Induced Matrix Norms (Operator Norms) ....... 35
4.2.2 Elementwise Norms ...................................... 36
4.2.3 Spectral Norms (Shatten Norms) ......................... 37
4.2.4 Norm Inequalities ....................................... 37
4.3 Positive (Semi)Definite Matrix ................................ 37
4.4 Spectral Bounds on Quadratic Forms .......................... 38
4.5 Continuity of Eigenvalues* .................................. 38

5 Unconstrained Convex Optimization ............................... 41
5.1 Basic Properties of Convex Functions .......................... 41
5.2 Properties of Convex Optimization ............................ 42
5.3 Strong Convexity .............................................. 43

6 Steepest Descent and Newton’s Method ............................. 47
6.1 (Exact) Line Search ........................................... 47
6.2 Search Direction $p_k$ ......................................... 47
6.2.1 Descent Direction ....................................... 47
6.2.2 Steepest Descent Direction ............................... 48
6.2.3 Newton Direction ....................................... 48
6.3 Inexact Line Search ........................................... 49
6.3.1 Wolfe Line Search ....................................... 49
6.3.2 Backtracking (Armijo) Line Search ..................... 51
6.3.3 Termination of Inexact Line Search ..................... 51

7 Convergence of Steepest Descent and Newton’s Methods ......... 57
7.1 Global Convergence: Descent Direction + Wolfe Line Search . . . 57
7.1.1 Condition Number ....................................... 59
7.2 Rate of Convergence ........................................... 59
7.2.1 Convergence Rate of Steepest Descent .................. 59
11.4.1 Strong Duality .................................................. 113
11.4.2 Sufficient Conditions for Strong Duality ............... 113
11.4.3 Wolfe Dual ....................................................... 114
11.5 Saddle-Point Interpretation of Duality .................... 117
11.6 Finding Primal Solutions from Dual Solutions ............ 118
Chapter 1

Elements of Analysis and Topology

This chapter presents basic mathematical concepts necessary for the lecture.

1.1 Analysis

1.1.1 Sequences

A sequence of an infinite number of points in $\mathbb{R}^n$ is denoted by $\{x_k\}_{k=1}^{\infty} = \{x_1, x_2, \ldots\} = \{x_k\}$.

1.1.2 Convergence of a Sequence and a Limit

A sequence of points $\{x_k\}$ in $\mathbb{R}^n$ converges to a point $x$, if for any $\varepsilon > 0$, there exists an index $K$ such that $\|x_k - x\|_2 \leq \varepsilon$, for all $k \geq K$.

Then we write $\lim_{k \to \infty} x_k = x$, or $x_k \to x$ as $k \to \infty$.

and call $x$ the limit of $\{x_k\}$.

Ex. $x_k = (1 - 2^{-k}, 1/k^2)^T$ converges to $(1, 0)^T$.

1.1.3 Subsequence

Given an infinite index set $S \subset \{1, 2, \ldots\}$, a subsequence of $\{x_k\}$ corresponding to $S$ can be constructed, and is denoted by $\{x_k\}_{k \in S}$.

1.1.4 Sequence with a Convergence Subsequence and a Limit Point

For a sequence $\{x_k\}$ in $\mathbb{R}^n$, if there exists an infinite set of indices $S = \{k_1, k_2, \ldots\}$ such that the corresponding subsequence $\{x_{k_i}\}_{i \in S}$ converges to a point $\hat{x} \in \mathbb{R}^n$, that is,

$$\lim_{i \to \infty} x_{k_i} = \hat{x},$$

then $\hat{x}$ is called a limit point (or, an accumulation point or a cluster point in the Euclidean space) of $\{x_k\}$. 


Alternatively, we say for any \( \epsilon > 0 \) and all positive integer \( K \), we have
\[
\|x_k - \hat{x}\| \leq \epsilon, \text{ for some } k \geq K.
\]

Ex. a sequence with two limit points \( \hat{x} = (0,0)^T \) and \( \hat{x} = (1,1)^T \):

\[
\begin{bmatrix}
1 \\
1/2 \\
1/4 \\
1/8 \\
\end{bmatrix}, \quad \begin{bmatrix}
1 \\
1/2 \\
1/4 \\
1/8 \\
\end{bmatrix}, \quad \ldots
\]

Ex. a sequence with an infinite number of limit points: \( x_k = \sin k \). Every point in the interval \([-1,1]\) is a limit point.

### 1.1.5 Cauchy Sequence

A sequence is said to be a Cauchy sequence if for any \( \epsilon > 0 \), there exists an integer \( K \) such that
\[
\|x_k - x_\ell\|_2 \leq \epsilon, \text{ for all } k \geq K \text{ and } \ell \geq K.
\]

### 1.1.6 Boundedness

A set of points in \( F \subset \mathbb{R}^n \) is **bounded** if there exists a scalar \( M > 0 \) such that
\[
\|x\|_2 \leq M, \quad \forall x \in F.
\]

### 1.1.7 Convergence of a Sequence (Finite Dimensions)

- A bounded sequence \( \{x_k\} \) in \( \mathbb{R}^n \) converges if it has a unique limit point.
- A sequence \( \{x_k\} \) in \( \mathbb{R}^n \) converges if it is a Cauchy sequence.
- *(Bolzano-Weierstrass theorem)* Every bounded sequence in \( \mathbb{R}^n \) has at least one limit point. Proof:
  - Every sequence \( \{x_k\} \) in \( \mathbb{R}^n \) has a monotone subsequence.
  - *(Monotone convergence theorem)* Every monotone sequence has a finite limit if and only if the sequence is bounded.

### 1.1.8 Convergence of a Scalar Sequence

Consider a **scalar sequence** \( \{t_k\}, t_k \in \mathbb{R} \) for all \( k \). This sequence is

- **bounded above**: if there exists a scalar \( u \) such that \( t_k \leq u \) for all \( k \).
- **bounded below**: if there exists a scalar \( \ell \) such that \( t_k \geq \ell \) for all \( k \).
- **nondecreasing**: if \( t_{k+1} \geq t_k \) for all \( k \).
- **nonincreasing**: if \( t_{k+1} \leq t_k \) for all \( k \).

The scalar sequence \( \{t_k\} \) converges if (monotone convergence theorem)

- it is nondecreasing and bounded above.
- it is nonincreasing and bounded below.

If a sequence is bounded below and bounded above, then it is called **bounded**.
1.1.9 Properties of Scalar Sequences

• \( \lim_{n \to \infty} (x_n \pm y_n) = \lim_{n \to \infty} x_n \pm \lim_{n \to \infty} y_n \).

• \( \lim_{n \to \infty} cx_n = c \lim_{n \to \infty} y_n \).

• \( \lim_{n \to \infty} \frac{x_n}{y_n} = \frac{\lim_{n \to \infty} x_n}{\lim_{n \to \infty} y_n} \text{ if } \lim_{n \to \infty} y_n \neq 0 \).

• \( \lim_{n \to \infty} x_n^p = (\lim_{n \to \infty} x_n)^p \).

• If \( x_n \leq y_n \) for all \( n \geq N \) for some positive integer \( N \), then \( \lim_{n \to \infty} x_n \leq \lim_{n \to \infty} y_n \).

• (Sandwich theorem) If \( x_n \leq z_n \leq y_n \) for all \( n \geq N \) for some positive integer \( N \) and \( \lim_{n \to \infty} x_n = \lim_{n \to \infty} y_n = \alpha \), then \( \lim_{n \to \infty} z_n = \alpha \).

• (L’Hospital theorem) For functions \( f \) and \( g \) differentiable on \( I \setminus c \) where \( I \) is an open interval containing \( c \) (\( c \) can be \( \pm \infty \)), if

\[
\lim_{x \to c} f(x) = \lim_{x \to c} g(x) = 0 \text{ or } \pm \infty \text{, and } \lim_{x \to c} \frac{f'(x)}{g'(x)} \text{ exists,}
\]

then

\[
\lim_{x \to c} \frac{f(x)}{g(x)} = \lim_{x \to c} \frac{f'(x)}{g'(x)}.
\]

1.1.10 Supremum and Infimum

For a scalar sequence \( \{t_k\} \),

• Supremum: the smallest real number \( u \): \( t_k \leq u \) for all \( k \), denoted by \( \sup \{t_k\} \).

  If there is no such number, then \( \sup \{t_k\} = \infty \). By convention, \( \sup \emptyset = -\infty \).

• Infimum: the largest real number \( \ell \): \( t_k \geq \ell \) for all \( k \), denoted by \( \inf \{t_k\} \).

  If there is no such number, then \( \inf \{t_k\} = -\infty \). By convention, \( \inf \emptyset = \infty \).

For two scalar sequences \( \{x_k\} \) and \( \{y_k\} \), we have \(^1\)

• \( \inf \{x_k\} + \inf \{y_k\} \leq \inf \{x_k + y_k\} \).

• \( \sup \{x_k\} + \sup \{y_k\} \geq \sup \{x_k + y_k\} \).

1.1.11 Limsup and Liminf

Given a sequence \( \{t_k\} \), define the sequence of suprema as \( \{u_i\} \):

\[
u_i := \sup\{t_k | k \geq i\}.
\]

Then \( \{u_i\} \) is an nonincreasing sequence. If bounded below, it converges to a finite number \( \bar{u} \), called “lim sup” of the underlying sequence \( \{t_k\} \), denoted by

\[
\limsup_{i \to \infty} \{t_k\} = \lim_{i \to \infty} u_i = \lim_{i \to \infty} \sup\{t_k | k \geq i\}
\]

---

\(^1\)Think about the proof. Hint: what is the inf or sup of a singleton set?
Similarly, define the sequence of infima as \( \{\ell_i\} \):

\[ \ell_i = \inf\{t_k | k \geq i\} \]

Then \( \{\ell_i\} \) is nondecreasing. If \( \{\ell_i\} \) is bounded above, it converges to a point \( \bar{v} \) which is called “lim inf” of \( \{t_k\} \), denoted by

\[ \lim \inf t_k = \lim_{i \to \infty} \ell_i = \lim_{i \to \infty} \inf\{t_k | k \geq i\} \]

Ex. the sequence 1, 1/2, 1, 1/4, 1, 1/8, ... has a lim inf of 0 and a lim sup of 1.

![Figure 1.1: Liminf and limsup illustrated \((y(x) = \exp(-x) \cos(2\pi x))\).](image)

Let \( \{x_k\} \) and \( \{y_k\} \) be scalar sequences. Then,

- \( \inf\{x_k\} \leq \liminf_{k \to \infty} x_k \leq \limsup_{k \to \infty} x_k \leq \sup\{x_k\} \).
- If \( \limsup_{k \to \infty} x_k \) (\( \liminf_{k \to \infty} x_k \)) is finite, then it is the largest (smallest, resp.) limit point of \( \{x_k\} \).
- \( \{x_k\} \) converges if and only if

\[ -\infty < \liminf_{k \to \infty} x_k = \limsup_{k \to \infty} x_k < \infty. \]

Furthermore, the limit of \( \{x_k\} \) is equal to the common scalar value of \( \liminf_{k \to \infty} x_k \) and \( \limsup_{k \to \infty} x_k \).

- If \( x_k \leq y_k \) for all \( k \), then

\[ \liminf_{k \to \infty} x_k \leq \liminf_{k \to \infty} y_k, \quad \limsup_{k \to \infty} x_k \leq \limsup_{k \to \infty} y_k. \]

- Finally, we have

\[ \liminf_{k \to \infty} x_k + \liminf_{k \to \infty} y_k \leq \liminf_{k \to \infty}(x_k + y_k) \]

\[ \limsup_{k \to \infty} x_k + \limsup_{k \to \infty} y_k \geq \limsup_{k \to \infty}(x_k + y_k) \]
1.2 Rate of Convergence

Let \( \{x_k\} \) be a sequence in \( \mathbb{R}^n \) that converges to \( x^* \). The sequence of interest is the one generated by an optimization algorithm. Convergence rate is an essential way to describe how fast such a sequence approaches a solution.

1.2.1 Q-Rate of Convergence

“Q” stands for “quotient”.

**Q-Linear**

The convergence is **Q-linear** if there exists a constant \( r \in (0, 1) \) such that

\[
\frac{\|x_{k+1} - x^*\|_2}{\|x_k - x^*\|_2} \leq r \quad \text{for all } k \text{ sufficiently large.}
\]

- Q-linear convergences implies that the distance to the solution \( x^* \) decreases at each iteration by a constant factor \( 0 < r < 1 \).

Ex. \( x_k = 1 + (0.5)^k \) converges to 1 Q-linearly with rate \( r = 0.5 \).

**Q-Superlinear**

The convergence is **superlinear** if

\[
\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|_2}{\|x_k - x^*\|_2} = 0.
\]

Ex. \( x_k = 1 + k^{-k} \).

(Hint)

\[ e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^n. \]

**Q-Sublinear**

The convergence is **sublinear** if

\[
\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|_2}{\|x_k - x^*\|_2} = 1.
\]

Ex. \( x_k = 1 + 1/k \).

**Q-Quadratic**

The convergence is **Q-quadratic** if there exists a constant \( M > 0 \) (not necessarily \( < 1 \)) such that

\[
\frac{\|x_{k+1} - x^*\|_2}{\|x_k - x^*\|_2^2} \leq M \quad \text{for all } k \text{ sufficiently large.}
\]

Ex. \( x_k = 1 + (0.5)^{2k} \).
1.2.2 R-Rate of Convergence

“R” stands for “root”. The convergence is R-linear if

$$\|x_k - x^*\| \leq u_k \quad \forall k \geq 1,$$

where \(\{u_k\}\) is a sequence of nonnegative scalars converges Q-linearly to zero. Other rates are defined similarly, e.g. R-superlinear is defined with \(\{u_k\}\) converges superlinearly.

R-rates are weaker than their corresponding Q-rates. For examples, let us consider a sequence \(\{x_k\}\) defined by

$$x_k = 1 + (.5)^k$$ if \(k\) even, \(x_k = 1\) otherwise.

This sequence converges R-linearly to 1, but not Q-linearly. The reason is that Q-rates require the error \(\|x_k - x^*\|\) should be nonincreasing over iterations, whilst R-rates do allow such situations.

1.2.3 Remarks

- Q-Rate of convergence rate depends on constants \(r\) and \(M\), whose values depend on algorithms and problems. But eventually a Q-quadratic sequence will always converge faster than a Q-linear sequence.

- Q-quadratic (e.g. Newton’s method nearby a solution) 
  \(\subseteq\) Q-superlinear (e.g. quasi-Newton)
  \(\subseteq\) Q-linear (e.g. gradient descent)
  \(\subseteq\) Q-sublinear (e.g. subgradient descent).
1.3 Topology and Metric Space

There are two seemingly different notions of “open sets”. Here we show how they are connected.

1.3.1 Topology

For a set \( X \), \( \mathcal{T} \subseteq 2^X \) is a topology on \( X \) if

- \( \emptyset \in \mathcal{T} \) and \( X \in \mathcal{T} \).
- Arbitrary unions of members of \( \mathcal{T} \) is in \( \mathcal{T} \).
- Finite intersections of members of \( \mathcal{T} \) is in \( \mathcal{T} \).

Ex. if \( X \neq \emptyset \), then \( 2^X \) and \( \{\emptyset, X\} \) are discrete and trivial topology, resp.

The pair \((X, \mathcal{T})\) is called a topological space. The members of \( \mathcal{T} \) are called open sets, and their complements are called closed sets.

1.3.2 Metric Space

A pair \( M = (X, d) \) with a set \( X \) and a function \( d : X \times X \to \mathbb{R} \) satisfying the following properties (\( d \) is called a metric) is a metric space:

- \( d(x, y) \geq 0 \) for all \( x, y \in X \).
- \( d(x, y) = 0 \) if and only if \( x = y \).
- \( d(x, y) = d(y, x) \) for all \( x, y \in X \).
- \( d(x, y) + d(y, z) \geq d(x, z) \) for all \( x, y, z \in X \) (triangle inequality).

The metric space \( M = (X, d) \) with a real vector space \( X = \mathbb{R}^n \) and the (Euclidean) metric \( d(x, y) = \|x - y\|_2 = (\sum_{i=1}^{n} (x_i - y_i)^2)^{1/2} \) is the \( n \)-dimensional Euclidean space. For convenience, we use \( \mathbb{R}^n \) to denote the Euclidean space.

1.3.3 Open Sets of a Metric Space

Let \((X, d)\) be a metric space. A subset \( S \subset X \) is open if for all point \( x \in S \) we can find a scalar \( \varepsilon > 0 \) such that

\[ \{ y \in X : d(x, y) < \varepsilon \} \subseteq S. \]

1.3.4 Metric-Induced Topology

The collection of all open sets in a metric space \( M = (X, d) \) constitutes a topology \( \mathcal{T} \) on \( X \) (For a proof, see Theorem 4.7 of https://www.dpmms.cam.ac.uk/~twk/Top.pdf.) A topology induced by the Euclidean metric \( d \) is called the Euclidean topology.

1.4 Topology of the Euclidean Space

Consider a constraint set \( X \subseteq \mathbb{R}^n \). An optimization algorithm typically generate a sequence \( \{x_k\} \subset X \).
1.4.1 Open Sets
A subset \( F \subset \mathbb{R}^n \) is open if for every \( x \in F \) we can find a positive number \( \varepsilon > 0 \) such that an open ball of radius \( \varepsilon \) centered at \( x \) is contained entirely in \( F \):

\[
B(x, \varepsilon) := \{ y \in \mathbb{R}^n : \|y - x\|_2 < \varepsilon \} \subseteq F.
\]

**These open sets are the members of the Euclidean topology. That is,**
- Arbitrary unions of open sets are open.
- Finite intersections of open sets are open.

Ex. (infinite intersections of open sets can be closed\(^2\))

\[
\bigcap_{n=1}^{\infty} \left(-\frac{1}{n}, \frac{1}{n}\right) = \{0\}
\]

1.4.2 Closed Sets
A set \( F \) is closed if its complement \( F^c = \mathbb{R}^n \setminus F \) is open.

Or, equivalently, if for all possible sequences of points \( \{x_k\} \) in \( F \), all limit points of \( \{x_k\} \) are elements in \( F \).

Ex.
- \( F = (0, 1) \cup (2, 10) \) is an open set of \( \mathbb{R} \).
- \( F = [0, 1] \cup [2, 5] \) is a closed set of \( \mathbb{R} \).
- \( F = (0, 1) \) is neither open nor closed.
- \( \mathbb{R}^n \) and \( \emptyset \) are both open and closed.

**From the properties of open sets, we have**
- Finite unions of closed sets are closed.
- Arbitrary intersections of closed sets are closed.

1.4.3 Interior
A point \( x \in F \subset \mathbb{R}^n \) is an interior point of the set \( F \), if there is an open ball \( B(x, \varepsilon) \) contained entirely in \( F \).

The interior of a set \( F \), denoted by \( \text{int} F \), is the largest open set contained in \( F \).

Or, equivalently, \( \text{int} F \) is the set of all interior points of \( F \).

Ex. \( F = \{ x \in \mathbb{R}^n : x_i \geq 0 \} \). \( \text{int} F = \{ x \in \mathbb{R}^n : x_i > 0 \} \).

\(^2\)Singleton sets are closed in any metric space.
1.4. Closure

The closure of a set $F$, denoted by $\text{cl} F$, is the smallest closed set containing $F$. That is,

$$x \in \text{cl} F \text{ if } \lim_{k \to \infty} x_k = x \text{ for some sequence } \{x_k\} \text{ of points in } F.$$ 

In other words, $\text{cl} F$ is the union of $F$ and all limit points from sequences in $F$.

- From definitions: $\text{int} F \subseteq F \subseteq \text{cl} F$.
- $\text{int}$ and $\text{cl}$ are idempotent: $\text{int} \text{int} F = \text{int} F$, $\text{cl} \text{cl} F = \text{cl} F$.
- $\text{int}$ and $\text{cl}$ are monotone: $X \subseteq Y$ implies that $\text{int} X \subseteq \text{int} Y$ and $\text{cl} X \subseteq \text{cl} Y$.
- $F$ is open iff $\text{int} F = F$.
- $F$ is closed iff $\text{cl} F = F$.
- If $F$ is a convex set, then $\text{int} F$ and $\text{cl} F$ are convex.

1.4.5 Boundary

The boundary of a set $F$, denoted by $\text{bd} F$, is

$$\text{bd}(F) = \text{cl} F \cap \text{cl} F^c, \quad F^c = \mathbb{R}^n \setminus F.$$ 

1.4.6 Compact Sets

A set $F$ is compact if every sequence $\{x_k\}$ of points in $F$ has at least one limit point, and all such limit points are in $F$. If $F \subset \mathbb{R}^n$ is closed and bounded, then $F$ is compact.

1.4.7 Neighborhood

For a point $x \in \mathbb{R}^n$, $N \subset \mathbb{R}^n$ is called a neighborhood of $x$, if it is an open set containing $x$. A common neighborhood is the ball centered at $x$ with radius $\epsilon > 0$, $B(x, \epsilon)$.

For a set $F \subset \mathbb{R}^n$, $N$ is a neighborhood of $F$ if $N$ is open and there exists $\epsilon > 0$ such that

$$\bigcup_{x \in F} B(x, \epsilon) \subseteq N.$$
CHAPTER 1. ELEMENTS OF ANALYSIS AND TOPOLOGY

Exercises

1.1. Consider an optimization problem

$$\min_{x \in X} f(x) \text{ subject to } x \in X$$

where $X \subseteq \mathbb{R}^n$ is a closed set, and consider designing an algorithm that generates a sequence of points, say $\{x_k\}$, where each $x_k \in X$. Your algorithm tries to find an optimal solution $x^* \in X$ eventually. However, it is more practical to find an $x_k$ that is close enough to $x^*$. Let’s assume that we can compute how good an $x_k$ is, e.g. if $\|x_k - x^*\|_2 \leq \varepsilon$ (although you do not know $x^*$), to stop the algorithm.

(a) Will it be enough that the sequence $\{x_k\}$ has a limit point which is close to $x^*$? Or is it necessary to make sure your algorithm to generate a convergent sequence, with a limit nearby $x^*$?

(b) What could happen if $X$ is not closed?

(c) Suppose that $X \subset \mathbb{R}^n$ but $X \neq \mathbb{R}^n$. Should $X$ be bounded to be able to find a minimizer of the above problem? Consider two cases, when (i) $f$ is convex, and (ii) $f$ is concave. ($f$ is concave if $-f$ is convex, see the definition in the next problem).

(d) A simple stopping criterion for optimization algorithms is to check if

$$\frac{\|x_{k+1} - x_k\|_2}{\max\{1, \|x_k\|_2\}} \leq \varepsilon$$

for some small $\varepsilon > 0$. Suppose that your algorithm use this. Does your answer to the problem (a) still hold?

1.2. Let $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, \ldots, m$ be convex functions. Show that

(a) $F(x) = \max\{f_1(x), f_2(x), \ldots, f_m(x)\}$,

(b) $F(x) = \sum_{i=1}^m f_i(x)$

are also convex functions. Use the definition that a function $F(x)$ is convex on a set $X \subseteq \mathbb{R}^n$ if

$$F(ax + (1-a)y) \leq aF(x) + (1-a)F(y)$$

is true for all $a \in [0, 1]$ and for all $x, y \in X$.

1.3. Let $\{x_k\}$ and $\{y_k\}$ be two scalar sequences. Show that the following arguments are true, using the definitions of inf and lim inf:

(a) If $c \leq x_k$ for all $k$ for a scalar $c$, then $c \leq \inf\{x_k\}$.

(b) $\inf\{x_k\} + \inf\{y_k\} \leq \inf\{x_k + y_k\}$.

(c) $\lim\inf\{x_k\} + \lim\inf\{y_k\} \leq \lim\inf\{x_k + y_k\}$.

1.4. Let $X$ and $Y$ be two closed sets in $\mathbb{R}^n$. Give an example of $X$ and $Y$ such that their sum $X + Y := \{x + y : x \in X, y \in Y\}$ is not a closed set. Choose any dimension $n$ you like. (Hint: try $n = 2$.)

1.5. Show that if a set $C$ is convex, then its closure $\text{cl} C$ is also convex. Use the definition that a set $C$ is convex if

$$ax + (1-a)y \in C$$

for any $x, y \in C$ and for any $a \in [0, 1]$. 
Chapter 2

Continuity and Differentiability

2.1 Continuous Function

A function $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuous at $x_0 \in \text{cl } D$ if

$$\lim_{x \rightarrow x_0} f(x) = f(x_0),$$

or, equivalently, if for all $\epsilon > 0$ there exists a value $\delta > 0$ such that

$$\|x - x_0\|_2 < \delta \text{ for } x \in D \Rightarrow \|f(x) - f(x_0)\|_2 < \epsilon.$$

A function $f$ is continuous on its domain $D$ if $f$ is continuous for all $x_0 \in \text{cl } D$ (note: $D$ will be a closed set for most functions we discuss).

A Special Case ($n = 1$) A function $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ is continuous at $x_0 \in \text{cl } D$ if

$$\lim_{x \uparrow x_0} f(x) = \lim_{x \downarrow x_0} f(x) = f(x_0).$$

For $x_0$ at the boundary of $D$, check either the limit from below or from above.

2.1.1 Lipschitz Continuity

A function $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$ is Lipschitz continuous on some set $N \subseteq D$ if there exists a constant $L > 0$ such that

$$\|f(x) - f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in N.$$

If it holds for a neighborhood $N$ of $\bar{x} \in \text{int } D$, $N \subseteq D$, then $f$ is called locally Lipschitz continuous.

2.2 Derivatives

2.2.1 On $\mathbb{R}$

Let $\phi : \mathbb{R} \rightarrow \mathbb{R}$ be a real-valued function of a real variable.
\{1st derivative\} \[ \frac{d\phi}{d\alpha} = \phi'(\alpha) := \lim_{\varepsilon \to 0} \frac{\phi(\alpha + \varepsilon) - \phi(\alpha)}{\varepsilon} \]

\{2nd derivative\} \[ \frac{d^2\phi}{d\alpha^2} = \phi''(\alpha) := \lim_{\varepsilon \to 0} \frac{\phi'(\alpha + \varepsilon) - \phi'(\alpha)}{\varepsilon} \]

Chain rule: Suppose that \(\alpha\) is a function of \(\beta\) (we write \(\alpha = \alpha(\beta)\)). Then the derivative of \(\phi\) w.r.t. \(\beta\) is
\[ \frac{d\phi(\alpha(\beta))}{d\beta} = \frac{d\phi}{d\alpha} \frac{d\alpha}{d\beta} = \phi'(\alpha)\alpha'(\beta). \]

2.2.2 On \(\mathbb{R}^n\)
Let \(f : \mathbb{R}^n \to \mathbb{R}\), and \(x = (x_1, x_2, \ldots, x_n)^T\) is a column vector.

Frechet differentiability \(f\) is differentiable at \(x\) if there exists a vector \(g \in \mathbb{R}^n\) such that
\[ \lim_{h \to 0} \frac{f(x + h) - f(x) - g^T h}{\|h\|} = 0, \] (2.1)
where \(\|\cdot\|\) is any vector norm of \(h\). The vector \(g\) is the Frechet derivative of \(f\) at \(x\).

Directional derivative A related notion is the directional derivative, which is defined for a given vector \(d \in \mathbb{R}^n\) such that
\[ \nabla_d f(x) = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon d) - f(x)}{\varepsilon}, \]
which is the same as Frechet's definition when one chooses \(h = \varepsilon d\).

Gradient We call the vector of partial derivatives as the gradient of \(f\) at \(x\), denoted by \(\nabla f(x)\):
\[ \nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \]
The partial derivatives \(\frac{\partial f}{\partial x_i}\) are directional derivatives w.r.t. standard axes, obtained with \(h = \varepsilon e_i\) for \(\varepsilon > 0\) and the \(i\)th unit vector \(e_i\) (which has 1 at the \(i\)th element and 0 at the rest). In other words,
\[ \frac{\partial f}{\partial x_i} = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon e_i) - f(x)}{\varepsilon}. \]
For a function \(f(x, y)\) of two vector variables \(x\) and \(y\), we use \(\nabla_y f(x, y)\) to denote the partial gradient w.r.t. a specific variable \(y\), so that
\[ \nabla f(x, y) = \begin{bmatrix} \nabla_x f(x, y) \\ \nabla_y f(x, y) \end{bmatrix}. \]
2.2. DERIVATIVES

Jacobian matrix  When \( f : \mathbb{R}^n \to \mathbb{R}^m \), then \( \nabla f(x) \) is an \( n \times m \) matrix: \( i \)th column is \( \nabla f_i(x) \). The Jacobian is \( J(x) = (\nabla f(x))^T \) with dimensions \( m \times n \).

Ex. \( f(x) = Ax \) for \( A \in \mathbb{R}^{k \times n} \). What is \( \nabla f(x) \)? What is \( J(x) \)?

Hessian  The matrix of second partial derivatives of \( f \) is called the Hessian:

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

Ex. \( f(x_1, x_2) = x_1^2 + x_1 x_2 \). Then,

\[
\nabla f(x_1, x_2) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 \end{bmatrix} \quad \nabla^2 f(x_1, x_2) = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}.
\]

Ex. \( f(x) = x^T H x \) where \( H \) is a symmetric matrix. \( \nabla f(x) \) and \( \nabla^2 f(x) \)? Suppose that \( x \in \mathbb{R}^n \) and \( H \in \mathbb{R}^{n \times n} \). Note that

\[
f(x) = x^T H x = \sum_{i=1}^n \sum_{j=1}^n x_i x_j H_{ij} = \sum_{i=1}^n x_i^2 H_{ii} + \sum_{i \neq j} x_i x_j H_{ij}
\]

Therefore,

\[
[\nabla f(x)]_k = \frac{\partial f(x)}{\partial x_k} = 2x_k H_{kk} + \sum_{j \neq k} x_j H_{kj} + \sum_{i \neq k} x_i H_{ik} = 2x_k H_{kk} + 2 \sum_{j \neq k} x_j H_{kj} \quad (H \text{ is symmetric})
\]

\[
= 2 \sum_{j=1}^n x_j H_{kj} = 2H_{k} x
\]

That is,

\[
\nabla f(x) = 2H x
\]

Check that \( \nabla^2 f(x) = 2H \).

\( f \) is called differentiable on a domain \( D \) if \( \nabla f(x) \) exists for all \( x \in D \). \( f \) is called twice differentiable on a domain \( D \) if \( \nabla^2 f(x) \) exists for all \( x \in D \).

Classes of Continuously Differentiable Functions  \( f \) is \( k \)-times continuously differentiable if \( \nabla^k f(x) \) is a continuous function of \( x \). The set of such functions is denoted as \( C^k \).

Note that if \( f \in C^2 \), twice continuously differentiable, then its Hessian is a symmetric matrix, since

\[
\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}.
\]
Chapter 2. Continuity and Differentiability

Chain Rule  Consider \( f : \mathbb{R}^k \rightarrow \mathbb{R}, \ x : \mathbb{R}^n \rightarrow \mathbb{R}^k \), and a composite function \( h(t) = f(x(t)) \). The chain rule provides \( \nabla h(t) \) as follows,

\[
\nabla h(t) = \sum_{i=1}^{k} \frac{\partial f}{\partial x_i} \nabla x_i(t) = \nabla x(t) \nabla f(x(t)) = J(x(t))^T \nabla f(x(t)).
\]

Note that \( J(x(t)) \in \mathbb{R}^{k \times n} \) and \( \nabla f(x(t)) \in \mathbb{R}^k \), so that \( \nabla h(t) \in \mathbb{R}^n \).

Ex. \( f(x_1, x_2) = x_1^2 + x_1 x_2 \) where \( x_1(t_1, t_2) = \sin t_1 + t_2^2 \) and \( x_2(t_1, t_2) = (t_1 + t_2)^2 \). What is \( \nabla h(t) \) for \( h(t) := f(x(t)) \)?

\[
\nabla h(t) = \sum_{i=1}^{2} \frac{\partial f}{\partial x_i} \nabla x_i(t)
\]

\[
= (2x_1 + x_2) \begin{bmatrix} \cos t_1 \\ 2t_2 \end{bmatrix} + x_1 \begin{bmatrix} 2(t_1 + t_2) \\ 2(t_1 + t_2) \end{bmatrix}
\]

\[
= \begin{bmatrix} 2(\sin t_1 + t_2^2) + (t_1 + t_2)^2 \cos t_1 \\ 2t_2 \end{bmatrix} \begin{bmatrix} \cos t_1 \\ 2(t_1 + t_2) \end{bmatrix} + \begin{bmatrix} \sin t_1 + t_2^2 \\ 2(t_1 + t_2) \end{bmatrix} \begin{bmatrix} 2(t_1 + t_2) \\ 2(t_1 + t_2) \end{bmatrix}
\]

Ex. \( x(t) = At \in \mathbb{R}^k \) for \( A \in \mathbb{R}^{k \times n} \) and \( t \in \mathbb{R}^n \), \( f(x) = ||x||^2_2 = x^T x \) and \( h(t) := f(x(t)) \).

( Method 1 ) \( \nabla x(t) = A^T \Rightarrow \nabla h(t) = 2A^T At \)

( Method 2 ) \( h(t) = (At)^T (At) = t^TA^T At \Rightarrow \nabla h(t) = 2A^T At \)
2.2. DERIVATIVES

Exercises

2.1. Consider two differentiable functions \( f_1 : \mathbb{R}^n \to \mathbb{R} \) and \( f_2 : \mathbb{R}^n \to \mathbb{R} \).

(a) Suppose that \( \nabla f_1 \) and \( \nabla f_2 \) are Lipschitz continuous, with constants \( L_1 > 0 \) and \( L_2 \), respectively. Show that \( \nabla f_1 + \nabla f_2 \) is Lipschitz continuous (also the constant).

(b) Suppose that \( f_1 \) and \( f_2 \) are Lipschitz continuous, with constants \( L'_1 > 0 \) and \( L'_2 > 0 \), respectively, and also that both are bounded in \( \mathbb{R}^n \), such that there exists \( M_1 > 0 \) with \( |f_1(x)| \leq M_1 \) and \( M_2 > 0 \) with \( f_2(x) \leq M_2 \) for all \( x \in \mathbb{R}^n \). Show that the function \( f_1(\cdot) f_2(\cdot) \) is Lipschitz continuous (also the constant).

2.2. For \( f(x) = \frac{1}{2} \|y - Ax\|^2 \) where \( x \in \mathbb{R}^n \), \( y \in \mathbb{R}^k \), and \( A \in \mathbb{R}^{k \times n} \), derive the expressions of \( \nabla f(x) \) and \( \nabla^2 f(x) \) using the chain rule.

2.3. (Moved to another chapter) Compute the gradient \( \nabla f(x) \) and the Hessian \( \nabla^2 f(x) \) of the Rosenbrock function,

\[
f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.
\]

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

2.4. (Moved to another chapter) Consider \( f(x) = \frac{1}{2} x^T H x \) where \( H \in \mathbb{R}^{n \times n} \) is a symmetric positive semidefinite matrix. Show that \( f(x) \) is convex on the domain \( \mathbb{R}^n \).

2.5. Consider \( f(x) = \frac{1}{2} x^T H x \), where \( x \in \mathbb{R}^n \) and \( H \in \mathbb{R}^{n \times n} \). From definitions, it is clear that \( \nabla^2 f(x) = H \) when \( H \) is symmetric. Derive the expression of \( \nabla^2 f(x) \) when \( H \) is not symmetric.

2.6. ** A function \( h : \mathbb{R}^n \to \mathbb{R} \) is called lower semi-continuous (l.s.c.) at \( x_0 \) when

\[
h(x_0) \leq \liminf_{x \to x_0} h(x),
\]

and upper semi-continuous (u.s.c.) at \( x_0 \) if

\[
h(x_0) \geq \limsup_{x \to x_0} h(x).
\]

Consider an unconstrained optimization problem

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

where \( f : \mathbb{R}^n \to \mathbb{R} \). Suppose that \( f \) is l.s.c. but not continuous at a minimizer \( x^* \). Would it be necessary \( f \) being continuous at \( x^* \), for an optimization algorithm to find \( x^* \)?
Chapter 3

Unconstrained Optimization

From this chapter, we begin our discussion about solving unconstrained optimization problems,

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

where $f : \mathbb{R}^{n} \to \mathbb{R}$ is an objective function.

3.1 Types of Solutions

Depending on the shape of the function, we may have one or more solutions with different characteristics. In some applications, it might be critical to distinguish one from the others.

Global Minimizer

A vector $x^* \in \mathbb{R}^{n}$ is a global minimizer if $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^{n}$.

Remarks:

- Finding a global minimizer can be a very difficult task in many cases, especially when the objective $f$ is not a convex function.
- There could be more than one global minimizer (examples?).

(Weak) Local Minimizer

$x^*$ is a local minimizer if $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}(x^*)$, where $\mathcal{N}(x^*)$ is a neighborhood of $x^*$.

Strict/Strong Local Minimizer

$x^*$ is a strict/strong local minimizer if $f(x^*) < f(x)$ for all $x \in \mathcal{N}(x^*)$ such that $x \neq x^*$.

Isolated Local Minimizer

$x^*$ is an isolated local minimizer if there exists a neighborhood $\mathcal{N}(x^*)$ of $x^*$ such that $x^*$ is the only local minimizer in $\mathcal{N}(x^*)$. 
CHAPTER 3. UNCONSTRAINED OPTIMIZATION

Figure 3.1: The function defined as $f(x) = x^4 \cos(1/x) + 2x^4 \forall x \neq 0$ and $f(0) = 0$ has a strict local minimizer $x^* = 0$, which is not isolated.

Remark:

- All isolated local minimizers are strict.
- Vice versa may not be true: see Figure 3.1.

Note that some real-world functions can be as nasty as the example in the above figure, having millions of local minima.

3.2 Local Models of a Function

In an optimization algorithm, it is crucial to know when to stop. Typically, we check if a point $x_k$ generated by the algorithm is a local minimizer. Considering the definition of local minimizers, it may require us to check if $f(x_k) \leq f(x)$ for all $x$ in a neighborhood of $x_k$, which will be very costly.

When $f$ is smooth (i.e., at least $C^1$), then there are much more efficient ways of characterizing local minimizers using the first-order $\nabla f(x_k)$ and/or the second-order information $\nabla^2 f(x_k)$.

3.2.1 First- and Second-Order Local Model

To study the optimality of a point $x$, we use a local model of $f(x)$ that behaves similarly to $f(x)$ at the vicinity of $x$. Taylor’s theorem gives such a model built with $\nabla f(x)$ and $\nabla^2 f(x)$.

Theorem 3.1 (Taylor’s theorem). Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is in $C^1$ and that $p \in \mathbb{R}^n$. Then there exists $t \in (0, 1)$ such that

$$f(x + p) = f(x) + \nabla f(x + tp)^T p.$$
If $f \in \mathcal{C}^2$, there exists $t \in (0, 1)$ such that

$$\nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 f(x + tp) p \, dt,$$

which gives us a second-order model,

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

Remark: Note that the above models are exact.

### 3.2.2 Alternative Forms of Local Models

Sometimes it is more convenient to consider alternative forms of the above results:

$$f(x + p) = f(x) + \nabla f(x)^T p + o(\|p\|_2)$$

or, replacing $x = x^*$ and $x^* + p = x$ (so that $p = x - x^*$), we have

$$f(x) = f(x^*) + \nabla f(x^*)^T(x - x^*) + o(\|x - x^*\|_2)$$

$$f(x) = f(x^*) + \nabla f(x^*)^T(x - x^*) + \frac{1}{2} (x - x^*)^T \nabla^2 f(x^*)(x - x^*) + o(\|x - x^*\|_2^2).$$

Clearly, we're interested in the case $x \to x^*$, i.e. $p \to 0$.

Comparing to the models in Theorem 3.1, the linear term (first-order) and the quadratic term (second-order) are "inexact": there is no scalar $t$ to make these terms exact. Still, errors are represented with the little-o notation.

**Order notation** Consider a function $h : \mathbb{R} \to \mathbb{R}$.

- Little-o: $h(v) = o(v)$ if $\frac{h(v)}{v} \to 0$, as $v \to 0$ or $v \to \infty$ (which one? It is often clear from their context).

- Big-o: $h(v) = O(v)$ if there is a constant $C > 0$ such that $|h(v)| \leq C|v|$ for all $v \in \mathbb{R}$.

Note that these notations are different from the ones used in computational theory. The O-notations here can be used within equations, inequalities, etc.

### 3.3 Optimality: Necessary Conditions

#### 3.3.1 First Order Necessary Conditions (FONC)

Let us begin with discussing the idea roughly. Consider a first-order approximation of $f \in \mathcal{C}^1$ near $x^*$:

$$f(x^* + \Delta x) - f(x^*) \approx \nabla f(x^*)^T \Delta x$$
CHAPTER 3. UNCONSTRAINED OPTIMIZATION

If \( x^* \) is a local minimizer, then the first order cost due to a small variation \( \Delta x \) is expected to be nonnegative:

\[
\nabla f(x^*)^T \Delta x \geq 0.
\]

Replacing \( \Delta x \) by \( -\Delta x \), we also get \( \nabla f(x^*)^T \Delta x \leq 0 \). The two inequalities imply that \( \nabla f(x^*)^T \Delta x = 0 \) for all \( \Delta x \), which in turn implies that

\[
\nabla f(x^*) = 0.
\]

This condition was first formulated by Fermat in 1637.

Let us discuss this more formally.

**Theorem 3.2 (FONC).** If \( x^* \) is a local (unconstrained) minimizer and \( f \) is continuously differentiable in an open neighborhood of \( x^* \), then \( \nabla f(x^*) = 0 \).

**Proof.** Suppose that \( x^* \) is a local minimizer in a neighborhood \( \mathcal{N} \). Choose a vector \( s \in \mathbb{R}^n \), \( s \neq 0 \), and a small constant \( \alpha > 0 \) such that \( x^* + \alpha s \in \mathcal{N} \), and consider \( g(\alpha) := f(x^* + \alpha s) \). Using the chain rule for differentiation,

\[
\frac{dg(0)}{d\alpha} = s^T \nabla f(x^*).
\]

Also, from the definition of differentiation,

\[
\frac{dg(0)}{d\alpha} = \lim_{\alpha \to 0} \frac{g(\alpha) - g(0)}{\alpha} = \lim_{\alpha \to 0} \frac{f(x^* + \alpha s) - f(x^*)}{\alpha} \geq 0,
\]

where the inequality in the end is from the fact that \( x^* \) is a local minimizer in \( \mathcal{N} \). The two results above imply that

\[
s^T \nabla f(x^*) \geq 0.
\]

Since the choice of \( s \) is arbitrary, the same inequality holds with \( s \) replaced by \( -s \), leading us to \( s^T \nabla f(x^*) \leq 0 \). Therefore, \( s^T \nabla f(x^*) = 0 \) for all nonzero vectors \( s \in \mathbb{R}^n \), which implies that \( \nabla f(x^*) = 0 \).

**Stationary point** A point \( x^* \) is called a stationary point if \( \nabla f(x^*) = 0 \). Any local minimizer is a stationary point.

### 3.3.2 Second Order Necessary Conditions

Consider a second-order approximation of \( f \) near \( x^* \):

\[
f(x^* + \Delta x) - f(x^*) \approx \nabla f(x^*)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x^*) \Delta x.
\]

If \( x^* \) is a local minimizer and the variation \( \Delta x \) is small, then the quantity on the right-hand side will be nonnegative. Since \( \nabla f(x^*)^T \Delta x = 0 \) from FONC, we have

\[
\Delta x^T \nabla^2 f(x^*) \Delta x \geq 0.
\]

This implies that \( \nabla^2 f(x^*) \) is positive semidefinite.
**Theorem 3.3** (SONC). If $x^*$ is an unconstrained local minimiser of $f$, and $\nabla^2 f$ exists and is continuous in an open neighborhood of $x^*$, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semidefinite.

**Proof.** Since $f$ is twice continuously differentiable near $x^*$, the second order Taylor series expansion yields for a scalar $\alpha > 0$ and a vector $d = d'/||d'||_2 \in \mathbb{R}^n$ (for an arbitrary vector $d' \in \mathbb{R}^n$, $d' \neq 0$, so that $||d||_2 = 1$),

$$f(x^* + ad) - f(x^*) = \alpha \nabla f(x^*)^T d + \frac{\alpha^2}{2} d^T \nabla^2 f(x^*) d + o(\alpha^2)$$

Using the condition $\nabla f(x^*) = 0$ from the FONC (Theorem 3.2), dividing both sides by $\alpha^2$, and using the fact that $x^*$ is a local minimizer, we obtain

$$0 \leq \frac{f(x^* + ad) - f(x^*)}{\alpha^2} = \frac{1}{2} d^T \nabla^2 f(x^*) d + \frac{o(\alpha^2)}{\alpha^2}.$$  

Taking the limit as $\alpha \to 0$ yields $d^T \nabla^2 f(x^*) d \geq 0$, which implies that $\nabla^2 f(x^*)$ is positive semidefinite. (Normalization of $d'$ does not limit the generality of the proof, since the same inequality holds with replacing $d$ by $d' = ||d'||_2 d$.)

### 3.4 Optimality: Sufficient Conditions

Unfortunately, there is no first-order sufficient condition of optimality in general unconstrained optimization. The situation changes, however, when the objective function is convex, which we will discuss in the next chapter.

#### 3.4.1 Second Order Sufficient Conditions (SOSC)

**Theorem 3.4** (SOSC). Suppose that $\nabla^2 f$ is continuous in an open neighborhood of $x^*$ and that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then $x^*$ is a strict local minimiser of $f$.

**Proof.** Since the Hessian $\nabla^2 f$ is continuous and positive definite at $x^*$, we can choose a small radius $r > 0$ so that $\nabla^2 f(x)$ remains positive definite for all $x \in B(x^*, r) := \{z : ||z - x^*||_2 < r\}$. Consider a nonzero vector $p$ with $||p||_2 < r$, so that $x^* + p \in B(x^*, r)$. Taylor’s theorem gives

$$f(x^* + p) - f(x^*) = \nabla f(x^*)^T p + \frac{1}{2} p^T \nabla^2 f(z) p = \frac{1}{2} p^T \nabla^2 f(z) p$$

where $z := x^* + tp$ for some $t \in (0, 1)$. Since $z \in B(x^*, r)$, we have $p^T \nabla^2 f(z) p > 0$. Therefore $f(x^* + p) > f(x^*)$, showing the claim.

**Proof.** *(alternative)* Let $\lambda_1$ and $\lambda_n$ be the smallest and the largest eigenvalues of a positive semidefinite matrix $H \in \mathbb{R}^{n \times n}$, resp. Then $\lambda_1 ||x||_2^2 \leq x^T H x \leq \lambda_n ||x||_2^2$ for all $x \in \mathbb{R}^n$.

---

1 this implies that its eigenvalues are continuous as well.
Let $\lambda$ be the smallest eigenvalue of $\nabla^2 f(x^*)$ ($\lambda > 0$ since $\nabla^2 f(x^*)$ is positive definite). Using the second order Taylor series expansion for a nonzero vector $d$ (with small enough norm $\|d\|_2$ so that $f$ is $C^2$ at $x + d$), we have

$$f(x^* + d) - f(x^*) = \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + o(\|d\|_2^2)$$

$$\geq \frac{\lambda}{2} \|d\|_2^2 + o(\|d\|_2^2) = \left( \frac{\lambda}{2} + \frac{o(\|d\|_2^2)}{\|d\|_2^2} \right) \|d\|_2^2 > 0.$$ 

Regarding the last inequality, consider a scalar sequence $\{v_{\|d\|_2}\}$ indexed by vector norms $\|d\|_2$ such that $v_{\|d\|_2} = \frac{o(\|d\|_2^2)}{\|d\|_2^2}$. From the definition of the little-o, we have $\lim_{\|d\|_2 \to 0} v_{\|d\|_2} = 0$. That is, there exists $\delta > 0$ for which $0 < \|d\|_2 < \delta$ and $|v_d| = \frac{o(\|d\|_2^2)}{\|d\|_2^2} < \frac{\lambda}{2}$. 

\[\square\]

### 3.5 Complexity of Global Minimization

Consider the minimization of a function $f : \mathbb{R}^n \to \mathbb{R}$ on an $n$-dimensional box,

$$f^* = \min_{x \in \mathbb{R}^n} f(x) \text{ subject to } x \in B_n := \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1, \ i = 1, \ldots, n\}.$$ 

Suppose that the function $f$ is Lipschitz continuous on $B_n$,

$$|f(x) - f(y)| \leq L \|x - y\|_2, \ \forall x, y \in B_n,$$

for a constant $L > 0$.

We try to find a global minimizer using a uniform grid search method, which is described as follows,

1. Form a uniform grid of $(s + 1)^n$ points for an integer $s > 0$,

$$x_{i_1, i_2, \ldots, i_n} = \left( \frac{i_1}{s}, \ldots, \frac{i_n}{s} \right),$$

for all $i_1 = 0, \ldots, s$, $i_2 = 0, \ldots, s$, $\ldots$, $i_n = 0, \ldots, s$.

2. Evaluate the objective $f$ on all points, and find the point $\hat{x}$ with the minimal value.

3. Return $(\hat{x}, f(\hat{x}))$.

**Theorem 3.5.** For the uniform grid search problem above, we have

$$f(\hat{x}) - f^* \leq L \frac{\sqrt{n}}{2^s}.$$
Proof. Let us denote by $x^*$ the global minimizer of the problem. There exists a grid point satisfying

$$x := x_{(i_1, i_2, \ldots, i_n)} \leq x^* \leq x_{(i_1+1, i_2+1, \ldots, i_n+1)} =: y.$$

Then $y_i - x_i = 1/s$ and $x_i^* \in [x_i, y_i]$, for all $i$. Consider a grid point $\tilde{x}$ as a “rounding” of $x^*$, constructed as follows,

$$\tilde{x}_i = \begin{cases} y_i & \text{if } x_i^* \geq (x_i + y_i)/2 \\ x_i & \text{o.w.} \end{cases}$$

Then $|\tilde{x}_i - x_i^*| \leq 1/(2s)$ for all $i$, and

$$\|\tilde{x} - x^*\|^2 = \sum_{i=1}^{n} (\tilde{x}_i - x_i^*)^2 \leq \frac{n}{4s^2}.$$

Since $\tilde{x}$ is a grid point, it implies with the Lipschitz continuity that

$$f(\tilde{x}) - f(x^*) \leq f(x) - f(x^*) \leq L\|\tilde{x} - x^*\| \leq \frac{\sqrt{n}}{2s}.$$

According to the above result, in order to find a point $\hat{x}$ which is $\epsilon$-suboptimal in terms of the objective function value, i.e., $f(\hat{x}) - f^* \leq \epsilon$, we need to make $s$ as large as

$$L\frac{\sqrt{n}}{2s} \leq \epsilon \Rightarrow s \geq \left\lceil L\frac{\sqrt{n}}{2\epsilon} \right\rceil.$$

And therefore the number of grid points required is,

$$\left(\left\lceil L\frac{\sqrt{n}}{2\epsilon} \right\rceil + 1\right)^n.$$

This is also the complexity of the uniform grid search method described above to find an $\epsilon$-suboptimal point to the global minimizer. It grows very fast, in terms of the problem dimension $n$ and optimality $\epsilon$. 
Exercises

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

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

Plot the contour of the function, using a software of your choice (e.g. use the “contour” function in Matlab). Show that $x^* = (1,1)^T$ is the only local minimizer of this function, and that the Hessian matrix at $x^*$ is positive definite.

3.2. Consider $f(x) = \frac{1}{2}x^T H x$ where $H \in \mathbb{R}^{n \times n}$ is a symmetric positive semidefinite matrix. Show that $f(x)$ is a convex function on the domain $\mathbb{R}^n$.

3.3. Show that the set of global minimizers of a convex function $f$ is a convex set.
Chapter 4

Spectral Properties of Matrices

This chapter contains basic properties of matrices and its eigen-spectrum. For optimization, understanding the smallest and the largest eigenvalues of the Hessian matrices are very important.

4.1 Eigenpairs

A pair \((v, \lambda)\) for a nonzero vector \(v \in \mathbb{R}^n\) and a scalar \(\lambda \in \mathbb{C}\) is called an eigenpair for a matrix \(A \in \mathbb{R}^{n \times n}\) if

\[Av = \lambda v.\]

\(v\) is called an eigenvector and \(\lambda\) is an eigenvalue.

Remarks:

- A square matrix \(A \in \mathbb{R}^{n \times n}\) can have at most \(n\) different eigenvalues. The collection of all eigenvalues is called as the eigenspectrum of \(A\).
- Different eigenvectors may exist for the same eigenvalue.
- We can assume that eigenvectors are normalized without loss of generality. That is, for an eigenpair \((v, \lambda)\), \((v/\|v\|_2, \lambda)\) is also an eigenpair.
- For an eigenvalue \(\lambda\) of \(A\), the following is true:

\[\lambda\] is an eigenvalue of \(A \iff A - \lambda I\) is singular \(\iff \det(A - \lambda I) = 0\]

where \(I\) is an \(n \times n\) identity matrix.

The LHS of the last equation defines a characteristic polynomial of \(A\).

Ex. \(A = \begin{bmatrix} 4 & 1 \\ 1 & -2 \end{bmatrix}\)

\[\det(A - \lambda I) = \det \begin{bmatrix} 4 - \lambda & 1 \\ 1 & -2 - \lambda \end{bmatrix} = (4 - \lambda)(-2 - \lambda) - 1 \cdot 1 = 0\]

\[\Rightarrow \lambda^2 - 2\lambda - 9 = (\lambda - 1)^2 - 10 = 0 \Rightarrow \lambda = 1 \pm \sqrt{10}\]

In \(\mathbb{R}\),
3.4 \ CHAPTER 4. SPECTRAL PROPERTIES OF MATRICES

> A = matrix(c(4,1,1,-2), nrow=2, ncol=2, byrow=TRUE)
> eigen(A)
$\text{values}$
[1] 4.162278 -2.162278

$\text{vectors}$
[,1] [,2]
[1,] -0.9870875 0.1601822
[2,] -0.1601822 -0.9870875

4.1.1 \ PageRank

Understanding eigenpairs is an important problem in many different fields, including signal processing, physics, statistics, machine learning, and optimization, just to mention a few. Here we show an example.

Consider $n$ webpages connected with outlinks / inlinks, as illustrated in Figure 4.1 for four webpages. We want to estimate the “reputation” or the “rank” of each webpage via the following formulation:

$$r_i = \sum_{j \in \text{inlink}(i)} \frac{r_j}{|\text{outlink}(j)|}, \ i = 1, \ldots, n.$$ 

Now the question is that if there exists a vector $r = (r_1, \ldots, r_n)$ satisfying the above set of equations.

Consider a matrix $A$ defined as follows:

$$A_{ij} = \begin{cases} 1/|\text{outlink}(j)| & \text{if } j \rightarrow i \\ 0 & \text{o.w.} \end{cases}. \quad (4.1)$$

The matrix $A$ corresponding to the example above is

$$A = \begin{bmatrix} 0 & 0 & 1/2 & 1 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}.$$ 

Using $A$, we can rewrite the rank equations as follows,

$$r_i = A_i r, \ i = 1, \ldots, n, \ \text{ or equivalently, } r = Ar.$$ 

How is this problem related to finding eigenpair problems?

![Figure 4.1: Web pages with in-/out-links.](image)

4.1.2 Properties of Eigenpairs

Suppose that \( \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \) is the set of all eigenvalues of \( A \in \mathbb{R}^{n \times n} \).

- If \( A \) is symmetric, then all of its eigenvalues are real-valued.
- If \( A \) is a triangular matrix, the eigenvalues are equal to the diagonal entries of \( A \).
- The eigenvalues of \( A + cI \), \( c \in \mathbb{R} \), are equal to \( \lambda_1 + c, \lambda_2 + c, \ldots, \lambda_n + c \).
- The eigenvalues of a square matrix \( A \) depends continuously on the elements of \( A \) (see below for a proof).
- \( A^2 v = A(Av) = A(\lambda v) = \lambda^2 v \). In general, \( A^k v = \lambda^k v \).
- \( A \) is invertible iff all eigenvalues of \( A \) are nonzero. If \( A \) is invertible and \( (v, \lambda) \) is an eigenpair of \( A \), then \( (v, \lambda^{-1}) \) is an eigenpair of \( A^{-1} \).
- The trace of \( A \): \( \text{tr}(A) := \sum_i A_{ii} = \sum_i \lambda_i \).
- The determinant of \( A \): \( \text{det}(A) = \prod_i \lambda_i \).

4.1.3 Eigendecomposition

If a matrix \( A \in \mathbb{R}^{n \times n} \) has \( n \) linearly independent eigenvectors \( q_i, i = 1, \ldots, n \), then we can write

\[
A = Q \Lambda Q^{-1}
\]

where \( Q = [q_1, q_2 \ldots q_n] \in \mathbb{R}^{n \times n} \) consists of eigenvectors as columns, and \( \Lambda \) is a diagonal matrix with the corresponding eigenvalues. If \( A \) can be written in this way, we also say \( A \) is diagonalizable.

If \( A \) is symmetric, then the decomposition becomes

\[
A = Q \Lambda Q^T,
\]

where the columns of \( Q \) comprise an orthonormal basis of eigenvectors, i.e., \( Q^T Q = I \). An equivalent expression is

\[
A = \sum_{i=1}^{n} \lambda_i q_i q_i^T.
\]

4.2 Matrix Norms

There are several ways to define matrix norms.

4.2.1 Vector-Induced Matrix Norms (Operator Norms)

For a matrix \( A \in \mathbb{R}^{m \times n} \), the vector-induced matrix norms are defined as

\[
\|A\| = \max_{\|x\|=1} \|Ax\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}
\]

Note that the type of norms should match.
Remark: From the definition, it follows that

\[ \|Ax\| \leq \|A\|\|x\| \quad \forall x. \]

This holds for the same type of induced norms. Do not confuse this with the Hölder’s inequality (see below).

- \( \|A\|_1 = \max_{j=1,...,n} \sum_{i=1}^m |A_{ij}| \), the max. absolute column sum.
  
  Note:
  \[ \|Ax\|_1 = \left\| \sum_{j=1}^n A_{j}x_{j} \right\|_1 \leq \sum_{j=1}^n \|A_{j}\|_1 |x_{j}| \leq \max_{1 \leq j \leq n} \|A_{j}\|_1. \]
  And \( x = e_j \) for the \( j \) maximizing \( \|A_j\|_1 \) attains the maximum.

- \( \|A\|_\infty = \max_{i=1,...,m} \sum_{j=1}^n |A_{ij}| \), the max. absolute row sum.

- \( \|A\|_2 = \sqrt{\lambda_{\text{max}}(A^T A)} = \sigma_{\text{max}}(A) \). The max. singular value of \( A \). When \( A \) is symmetric, then \( \|A\|_2 = \lambda_{\text{max}}(A) \). \( \|A\|_2 \) is also called as the spectral radius of \( A \).

Equivalence of Induced Norms

- \( \|A\|_\infty \leq \|A\|_2 \leq \sqrt{mn} \|A\|_\infty \).
- \( \frac{1}{\sqrt{m}} \|A\|_\infty \leq \|A\|_2 \leq \sqrt{m} \|A\|_\infty \).
- \( \frac{1}{\sqrt{n}} \|A\|_1 \leq \|A\|_2 \leq \sqrt{n} \|A\|_1 \).

Hölder’s Inequality

For two matrices \( A \in \mathbb{R}^{m \times k} \) and \( B \in \mathbb{R}^{k \times n} \), we have

\[ \|AB\|_2 \leq \|A\|_p \|B\|_q \]

where \( p, q \in [1, \infty] \) and \( 1/p + 1/q = 1 \). In particular,

\[ \|A\|_2^2 = \|A^T A\|_2 \leq \|A\|_1 \|A\|_\infty. \]

In fact, the inequality holds for more general cases with measurable functions. A special case with \( p = q = 2 \) gives Cauchy-Schwarz inequality.

4.2.2 Elementwise Norms

The 2-norm is defined as follows:

Frobenius Norm (Hilbert-Schmidt Norm)

\[ \|A\|_F = \left( \sum_i \sum_j |A_{ij}|^2 \right)^{1/2} = \sqrt{\text{tr}(A^T A)} = \left( \sum_{i=1}^{\min(m,n)} \sigma_i^2 \right)^{1/2}. \]

The 1-norm and \( \infty \)-norm can be defined similarly.
4.3. POSITIVE (SEMI)DEFINITE MATRIX

Remark: \( \|A\|_2 \leq \|A\|_F \).

4.2.3 Spectral Norms (Shatten Norms)

\[
\|A\|_p = \left( \sum_{i=1}^{\min\{m,n\}} \sigma_i^p \right)^{1/p}.
\]

- \( p = 2 \): Frobenius norm.
- \( p = \infty \): Spectral radius.
- \( p = 1 \): Nuclear norm (Ky Fan norm):
  \[
  \|A\|_1 = \text{tr}(\sqrt{A^T A}) = \sum_{i=1}^{\min\{m,n\}} \sigma_i.
  \]

4.2.4 Norm Inequalities

Consider \( A \in \mathbb{R}^{m \times n} \) with rank \( r \).

- \( \|A\|_2 \leq \|A\|_F \leq \sqrt{r} \|A\|_2 \).
- \( \|A\|_F \leq \|A\|_* \leq \sqrt{r} \|A\|_F \).

4.3 Positive (Semi)Definite Matrix

Definition 4.1. A matrix \( A \in \mathbb{R}^{n \times n} \) is positive semidefinite (p.s.d.) if

\[
x^T A x \geq 0 \text{ for all } x \in \mathbb{R}^n.
\]

A \( A \in \mathbb{R}^{n \times n} \) is positive definite (p.d.) if

\[
x^T A x > 0 \text{ for all } x \neq 0 \in \mathbb{R}^n.
\]

Remark:

- We denote by \( A \succeq 0 \) when \( A \) is psd, and \( A \succ 0 \) when \( A \) is pd.\(^1\)
- If \( A \in \mathbb{R}^{n \times n} \) is symmetric and psd, all eigenvalues of \( A \) are nonnegative.

Proof. Let \( \lambda \) and \( v \) be an eigenvalue and its corresponding eigenvector of \( A \). Since \( A \) is psd, we have

\[
0 \leq v^T (A v) = v^T (\lambda v) = \lambda \|v\|_2^2.
\]

This implies that \( \lambda \geq 0 \). \( \square \)

- If \( A \in \mathbb{R}^{n \times n} \) is symmetric and pd, all eigenvalues of \( A \) are strictly positive.

\(^1\)Do not use these notations in your handwritten homework solutions.
4.4 Spectral Bounds on Quadratic Forms

**Lemma 4.1 (Rayleigh Quotient).** Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$. Let $v_1, \ldots, v_n$ be the associated orthonormal eigenvectors ($\|v_i\|_2 = 1$). For all $x \in \mathbb{R}^n$,

$$\lambda_1 \|x\|_2^2 \leq x^T A x \leq \lambda_n \|x\|_2^2.$$  

In particular, for $x \neq 0$, we have

$$\lambda_1 \leq R(A, x) \leq \lambda_n,$$

where $R(A, x) := \frac{x^T A x}{x^T x}$ is called the Rayleigh quotient of $A$.

**Proof.** Since $A \in \mathbb{R}^{n \times n}$ is symmetric, it has a set of $n$ mutually orthonormal eigenvectors, which forms a basis of $\mathbb{R}^n$. That is, any vector $x \in \mathbb{R}^n$ can be expressed as $x = \sum_{i=1}^n c_i v_i$ with each $c_i \in \mathbb{R}$. Then

$$x^T A x = x^T \left( \sum_{i=1}^n c_i A v_i \right) = \sum_{i=1}^n c_i (\lambda_i v_i)$$

$$= \left( \sum_{i=1}^n c_i v_i \right)^T \left( \sum_{i=1}^n c_i \lambda_i v_i \right) = \sum_{i=1}^n \lambda_i c_i^2 \|v_i\|_2^2 = \sum_{i=1}^n \lambda_i c_i^2.$$

And,

$$\|x\|_2^2 = x^T x = \left( \sum_{i=1}^n c_i v_i \right)^T \left( \sum_{i=1}^n c_i v_i \right) = \sum_{i=1}^n c_i^2.$$

These two relations prove the claim. 

4.5 Continuity of Eigenvalues**

We discuss the continuity of eigenvalues for symmetric matrices.

**Theorem 4.2 (Gershgorin Disk Theorem).** Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix, and $\lambda$ be one of the eigenvalues of $A$. Then for some $i : 1 \leq i \leq n$,

$$|\lambda - A_{ii}| \leq r_i(A), \quad r_i(A) = \sum_{j \neq i} |A_{ij}|.$$

We say $\lambda$ belongs to the Gershgorin disk about $A_{ii}$.

**Proof.** Suppose that $v \neq 0$ is an eigenvector of $A$ associated with $\lambda$. From $Av = \lambda v$, we have for each $i$,

$$\lambda v_i = \sum_{j=1}^n A_{ij} v_j.$$
We choose an index $\ell$ corresponding to the maximal absolute value in $v$, that is, $|v_\ell| = \max_{1 \leq i \leq n} |v_i|$. Then
\[
\lambda v_\ell = \sum_{j=1}^n A_{\ell j} v_j \implies (\lambda - A_{\ell \ell}) = \sum_{j \neq \ell} A_{\ell j} \frac{v_j}{v_\ell}.
\]
Since $\frac{|v_j|}{v_\ell} \leq 1$ for all $j$, we have
\[
|\lambda - A_{\ell \ell}| \leq \sum_{j \neq \ell} |A_{\ell j}|.
\]

Using the above theorem, we prove the continuity of eigenvalues for symmetric matrices.

**Theorem 4.3 (Continuity of Eigenvalues).** Let $\lambda_i : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ is a function that gives the $i$th eigenvalue of $n \times n$ symmetric matrices. Then for $n \times n$ symmetric matrices $A$ and $B$,
\[
|\lambda_i(A) - \lambda_i(B)| \leq n\|A - B\|_2,
\]
and the function $\lambda_i$ is continuous.

**Proof.** Since $A$ is square symmetric, there exists an eigendecomposition of $A = U\Sigma U^T$ where $U = \{u_1, u_2, \ldots, u_n\}$ is an orthonormal matrix ($U^T U = UU^T = I$) consisting of the (normalized) eigenvectors of $A$ and $\Sigma = \text{diag}(\lambda_1(A), \ldots, \lambda_n(A))$ is a diagonal matrix of corresponding eigenvalues. Then,
\[
U^T BU = U^T (A + (B - A))U = U \underbrace{(\Sigma + \Delta)U}_{= \Delta} = \Sigma + \Delta.
\]
This implies that $B$ is similar to $\Sigma + \Delta$, in other words,
\[
\lambda_i(B) = \lambda_i(\Sigma + \Delta), \quad i = 1, 2, \ldots, n. \tag{4.2}
\]

(Two $n \times n$ matrices $A$ and $B$ are similar if $B = P^{-1}AP$ for some invertible matrix $P$. ) Also, since $B - A$ is similar to $\Delta$ by definition, we have
\[
\|B - A\|_2 = \|\Delta\|_2.
\]
For each $i$, we have
\[
r_i(\Sigma + \Delta) = \sum_{j \neq i} |\Sigma_{ij} + \Delta_{ij}| \leq (n - 1)\|\Delta\|_2,
\]
where the last inequality uses the fact that
\[
|\Delta_{ij}| = |e_i^T \Delta e_j| \leq \|e_i\|_2 \|\Delta e_j\|_2 \leq \|\Delta\|_2 \|e_i\|_2 \|e_j\|_2 = \|\Delta\|_2.
\]
Also,
\[
[\Sigma + \Delta]_{ii} = \lambda_i(A) + e_i^T (B - A)U e_i = \lambda_i(A) + u_i^T (B - A)u_i
\]
From the Gershgorin theorem 4.2,
\[
|\lambda_i(\Sigma + \Delta) - [\lambda_i(A) + u_i^T (B - A)u_i]| \leq r_i(\Sigma + \Delta) \leq (n - 1)\|\Delta\|_2.
\]
Using (4.2), we get (using the reverse triangle inequality \(|a - b| \leq |a| - |b|)\),

\[ |\lambda_i(B) - \lambda_i(A)| \leq (n - 1)\|\Delta\|_2 + |u_i^T (B - A) u_i| \leq n\|\Delta\|_2 = n\|B - A\|_2. \]

For any \(\epsilon > 0\), we can choose \(\delta = \epsilon/n\) and consider matrices \(A\) and \(B\) such that \(\|B - A\|_2 < \delta\). Then from the above inequality, we have \(|\lambda_i(A) - \lambda_i(B)| < \epsilon\).

4.1. Consider the Rayleigh quotient \(R(A, x) := \frac{x^T Ax}{x^T x}\) defined for a symmetric matrix \(A \in \mathbb{R}^{n \times n}\) and for all nonzero vectors \(x \in \mathbb{R}^n\). Find a minimizer and a maximizer of this quantity.

4.2. For a matrix \(A \in \mathbb{R}^{m \times n}\), show that

\[ \|A\|_2 = \sqrt{\lambda_{\text{max}}(A^T A)} \]

using the definition of the vector-induced norm.

4.3. Suppose that \(A = uv^T \in \mathbb{R}^{m \times n}\), constructed as an outer product of \(u \in \mathbb{R}^m\) and \(v \in \mathbb{R}^n\). Show that

\[ \|A\|_2 = \|u\|_2 \|v\|_2. \]

4.4. Ridge regression (a.k.a. Tikhonov regularization) solves the following estimation problem for a given constant \(\lambda > 0\),

\[ \min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2}\|b - Ax\|^2_2 + \lambda\|x\|^2_2, \]

in order to find the best estimator \(\hat{x}\) of the true model \(x\) that generates the observations under the linear data model \(b = Ax + \epsilon\), where \(A \in \mathbb{R}^{m \times n}\) is a design matrix and \(\epsilon\) is a noise vector of length \(m\). In contrast, the OLS (ordinary least squares) regression solves the above problem without the regularization term \(\lambda\|x\|^2_2\) (i.e., \(\lambda = 0\)).

(a) Derive the expression of \(\nabla f\) and \(\nabla^2 f\).

(b) \(f\) is a convex function, in which case FONC is also sufficient (to be discussed in the next chapter). Given this, derive the expression of a minimizer. Is there any benefit from considering the extra regularization term while doing this, compared to the OLS?
Chapter 5

Unconstrained Convex Optimization

In this chapter we discuss unconstrained convex optimization problems,

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable convex function. Convex optimization is a major tool in many areas, including machine learning, statistics, and signal processing, due to their simplicity compared to non-convex cases.

5.1 Basic Properties of Convex Functions

**Definition 5.1 (Convex Function).** Let $C$ a convex subset of $\mathbb{R}^n$. A function $f : C \to \mathbb{R}$ is convex over the set $C$ if

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \, \forall x, y \in C, \forall \alpha \in [0, 1].$$

- $f$ is called concave if $-f$ is convex.
- $f$ is called strictly convex if the above inequality is strict for all $x, y \in C$ with $x \neq y$.
- **We can define convex function with extended real values: the function $f : C \to (-\infty, \infty]$, where $C$ is a convex subset of $\mathbb{R}^n$, is convex if the above inequality holds, applying extended arithmetic rules: $\infty + \infty = \infty$, $0 \cdot \infty = 0$, $\alpha \cdot \infty = \infty$ if $\alpha > 0$.**

A generalization of the definition gives us an important inequality:

**Lemma 5.1 (Jensen’s Inequality).** Suppose that $f : C \to \mathbb{R}$ is a convex function defined on a convex domain $C \subset \mathbb{R}^n$. For any $x_1, \ldots, x_m \in C$ and $\alpha_1, \ldots, \alpha_m \geq 0$ such that $\sum_{i=1}^m \alpha_i = 1$, we have

$$f \left( \sum_{i=1}^m \alpha_i x_i \right) \leq \sum_{i=1}^m \alpha_i f(x_i).$$
42  

CHAPTER 5. UNCONSTRAINED CONVEX OPTIMIZATION

**This is a special case of a more general inequality,**

\[
f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)], \quad \text{where } \mathbb{E}[X] = \int_{\Omega} x \, p(x) \, dx, \quad \int_{\Omega} p(x) \, dx = 1.
\]

5.2 Properties of Convex Optimization

**Theorem 5.2.** When \( f \) is convex, any local minimizer \( x^* \) is a global minimizer of \( f \).

**Proof.** For contradiction, suppose that \( x^* \) is a local minimizer but not a global minimizer of \( f \). Then we can find a point \( z \in \mathbb{R}^n \) such that \( f(z) < f(x^*) \). Consider a point \( x \) in the line segment between \( x^* \) and \( z \),

\[
x = \alpha z + (1 - \alpha)x^*, \quad \text{for some } \alpha \in (0, 1].
\]

By the convexity of \( f \),

\[
f(x) \leq \alpha f(z) + (1 - \alpha)f(x^*) < f(x^*). \tag{5.1}
\]

Since any neighborhood \( \mathcal{N} \) of \( x^* \) contains a piece of the line segment between \( x^* \) and \( z \), so there will always exist a point \( x \in \mathcal{N}(x \neq x^*) \) at which (5.1) is satisfied. That is, \( x^* \) is not a local minimizer. This contradicts our assumption that \( x^* \) is a local minimizer. Therefore, \( x^* \) must be a global minimizer.

Now we discuss the first-order sufficient conditions for convex functions. Remember that FONC still holds for convex functions.

**Theorem 5.3 (FOSC for Convex Optimization).** If \( f \) is convex and differentiable, then any stationary point \( x^* \) is a global minimizer of \( f \).

**Proof.** Suppose for contradiction that a global minimizer exists but a stationary point \( x^* \) is not a global minimizer. Then we can find a point \( x \in \mathbb{R}^n \) such that \( f(x) < f(x^*) \). From convexity of \( f \), we get

\[
(x - x^*)^T \nabla_x f(x^*) = \frac{d}{d\lambda} f(x^* + \lambda(x - x^*)) \bigg|_{\lambda=0} \\
= \lim_{\lambda \downarrow 0} \frac{f(x^* + \lambda(x - x^*)) - f(x^*)}{\lambda} \\
\leq \lim_{\lambda \downarrow 0} \frac{\lambda f(x) + (1 - \lambda)f(x^*) - f(x^*)}{\lambda} \\
= f(x) - f(x^*) < 0.
\]

This implies that \( \nabla f(x^*) \neq 0 \), i.e., \( x^* \) is not a stationary point. The claim is proved by contradiction.

**Corollary 5.1.** \( x^* \) is an unconstrained global minimizer of a continuously differentiable convex function \( f : \mathbb{R}^n \to \mathbb{R} \) if and only if \( \nabla f(x^*) = 0 \).
5.3 Strong Convexity

Consider a function \( f : C \rightarrow \mathbb{R} \), where \( C \subset \mathbb{R}^n \) is an open convex set.

**Definition 5.2.** \( f \) is \( \sigma \)-strongly convex for \( \sigma > 0 \) on \( C \) if the following holds for all \( x, y \in C \) and all \( \alpha \in [0, 1] \),

\[
(i) \quad f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) - \frac{\sigma}{2} \alpha(1-\alpha)\|x-y\|^2.
\]

**Theorem 5.4.** The above definition is equivalent to the following conditions when \( f \) is continuously differentiable:

(ii) For all \( x, y \in C \),

\[
f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\sigma}{2} \|y-x\|^2.
\]

(iii) For all \( x, y \in C \),

\[
(\nabla f(y) - \nabla f(x))^T(y-x) \geq \sigma \|y-x\|^2.
\]

If \( f \) is twice continuously differentiable, the definition is also equivalent to the condition

(iv) For all \( x \in C \),

\[
\nabla^2 f(x) \succeq \sigma I,
\]

where the partial order symbol means \( \nabla^2 f(x) - \sigma I \) is positive semi-definite.

**Proof.** **(i) \Rightarrow (ii):** From (i),

\[
\frac{f(y + \alpha(x-y)) - f(y)}{\alpha} \leq f(x) - f(y) - \frac{\sigma}{2}(1-\alpha)\|x-y\|^2.
\]

Taking \( \alpha \rightarrow 0 \) gives (ii).

(ii) \Rightarrow (iii): Applying (ii) twice,

\[
f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\sigma}{2} \|y-x\|^2
\]

\[
f(x) \geq f(y) + \nabla f(y)^T(x-y) + \frac{\sigma}{2} \|x-y\|^2.
\]

(iii) follows when we add the above two inequalities.

(iii) \Rightarrow (ii): From (iii),

\[
(\nabla f(x + t(y-x)) - \nabla f(x))^T(y-x)dt \geq \sigma t \|y-x\|^2
\]

Integrating both sides for \( t \in [0, 1] \) gives (ii).

(ii) \Rightarrow (i): Applying (ii) twice:

\[
f(y) \geq f(y + \alpha(x-y)) + \nabla f(y + \alpha(x-y))^T(\alpha(y-x)) + \frac{\sigma}{2} \alpha^2 \|y-x\|^2
\]

\[
f(x) \geq f(y + \alpha(x-y)) + \nabla f(y + \alpha(x-y))^T((1-\alpha)(x-y)) + \frac{\sigma}{2}(1-\alpha)^2 \|y-x\|^2
\]
Adding the first \( \times (1 - \alpha) \) and the second \( \times \alpha \) gives (i).

(iii) \( \Rightarrow \) (iv): Let \( x, y, z \in C \). From Taylor’s theorem, for a small \( \alpha > 0 \) such that \( z + \alpha(y - x) \in C \), we have

\[
\nabla f(z + \alpha(y - x))^T (y - x) = \nabla f(z)^T (y - x) + \alpha(y - x)^T \nabla^2 f(z + \alpha(y - x))(y - x) + o(\alpha).
\]

Rearranging terms, dividing both sides by \( \alpha \), and using (iii) leads to

\[
(y - x)^T \nabla^2 f(z + \alpha(y - x))(y - x) = \frac{1}{\alpha} (\nabla f(z + \alpha(y - x)) - \nabla f(z))^T (y - x) - \frac{o(\alpha)}{\alpha} \\
\geq \sigma \|y - x\|^2 - \frac{o(\alpha)}{\alpha}.
\]

Taking \( \alpha \to 0 \) and applying Lemma 4.1 gives (iv).

(iv) \( \Rightarrow \) (iii):

\[
(\nabla f(y) - \nabla f(x))^T (y - x) = \left( \int_0^1 \nabla^2 f(x + t(y - x))(y - x)dt \right)^T (y - x) \geq \sigma \|y - x\|^2,
\]

\( \square \)
5.3. STRONG CONVEXITY

Exercises

5.1. ** Show that a convex function \( f : \mathbb{R}^n \to \mathbb{R} \) is continuous.

5.2. Consider \( m \) real-valued scalars \( x_1, x_2, \ldots, x_m \). Show that the arithmetic mean is greater than or equal to the geometric mean of these numbers, i.e.,

\[
\frac{1}{m} \sum_{i=1}^{m} x_i \geq \left( \prod_{i=1}^{m} x_i \right)^{1/m}.
\]

(Hint: log is a concave function.)

5.3. Consider the following penalized regression problem,

\[
\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \| b - Ax \|^2_2 + \lambda \| x \|^2_2,
\]

where \( \lambda \geq 0 \) is a constant, \( A \in \mathbb{R}^{m \times n} \), and \( b \in \mathbb{R}^m \).

(a) Suppose that \( \lambda = 0 \). Is the objective \( f(x) \) strongly convex?

(b) Suppose that \( \lambda > 0 \). Is \( f(x) \) always strongly convex?
Chapter 6

Steepest Descent and Newton’s Method

Consider the unconstrained minimization of a continuously differentiable objective
function \( f : \mathbb{R}^n \to \mathbb{R} \),
\[
\min_{x \in \mathbb{R}^n} f(x).
\]

In general, optimization algorithms are composed of the following elements:

- Check the optimality of the current iterate \( x_k \).
- Find a direction \( p_k \) which improves the objective function value.
- Find a step size \( s_k \) along the direction to have sufficient improvement.
- Update the current iterate, using the direction and stepsize,
\[
x_{k+1} = x_k + s_k p_k.
\]

So far, we have discussed the first element, i.e., the optimality conditions. In this
chapter we begin our discussion on the choice of directions and stepsizes.

6.1 (Exact) Line Search

Suppose that a descent direction \( p_k \) is given: we try to find a step length (step size)
\( \alpha > 0 \) such that
\[
\min_{\alpha > 0} f(x_k + \alpha p_k).
\]
This procedure is called line search.

Finding an optimal step size is called the exact line search, which is often very
expensive. Therefore we need more economical ways to find reasonably good step
sizes.

6.2 Search Direction \( p_k \)

6.2.1 Descent Direction

First of all, in minimization we need to find a direction that will lead to reduction
in objective function values. For the purpose, we use a descent direction \( p \), which
satisfies 
\[ p^T \nabla f(x_k) < 0 \]
at a given point \( x_k \).

For such \( p \), we can see from Taylor’s series (6.1) that,
\[ f(x_k + \alpha p) - f(x_k) \approx \alpha p^T \nabla f(x_k) < 0, \]
and therefore \( f(x_k + \alpha p) < f(x_k) \) for sufficiently small \( \alpha > 0 \).

### 6.2.2 Steepest Descent Direction

The **steepest descent direction**
\[ p_k = -\nabla f(x_k) \]
is a popular choice for a search direction \( p_k \). This is the direction along which the objective \( f \) changes the most rapidly if we move away from \( x_k \). To see this, consider a local model from Taylor’s theorem:
\[ f(x_k + \alpha p) = f(x_k) + \alpha p^T \nabla f(x_k) + o(\|p\|_2). \quad (6.1) \]

Ignoring the little-o term, the factor \( p^T \nabla f(x_k) \) determines how fast \( f \) changes along \( p \), starting from \( x_k \). Indeed the steepest direction is a solution of
\[ p_k = \text{argmin}_{p \in \mathbb{R}^n} p^T \nabla f(x_k), \text{ subject to } \|p\|_2 = 1. \]

Since \( p^T \nabla f(x_k) = \|p\|_2 \|\nabla f(x_k)\|_2 \cos \theta \) where \( \theta \) is the angle between \( p \) and \( \nabla f(x_k) \), the quantity is minimized when \( \cos \theta = -1 \) and
\[ p_k = -\nabla f(x_k)/\|\nabla f(x_k)\|_2. \]

As we discuss later, steepest descent directions may suffer from scaling issues.

### 6.2.3 Newton Direction

When \( f \) is twice continuously differentiable, we can use the **Newton direction** derived from the second-order Taylor series approximation to \( f(x_k + p) \),
\[ f(x_k + p) \approx f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p = m_k(p). \]

Note that the error introduced by the second-order approximation is \( o(\|p\|^2) \).

The Newton direction \( p_k^N \) is well-defined only if \( \nabla^2 f(x_k) \) is not singular, and derived as the minimizer of the strongly convex function \( m_k(p) \)
\[ p_k^N = \text{argmin}_{p \in \mathbb{R}^n} m_k(p) \]
\[ = -\left( \nabla^2 f(x_k) \right)^{-1} \nabla f(x_k). \]

The Newton direction \( p_k^N \) is a descent direction when \( \nabla^2 f(x_k) \) is positive definite and \( \nabla f(x_k) \neq 0 \), since
\[ (p_k^N)^T \nabla f(x_k) = -(p_k^N)^T \left( \nabla^2 f(x_k) \right)^{-1} p_k^N \leq -\lambda_{\min} \|p_k^N\|_2^2 < 0 \]
where \( \lambda_{\min} > 0 \) is the smallest eigenvalue of \( \nabla^2 f(x_k) \).

When \( \nabla^2 f(x_k) \) is not positive definite, we can have two cases: (i) the direction is not defined since \( \nabla^2 f(x_k) \) is singular, or (ii) the direction is not descent. We can make modifications to \( \nabla^2 f(x_k) \) to make \( p_k^N \) a descent direction (discussed later).
6.3 Inexact Line Search

Given the current iterate \( x_k \) and a descent direction \( p_k \), we would like to find a step size \( \alpha_k > 0 \) that minimizes the function

\[
\phi(\alpha) := f(x_k + \alpha p_k).
\]

We face a trade-off: we want such an \( \alpha \) that gives reduction in \( f \) as much as possible, but we do not want to spend too much time choosing a value for \( \alpha \). Therefore, it is practical to consider inexact line search to identify a step size that achieves “good” reduction with reasonable cost.

The shape of \( \phi(\alpha) \) may vary depending on \( f \) and \( p_k \), but the following is always true:

\[
\phi(0) = f(x_k),
\]

\[
\phi'(0) = p_k^T \nabla f(x_k + \alpha p_k) \bigg|_{\alpha=0} = p_k^T \nabla f(x_k) < 0,
\]

where the inequality requires that \( p_k \) is a descent direction. Therefore, \( \phi(\alpha) \) is strictly decreasing near \( \alpha = 0 \), so that \( \phi(\alpha) < \phi(0) \) for small \( \alpha > 0 \).

A naive way of choosing \( \alpha_k > 0 \) that satisfies \( f(x_k + \alpha_k p) < f(x_k) \) could be very inefficient, or it may produce a sequence of iterates that does not even converge to a minimizer.

6.3.1 Wolfe Line Search

For a given descent direction \( p_k \), Wolfe LS searches for a stepsize satisfying two following two conditions, which is collectively called as Wolfe conditions.

**Sufficient Decrease Condition (Armijo Condition)**

The first condition is for finding \( \alpha_k > 0 \) that gives a sufficient decrease in the objective function \( f \), satisfying the following inequality,

\[
\underbrace{f(x_k + \alpha p_k)}_{\phi(\alpha)} \leq \underbrace{f(x_k) + c_1 \alpha \nabla f(x_k)^T p_k}_{\ell(\alpha)}, \quad \text{for some } c_1 \in (0, 1). \quad (6.2)
\]

\( \ell(\alpha) \) is a linear function with a negative slope \((c_1 \nabla f(x_k)^T p_k < 0 \text{ when } p_k \text{ is descent})\), which can be rewritten as

\[
\ell(\alpha) = \phi(0) + c_1 \alpha \phi'(0).
\]

Since \( \phi(\alpha) < f(x_k) \) and \( c_1 \in (0, 1) \), \( \phi(\alpha) \leq \ell(\alpha) \) is true for small \( \alpha > 0 \). In practice, \( c_1 \) is chosen to be a quite small value, e.g. \( c_1 = 10^{-4} \).

**Curvature Condition**

The sufficient condition itself may not be enough since very small stepsizes can be accepted. To avoid this, we also consider the curvature condition,

\[
\nabla f(x_k + \alpha p_k)^T p_k \geq c_2 \nabla f(x_k)^T p_k, \quad \text{for some constant } c_2 \in (c_1, 1). \quad (6.3)
\]
Note that the above expression is exactly the same to write
\[ \phi'(\alpha) \geq c_2 \phi'(0). \]
That is, we look for \( \alpha_k > 0 \) such that the slope \( \phi' \) at \( \alpha_k \) is at least \( c_2 \phi'(0) \). Typical values: \( c_2 = 0.9 \) when \( p_k \) is steepest descent or Newton direction; \( c_2 = 0.1 \) when \( p_k \) is from a nonlinear conjugate gradient method.

\begin{align*}
\phi(\alpha) &= f(x_k + \alpha p_k) \\
\phi'(\alpha) &= \nabla f(x_k + \alpha p_k) \\
\n\end{align*}

\( \alpha \) desired slope

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6.2}
\caption{Sufficient decrease condition illustrated.}
\end{figure}

(Strong) Wolfe Conditions

To summarize, the two Wolfe conditions (6.2) and (6.3) are:

\[ f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T p_k, \]
\[ \nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f(x_k)^T p_k \]

for \( 0 < c_1 < c_2 < 1 \).
The strong Wolfe conditions modifies the second condition so that
\[
\begin{align*}
f(x_k + a_k p_k) & \leq f(x_k) + c_1 a_k \nabla f(x_k)^T p_k, \\
|\nabla f(x_k + a_k p_k)| & \leq c_2 |\nabla f(x_k)| p_k,
\end{align*}
\]
for \(0 < c_1 < c_2 < 1\). The only difference to the Wolfe conditions is that \(\phi'(a_k) = \nabla f(x_k + a_k p_k)^T p_k\) can no longer be large positive value.

### 6.3.2 Backtracking (Armijo) Line Search

Another way to avoid too small stepsizes is to use backtracking line search, which is much simpler than the Wolfe LS and more popular.

**Algorithm 6.1: Backtracking Linesearch**

**Input**: \(a_0 > 0, p_k, \eta \in (0, 1), c \in (0, 1)\);

**for** \(k = 0, 1, 2, \ldots \) **do**

- If \(a_k\) satisfies the sufficient decrease condition (6.2),
  \[
f(x_k + a_k p_k) \leq f(x_k) + c a_k \nabla f(x_k)^T p_k
  \]
  stop;
- Otherwise, \(a_{k+1} = \eta a_k\);

**end**

It is quite obvious from our previous discussion that this algorithm will eventually stop (see the next section for a formal discussion).

In Newton and quasi-Newton methods, the initial stepsize \(a_0\) is chosen to be 1. In other cases, it can have different values.

### 6.3.3 Termination of Inexact Line Search

We can show that there always exists step lengths satisfying the Wolfe conditions. For this, we make use of the following lemma.

**Lemma 6.1** (Mean Value Theorem). For a function \(f : \mathbb{R} \rightarrow \mathbb{R}\) continuous on \([a, b]\) \((a < b)\) and differentiable on \((a, b)\), there exists \(\xi \in (a, b)\) satisfying
\[
\frac{f(b) - f(a)}{b - a} = f'(\xi).
\]

For a function \(f : \mathbb{R}^n \rightarrow \mathbb{R}\) be differentiable on an open set \(D \subset \mathbb{R}^n\). For any two vectors \(x, y \in D\), there exists \(\alpha \in (0, 1)\) satisfying that
\[
f(y) - f(x) = \nabla f(x + \alpha(y - x))^T (y - x).
\]

**Proof.** The univariate case is implied by Rolle’s theorem, so we skip the proof. For the multivariate case, let \(g(y) = f(x + \gamma(y - x))\) for \(\gamma \in [0, 1]\). \(g(\gamma)\) is continuous on \([0, 1]\) and differentiable on \((0, 1)\). From the mean value theorem for univariate functions there exists \(\alpha \in (0, 1)\) such that
\[
g(1) - g(0) = g'(\alpha)
\]
Using $g(1) = f(y)$, $g(0) = f(x)$, and $g'(a) = (y - x)^T \nabla f(x + a(y - x))$ gives the result.

\begin{theorem}
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $p_k$ be a descent direction at $x_k \in \mathbb{R}^n$, and $f$ be bounded below along the ray $\{x_k + \alpha p_k : \alpha > 0\}$. For $0 < c_1 < c_2 < 1$, there exist step lengths satisfying the (strong) Wolfe conditions.
\end{theorem}

\begin{proof}
$\phi(\alpha) = f(x_k + \alpha p_k)$ is bounded below under the given conditions. Since the line $\ell(\alpha) = f(x_k) + c_1 \alpha \nabla f(x_k)^T p_k$ has a strictly negative slope, it must intersect with $\phi$ at some $\alpha > 0$. Let the smallest value of such $\alpha$ be $\alpha' > 0$, i.e.
\[
f(x_k + \alpha' p_k) = f(x_k) + \alpha' c_1 \nabla f(x_k)^T p_k.
\]
Then the sufficient decrease condition is satisfied for all $\alpha \in (0, \alpha']$.

Also, by the mean value theorem, there exists $\alpha'' \in (0, \alpha')$ such that
\[
f(x_k + \alpha' p_k) - f(x_k) = \alpha' \nabla f(x_k + \alpha'' p_k)^T p_k.
\]
The above two formula lead to
\[
\nabla f(x_k + \alpha'' p_k)^T p_k = c_1 \nabla f(x_k)^T p_k > c_2 \nabla f(x_k)^T p_k
\]
since $c_1 < c_2$ and $\nabla f(x_k)^T p_k < 0$. Therefore the curvature condition is also satisfied by $\alpha'' > 0$. (In fact, since $\nabla f$ is continuous, there exists a small interval around $\alpha''$ for which the curvature condition still hold.) That is, the Wolfe conditions are satisfied by (a small interval including) $\alpha''$.

Also, since $\nabla f(x_k + \alpha'' p_k)^T p_k = c_1 \nabla f(x_k)^T p_k < 0$, the strong Wolfe conditions are satisfied by the same $\alpha'' > 0$ (including the interval). \qed}

Exercises

6.1. Consider the (exact) line search problem for a given point \( x_k \in \mathbb{R}^n \) and a descent direction \( p_k \in \mathbb{R}^n \), searching for \( \alpha_k > 0 \) that solves
\[
\min_{\alpha > 0} \phi(\alpha) = f(x_k + \alpha p_k)
\]
for a continuously differentiable function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \). Show that for a convex quadratic function \( f(x) = \frac{1}{2} x^T Q x + c^T x \) where \( Q \) is positive definite, the unique global minimizer \( \alpha_k > 0 \) of \( \phi(\alpha) \) is given by
\[
\alpha_k = -\nabla f(x_k)^T p_k / p_k^T Q p_k.
\]
Hint: first show that \( \phi(\alpha) \) is a strictly convex function, to argue the uniqueness of the solution.

6.2. (Logistic regression) In binary classification tasks, we use a data set \( \{(x_i, y_i)\}_{i=1}^m \) consisting of pairs of an input vector \( x_i \in \mathbb{R}^n \) and a binary class label \( y_i \in \{-1, +1\} \). (Penalized) logistic regression tries to find a probability modeled as a logistic function parametrized by \( \theta \in \mathbb{R}^n \),
\[
p_i := P_\theta(Y = +1 | X = x_i) = \frac{1}{1 + \exp(-\theta^T x_i)}.
\]
Assuming the data points are generated independently from an identical probability distribution, the log likelihood of the model to generate such \( m \) data points is
\[
LL(\theta) = \ln \left( \prod_{i=1}^m p_i^{s_i}(1 - p_i)^{1-s_i} \right) = \sum_{i=1}^m s_i \ln(p_i) + (1 - s_i) \ln(1 - p_i).
\]
where \( s_i \) is the success indicator, \( s_i = +1 \) if \( y_i = +1 \), and \( s_i = 0 \) otherwise. Using the fact that
\[
1 - p_i = \frac{\exp(-\theta^T x_i)}{1 + \exp(-\theta^T x_i)} = \frac{1}{1 + \exp(\theta^T x_i)},
\]
we can simplify the formulation,
\[
LL(\theta) = -\sum_{i=1}^m \ln(1 + \exp(-y_i \theta^T x_i)).
\]
Now we find the best \( \theta^* \) that minimizes the negative log likelihood (plus \( \ell_2 \) regularization),
\[
\theta^* \in \text{argmin}_{\theta \in \mathbb{R}^n} f(\theta) = \sum_{i=1}^m \ln(1 + \exp(-y_i \theta^T x_i)) + \frac{\lambda}{2} \| \theta \|_2^2,
\]
where \( \lambda \geq 0 \) is a penalty parameter.

(1) Derive the expression of \( \nabla f(\theta) \).
2. Implement a function to find $\theta^*$, using gradient descent algorithm (in \texttt{Matlab} or \texttt{Octave}) with backtracking linesearch to find a minimizer of $f(\theta)$. The function should look like:

$$[\text{theta, status}] = \text{LogRegGD}(X, y, \lambda, \epsilon, \text{maxiter}, \text{maxLSiter})$$

- $X \in \mathbb{R}^{m \times n}$: input vectors
- $y \in \{-1, +1\}^m$: class labels
- $\lambda \geq 0$.
- $\epsilon > 0$: optimality threshold. In your algorithm, declare that the iterate $\theta_k$ is optimal when
  \[
  \|\nabla f(\theta_k)\|_\infty \leq \epsilon.
  \]
  Use $\epsilon = 10^{-4}$.
- maxiter: maximum number of gradient descent iterations. Stop your algorithm if the iteration counter exceeds this value (default: 1000).
- maxLSiter: maximum number of linesearch trials (default: 25).

For linesearch, use these parameters: initial guess $\alpha_0 = 1.0$, multiplier $\eta = .5$, and $c = 10^{-4}$.

The outcome variables are:
- theta: the solution.
- status: the status of stopping (optimality or maxiter).

In each iteration, your algorithm should show for the current iterate $\theta_k$,
- $k$: the iteration number.
- $f(\theta_k)$: the objective value.
- $\|\nabla f(\theta_k)\|_\infty$: the optimality criterion value.
- a stepsize found by backtracking linesearch.
- the number of linesearch iterations taken to find the stepsize.

3. Download the MNIST digit classification data from the lecture website: filename \texttt{mnist67.scale.1k.mat}. The file contains two data variables: $X \in \mathbb{R}^{m \times n}$, where each row of $X$ is a vectorized $28 \times 28$ images of handwritten digits. Only digits 6 and 7 are present in $X$; $y \in \mathbb{R}^m$, which contains binary labels $y_i = +1$ if the digit is 6, and $y_i = -1$ if 7. Run your code to find solutions on this data set.

3.1. Compute the prediction accuracy, i.e.,

$$\text{accuracy} = \frac{\# \text{ of correct predictions}}{\# \text{ of examples queried for prediction}}$$

The prediction for an input point $x_i$ is accurate if $p_i > .5$ and $y_i = +1$, or $p_i \leq .5$ and $y_i = -1$. Run your code and show prediction accuracy on the given data set with $\lambda = 0$, $\lambda = 100$, and $\lambda = 10^4$. 

(3.b) What do you observe, as you try larger \( \lambda \) values?

(3.c)** Plot the image of the solution vector \( \theta^* \). You can reshape and plot it as a 28 \times 28 image in Matlab as follows,

```matlab
d = reshape(theta, 28, 28);
imagesc(d);
colormap(gray);
```

What differences do you see, as \( \lambda \) become larger?
Chapter 7
Convergence of Steepest Descent and Newton’s Methods

7.1 Global Convergence: Descent Direction + Wolfe Line Search

Consider the following unconstrained minimization of a continuously differentiable objective function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \),
\[
\min_{x \in \mathbb{R}^n} f(x).
\]
Given the current iterate \( x_k \in \mathbb{R}^n, k \geq 1 \), and a descent direction \( p_k \in \mathbb{R}^n \), we generate the next iterate \( x_{k+1} \) by
\[
 x_{k+1} = x_k + \alpha_k p_k. \tag{7.1}
\]
where the step length \( \alpha_k > 0 \) can be determined by a line search strategy satisfying the Wolfe conditions (for \( 0 < c_1 < c_2 < 1 \)),
\[
 f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T p_k \quad \text{(Sufficient Decrease)}
\]
\[
 \nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f(x_k)^T p_k \quad \text{(Curvature)}
\]
For analysis, we define the angle \( \theta_k \) between \(-\nabla f(x_k)\) and a search direction \( p_k \),
\[
 \cos \theta_k = \frac{-\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|_2 \|p_k\|_2}
\]

**Theorem 7.1** (Global Convergence of Wolfe Line Search). Consider iterations of the form (7.1) with a descent direction \( p_k \) and \( \alpha_k \) satisfying the Wolfe conditions. Suppose that \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is bounded below in \( \mathbb{R}^n \) and \( f \) is continuously differentiable in an open set \( \mathcal{N} \) containing the level set \( \mathcal{L} := \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\} \), where \( x_0 \) is the starting point of the iteration. Suppose also that the gradient \( \nabla f \) is Lipschitz continuous on \( \mathcal{N} \), i.e. there exist a constant \( L > 0 \) such that
\[
 \|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2, \text{ for all } x, y \in \mathcal{N}.
\]
When plugging in this into the sufficient decrease condition, we get
\[
\sum_{k \geq 0} \cos^2 \theta_k \| \nabla f(x_k) \|_2 < \infty \quad (\text{Zoutendijk condition}).
\]

**Proof.** Subtracting \( \nabla f(x_k)^T p_k \) from both side of the curvature condition and replacing \( x_{k+1} = x_k + \alpha_k p_k \) gives
\[
(\nabla f(x_{k+1}) - \nabla f(x_k))^T p_k \geq (c_2 - 1) \nabla f(x_k)^T p_k
\]
On the other hand, Cauchy-Schwarz inequality and the Lipschitz condition lead to
\[
(\nabla f(x_{k+1}) - \nabla f(x_k))^T p_k \leq \| \nabla f(x_{k+1}) - \nabla f(x_k) \|_2 \| p_k \|_2 \leq \alpha_k L \| p_k \|_2^2.
\]
Combining the above two inequalities, we obtain
\[
\alpha_k \geq \frac{(c_2 - 1) \nabla f(x_k)^T p_k}{L \| p_k \|_2^2}.
\]
Plugging in this into the sufficient decrease condition, we get
\[
f(x_{k+1}) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T p_k \leq f(x_k) + \frac{c_1}{L \| p_k \|_2^2} (c_2 - 1) \nabla f(x_k)^T p_k^2
\]
Using the definition of \( \cos \theta_k \), this can be rewritten as
\[
f(x_{k+1}) \leq f(x_k) - c \cos^2 \theta_k \| \nabla f(x_k) \|_2^2
\]
for \( c := c_1 (1 - c_2) / L > 0 \). Summing up this expression for all indices up to \( k \), we have
\[
f(x_{k+1}) \leq f(x_0) - c \sum_{j=0}^{k} \cos^2 \theta_j \| \nabla f(x_j) \|_2^2.
\]
That is, for all \( k \) we have
\[
\sum_{j=0}^{k} \cos^2 \theta_j \| \nabla f(x_j) \|_2^2 \leq \frac{1}{c} (f(x_0) - f(x_{k+1}))
\]
Since \( f \) is bounded below, \( f(x_{k+1}) > -M \) for some positive scalar \( M \) for all \( k \), and therefore \( f(x_0) - f(x_{k+1}) < f(x_0) + M < M' \) for some positive scalar \( M' \) for all \( k \). Hence, by taking \( k \to \infty \), we get the desired result:
\[
\sum_{k=0}^{\infty} \cos^2 \theta_k \| \nabla f(x_k) \|_2^2 < \infty.
\]

The Zoutendijk condition implies that
\[
\lim_{k \to \infty} \cos^2 \theta_k \| \nabla f(x_k) \|_2^2 = 0.
\]
When \( p_k \) is a descent direction, we have \( \cos \theta_k > 0 \) for all \( k \) (strictly speaking, we need \( \cos \theta_k \geq \delta \) for some \( \delta > 0 \), as we discuss below), and therefore the above condition implies that
\[
\lim_{k \to \infty} \| \nabla f(x_k) \|_2 = 0 \quad (\text{Global Convergence}).
\]
We call algorithms that satisfy the above condition as *globally convergent*. Theorem 7.1 is the strongest global convergence result available for Wolfe line search strategy. Note however that it only guarantees to find a stationary point, not necessarily a minimizer (in convex minimization a stationary point is a global minimizer).

### 7.1.1 Condition Number

The condition $\cos \theta_k \geq \delta, \delta > 0$ for all $k$ is crucial to guarantee the global convergence of line search.

- Steepest descent: $p_k = -\nabla f(x_k)$, and therefore $\cos \theta_k = 1 > 0$ when $\nabla f(x_k) \neq 0$.
- Newton’s direction: $p_N^k = -H_k^{-1}\nabla f(x_k)$ with $H_k = \nabla^2 f(x_k)$. Recall that when $H_k$ is positive definite, we showed that $p_N^k$ is a descent direction. We define the condition number of $H_k$,

$$\kappa(H_k) = \|H_k\|_2 \|H_k^{-1}\|_2 = \frac{\lambda_{\text{max}}(H_k)}{\lambda_{\text{min}}(H_k)}$$

which is equal to the ratio between the largest and the smallest eigenvalues of $H_k$. When $\kappa(H_k) \leq M$ for all $k$ for some $M > 0$, it follows that $\cos \theta_k \geq \frac{1}{M} > 0$ for all $k$.

### 7.2 Rate of Convergence

#### 7.2.1 Convergence Rate of Steepest Descent

We first study the behavior of steepest descent for an easy case: the objective function is a strongly convex quadratic,

$$f(x) = \frac{1}{2}x^TQx + c^Tx$$

where $Q \in \mathbb{R}^{n \times n}$ is symmetric and positive definite. In this case, we can perform an exact line search, finding $\alpha > 0$ that minimizes $f(x_k - \alpha \nabla f(x_k))$ (ref. Ex. 6.1.)

$$\alpha_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T Q \nabla f(x_k)} > 0.$$

(why $\alpha_k > 0$?) (what is the range of $\alpha_k$?) (what would happen if $Q$ is not positive definite?) Using this, we can write an explicit expression for the steepest descent update of minimizing strongly convex quadratic functions $f$ as

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

To quantify the rate of convergence, we define the weighted norm

$$\|x\|^2_Q = x^TQx.$$
CHAPTER 7. CONVERGENCE OF STEEPEST DESCENT AND NEWTON’S METHODS

(This is a norm when \( Q \) is symmetric positive definite.) Using the fact that an unconstrained minimizer \( x^* \) of \( f(x) \) satisfies \( \nabla f(x^*) = Qx^* + c = 0 \), we can show that
\[
\frac{1}{2} \| x - x^* \|_Q^2 = f(x) - f(x^*). \tag{7.2}
\]

Using the update rule above and the fact that \( g_k := \nabla f(x_k) = Q(x_k - x^*) \), we can show that
\[
\| x_{k+1} - x^* \|_Q^2 = \left( 1 - \frac{(g_k^T g_k)^2}{(g_k^T Q g_k)(g_k^T Q^{-1} g_k)} \right) \| x_k - x^* \|_Q^2.
\]

This can be further simplified to give the following theorem:

**Theorem 7.2** (Convergence Rate of Steepest Descent: Quadratic Case). For the steepest descent directions with exact line search for minimizing the strongly convex quadratic function \( f(x) = \frac{1}{2} x^T Q x + c^T x \), we have
\[
\| x_{k+1} - x^* \|_Q^2 = \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \| x_k - x^* \|_Q^2.
\]
where \( 0 < \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n \) are the eigenvalues of \( Q \).

This theorem can be shown using the Kantorovich inequality:

**Lemma 7.3** (Kantorovich Inequality). For a positive definite symmetric \( n \times n \) matrix \( Q \) and for any nonzero vector \( x \in \mathbb{R}^n \),
\[
\frac{(x^T x)^2}{(x^T Q x)(x^T Q^{-1} x)} \geq 4 \lambda_1 \lambda_n \frac{(\lambda_1 + \lambda_n)^2}{(\lambda_1 + \lambda_n)^2}
\]
where \( \lambda_1 \) and \( \lambda_n \) are the smallest and the largest eigenvalues of \( Q \).

Few observations can be made for Theorem 7.2:

- With (7.2), it implies that
\[
f(x_{k+1}) - f(x^*) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 (f(x_k) - f(x^*)).\]

- That is, the rate of convergence in terms of objective function values is \( Q \)-linear, since
\[
\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} = \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 \in (0, 1) \ \forall k
\]

- The rate depends on the condition number \( \kappa(Q) = \lambda_n / \lambda_1 \), since
\[
\left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 = \left( \frac{\kappa(Q) - 1}{\kappa(Q) + 1} \right)^2.
\]

- If \( \kappa(Q) \) is large (the contour of quadratic function \( f \) is elliptical), then convergence becomes slow. In fact, zigzagging can happen.
• In particular, when all eigenvalues are the same, so that \( \lambda_1 = \lambda_n \) (so that the contour of \( f \) is perfectly spherical, the steepest descent converges in a single iteration in terms of objective function values.

The convergence rate of steepest descent with exact line searches for general nonlinear objective functions is given by the following theorem:

**Theorem 7.4** (Convergence of Steepest Descent). Let \( f : \mathbb{R}^n \to \mathbb{R} \) is twice continuously differentiable, and suppose that the iterates generated by the steepest descent with the exact line search converges to a point \( x^* \) at which \( \nabla^2 f(x^*) \) is positive definite. Let \( r \) be a scalar such that

\[
 r \in \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}, 1 \right)
\]

where \( \lambda_1 \leq \cdots \leq \lambda_n \) are the eigenvalues of \( \nabla^2 f(x^*) \). Then for all sufficiently large \( k \), we have

\[
 f(x_{k+1}) - f(x^*) \leq r^2 (f(x_k) - f(x^*)).
\]

In general, when an inexact line search is used (e.g. Wolfe line search), the rate of convergence does not improve.

### 7.2.2 Convergence Rate of Newton’s Method

For a twice continuously differentiable function \( f : \mathbb{R}^n \to \mathbb{R} \), Newton directions are given by

\[
p_N^k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k).
\]

Suppose that the inverse exists. Recall that if \( \nabla^2 f(x_k) \) is not positive definite, \( p_N^k \) may not be a descent direction.

Here we discuss the local convergence of Newton’s method: we know that if \( \nabla f^2(x^*) \) is p.d., then if the Hessian \( \nabla^2 f(x) \) is continuous then \( \nabla^2 f(x) \) will be p.d. as well in a neighborhood of \( x^* \). In such a region Newton directions are well defined and descent.

**Theorem 7.5** (Local Convergence of Newton’s Method). Suppose that \( f : \mathbb{R}^n \to \mathbb{R} \) is twice differentiable and that \( \nabla^2 f(x) \) is Lipschitz continuous in a neighborhood of a solution \( x^* \) with a constant \( L > 0 \), that is,

\[
 \|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L \|x - y\|_2 \quad \forall x, y \in \mathcal{N}(x^*)
\]

where at \( x^* \) the SOSC is satisfied. Consider iterations \( x_{k+1} = x_k + p_N^k \) (i.e., with stepsize 1), which starts from a point point \( x_0 \) which is sufficiently close to \( x^* \). Then,

- the sequence \( \{x_k\} \) converges Q-quadratically to \( x^* \),
- the sequence \( \{\|\nabla f(x_k)\|_2\} \) converges Q-quadratically to zero.
CHAPTER 7. CONVERGENCE OF STEEPEST DESCENT AND NEWTON’S METHODS

Proof. **

\[ x_{k+1} - x^* = x_k + p_k^N - x^* = x_k - x^* - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \]
\[ = \nabla^2 f(x_k)^{-1} [\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))]. \]

Note \( \nabla f(x^*) = 0 \) from SOSC. Using the fact above,

\[ \nabla f(x_k) - \nabla f(x^*) = \int_0^1 J_t[\nabla f(x^* + t(x_k - x^*))] dt \]
\[ = \int_0^1 \nabla^2 f(x^* + t(x_k - x^*))(x_k - x^*) dt, \]

we get

\[ \|x_{k+1} - x^*\|_2 \]
\[ = \|\nabla^2 f(x_k)^{-1} [\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))]\|_2 \]
\[ \leq \|\nabla^2 f(x_k)^{-1}\|_2 \|\nabla^2 f(x_k)(x_k - x^*) - (\nabla f(x_k) - \nabla f(x^*))\|_2 \]
\[ = \|\nabla^2 f(x_k)^{-1}\|_2 \left\| \int_0^1 [\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))](x_k - x^*) dt \right\|_2 \]
\[ \leq \|\nabla^2 f(x_k)^{-1}\|_2 \int_0^1 \| [\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))](x_k - x^*) \|_2 dt \]
\[ \leq \|\nabla^2 f(x_k)^{-1}\|_2 \| [\nabla^2 f(x_k) - \nabla^2 f(x^* + t(x_k - x^*))] \|_2 \|x_k - x^*\|_2 dt \]
\[ \leq \|\nabla^2 f(x_k)^{-1}\|_2 \|x_k - x^*\|_2^2 \int_0^1 L(1 - t) dt \]
\[ = \frac{L}{2} \|\nabla^2 f(x_k)^{-1}\|_2 \|x_k - x^*\|_2^2 \]

Since \( \nabla^2 f(x^*) \) is positive definite, there exists a radius \( r > 0 \) such that \( \|\nabla^2 f(x_k)^{-1}\|_2 \leq 2\|\nabla^2 f(x^*)^{-1}\|_2 \) for all \( x_k \) with \( \|x_k - x^*\|_2 \leq r \). Together with the expression above, we obtain

\[ \|x_{k+1} - x^*\|_2 \leq L \|\nabla^2 f(x^*)^{-1}\|_2 \|x_k - x^*\|_2^2 \leq \bar{L} \|x_k - x^*\|_2^2 \]

where \( \bar{L} = L \|\nabla^2 f(x^*)^{-1}\|_2 > 0 \).

Choosing \( x_0 \) so that \( \|x_0 - x^*\|_2 \leq \min(r, 1/(2\bar{L})) \), we can deduce from this inequality that the sequence converges to \( x^* \) (try this by yourself), where the rate of convergence is quadratic.
Finally, using the facts that $x_{k+1} - x_k = p_k^N$ and $\nabla f(x_k) + \nabla^2 f(x_k)p_k^N = 0$,

$$\|\nabla f(x_{k+1})\|_2 = \|\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k)p_k^N\|_2$$

$$= \left\| \int_0^1 \nabla^2 f(x_k + tp_k^N)(x_{k+1} - x_k) - \nabla^2 f(x_k)p_k^N dt \right\|_2$$

$$\leq \int_0^1 \|\nabla^2 f(x_k + tp_k^N) - \nabla^2 f(x_k)\|_2 \|p_k^N\|_2 dt$$

$$\leq \|p_k^N\|_2^2 \int_0^1 L t dt = \frac{L}{2} \|p_k^N\|^2_2$$

$$\leq \frac{L}{2} \|\nabla^2 f(x_k)^{-1}\|_2^2 \|\nabla f(x_k)\|_2^2$$

$$= \tilde{L} \|\nabla f(x_k)\|_2^2.$$  

7.3 Newton’s Method with Hessian Modification

The Newton’s direction $p_k^N$ is obtained from solving

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

However, when $\nabla^2 f(x_k)$ is not positive definite, the direction $p_k^N$ may not be a descent direction. We describe Newton’s algorithm with modified Hessian to overcome the issue:

**Algorithm 7.1:** Newton’s Method with Linesearch and Hessian Modification

*Input:* $x_0$;

*for* $k = 0, 1, 2, \ldots$ *do*

Choose $B_k = \nabla^2 f(x_k) + E_k$,

where

$$E_k = \begin{cases} 0 & \text{if } \nabla^2 f(x_k) \text{ is sufficiently positive definite} \\ E_k & \text{otherwise } E_k \text{ is chosen to ensure that } B_k \text{ is sufficiently positive definite} \end{cases}$$

Solve $B_k p_k = -\nabla f(x_k)$;

Choose $\alpha_k$ from Wolfe or Armijo backtracking linesearch;

$x_{k+1} \leftarrow x_k + \alpha_k p_k$;

*end*

7.3.1 Global Convergence

The global convergence of this algorithm is given in the following theorem:
CHAPTER 7. CONVERGENCE OF STEEPEST DESCENT AND NEWTON’S METHODS

Theorem 7.6. Let \( f : \mathbb{R}^n \to \mathbb{R} \) be twice continuously differentiable on an open set \( D \subset \mathbb{R}^n \), and assume that the starting point \( x_0 \) of Algorithm 7.1 is such that the level set \( \mathcal{L} = \{ x \in D : f(x) \leq f(x_0) \} \) is compact. Then if for some \( C > 0 \),
\[
\kappa(B_k) = \|B_k\|_2\|B_k^{-1}\|_2 \leq C, \quad k = 0, 1, 2, \ldots.
\]
then
\[
\lim_{k \to \infty} \nabla f(x_k) = 0.
\]

Remark:

- Suppose that \( \{x_k\} \) converges to \( x^* \) where \( \nabla^2 f(x^*) \) is sufficiently positive definite. Then for all sufficiently large \( k \), \( E_k = 0 \) and \( \alpha_k = 1 \) (due to Theorem 7.5 will be chosen, so that Algorithm 7.1 reduces to a pure Newton method with local quadratic rate of convergence).
- When \( \nabla^2 f(x^*) \) is singular or close to singular, then \( E_k \) may not vanish, and the convergence might be only linear.
- We would like to make the modification as small as possible, so that the curvature information in the Hessian is preserved.

7.3.2 Eigenvalue Modification

Consider the eigendecomposition of the Hessian matrix \( \nabla^2 f(x_k) \),
\[
\nabla^2 f(x_k) = Q\Lambda Q^T = \sum_{i=1}^{n} \lambda_i q_i q_i^T
\]
where \( Q \in \mathbb{R}^{n \times n} \) is orthonormal, i.e., \( Q^T Q = QQ^T = I \).

Take an example where \( \nabla f(x_k) = (1, -3, 2)^T \) and \( \nabla^2 f(x_k) = \text{diag}(10, 3, -1) \). At \( x_k \), the pure Newton direction is \( p_N^k = (-1, 1, 2)^T \) is not descent. We can consider a positive definite approximation \( B_k \) of \( \nabla^2 f(x_k) \), replacing all negative eigenvalues by a small \( u > 0 \) (e.g. \( u = \sqrt{\epsilon} \), where \( \epsilon \) is the machine precision). In our example, we use (for \( \epsilon = 10^{-16} \)),
\[
B_k = \sum_{i=1}^{2} \lambda_i q_i q_i^T + u q_3 q_3^T = \text{diag}(10, 3, 10^{-8}) > 0.
\]

This preserves the curvature information along \( q_1 \) and \( q_2 \). However, the search direction based on \( B_k \) is
\[
p_k = -B_k^{-1} \nabla f(x_k) = -\sum_{i=1}^{2} \frac{1}{\lambda_i} q_i (q_i^T p_N^k) - \frac{1}{u} q_3 (q_3^T \nabla f(x_k))
\approx -(2 \cdot 10^8) q_3.
\]

This is nearly parallel to \( q_3 \) and quite long for a small \( u \). Therefore it is not clear if this approach will be effective, since it is against the reason to use Newton's method.
Other modifications are possible: (1) flip the sign of the negative eigenvalues; (2) set the terms corresponding to the negative eigenvalues to zero in the above expression, so that the associated eigenvectors will have no effect; (3) we can choose \( u \) adaptively, so that it will not be excessive.

**Diagonal and Non-Diagonal Modifications**

We can consider eigenvalue modification problem, for \( H = \nabla^2(x_k) = Q\Lambda Q^T \), as finding \( \Delta H \) such that \( \lambda_{\min}(H + \Delta H) \geq u \).

**Non-Diagonal:** \( \Delta H = Q \text{diag}(\tau) Q^T \), \( \tau_i = \begin{cases} 0, & \lambda_i \geq u \\ u - \lambda_i, & \lambda_i < u \end{cases} \).

This choice minimizes the Frobenius norm of the correction matrix. It follows that \( B_k = H + \Delta H = Q(\Lambda + \text{diag}(\tau))Q^T \).

**Diagonal:** Another choice, to minimize the spectral radius of the correction matrix, is \( \Delta H = \tau I \), \( \tau = \max(0, u - \lambda_{\min}(H)) \).

This shifts all eigenvalues.

### 7.3.3 Modified Cholesky Factorization

For a symmetric positive definite matrix \( A \in \mathbb{R}^{n \times n} \), there exists a Cholesky factorization,

\[
A = LDL^T,
\]

where \( L \) is a lower triangular matrix with 1’s on the diagonal, and \( D \) is a diagonal matrix with positive elements on the diagonal.

\[
A = LDL^T = \\
\begin{bmatrix}
1 & 0 & 0 \\
L_{21} & 1 & 0 \\
L_{31} & L_{32} & 0
\end{bmatrix} \\
\begin{bmatrix}
D_1 & 0 & 0 \\
0 & D_2 & 0 \\
0 & 0 & D_3
\end{bmatrix} \\
\begin{bmatrix}
1 & L_{21} & L_{31} \\
0 & 1 & L_{32} \\
0 & 0 & 1
\end{bmatrix}
\]

When \( A \) is positive definite, one can show that the diagonal elements \( D_j \) are positive.

If \( A \) is not positive definite, then the factorization may not exist, or it may exists but with very large elements in \( L \) and \( D \). In such cases, we can modify the algorithm so that the elements of \( D \) will be sufficiently positive.

Therefore we choose two parameters \( \delta \) and \( \beta \), so that in Algorithm 7.2 it is satisfied that \( D_j \geq \delta \), \( |L_{ij}\sqrt{D_j}| \leq \beta, \ i = j + 1, \ldots, n \).
Algorithm 7.2: Cholesky Factorization

\[
\text{for } j = 1, 2, \ldots, n \text{ do}
\]
\[
D_j = A_{jj} - \sum_{s=1}^{j-1} D_s L_{js}^2.
\]
\[
\text{for } i = j + 1, \ldots, n \text{ do}
\]
\[
L_{ij} = \frac{1}{D_j} \left( A_{ij} - \sum_{s=1}^{j-1} D_s L_{is} L_{js} \right).
\]
\end{algorithm}

Algorithm 7.3: Modified Cholesky Factorization

Input: $\delta > 0, \beta > 0$

\[
\text{for } j = 1, 2, \ldots, n \text{ do}
\]
\[
c_{jj} = A_{jj} - \sum_{s=1}^{j-1} D_s L_{js}^2.
\]
\[
D_j = \max \left\{ |c_{jj}|, \left( \frac{\theta}{\beta} \right)^2, \delta \right\}, \quad \theta_j = \max_{j<i \leq n} |c_{ij}|
\]
\[
\text{for } i = j + 1, \ldots, n \text{ do}
\]
\[
c_{ij} = A_{ij} - \sum_{s=1}^{j-1} D_s L_{is} L_{js},
\]
\[
L_{ij} = c_{ij} / D_j.
\]
\end{algorithm}

Algorithm 7.4: Cholesky Factorization (In-Memory)

\[
R = A;
\]
\[
\text{for } k = 1, 2, \ldots, n \text{ do}
\]
\[
\text{for } j = k + 1, \ldots, n \text{ do}
\]
\[
R_{j,j:n} = R_{j,j:n} - R_{k,j:n} R_{k,j} / R_{k:k}
\]
\end{algorithm}

\[
R_{k;k:n} = R_{k;k:n} / \sqrt{R_{k:k}}.
\]
From Algorithm 7.3, we have \( c_{ij} = L_{ij}D_j \) and therefore

\[
|L_{ij}\sqrt{D_j}| = \frac{|c_{ij}|}{\sqrt{D_j}} \leq \frac{|c_{ij}|\beta}{\theta_j} \leq \beta, \quad \forall i > j.
\]

The most efficient implementation include in-memory versions, shown in Algorithm 7.4, to create \( A = R^T R \) for upper triangluar matrix \( R \). In each of the inner loop, we need one division, \( n - j + 1 \) multiplications, and \( n - j + 1 \) subtractions, for a total of \( \sim 2(n - j) \) flops. And therefore the total number of flops is approximately

\[
\sum_{k=1}^{n} \sum_{j=k+1}^{n} 2(n - j) \sim 2 \sum_{j=1}^{n} j \sum_{k=1}^{n} k^2 \sim \frac{1}{3} n^3.
\]

This is half of the cost of Gaussian elimination, which takes \( \sim \frac{2}{3} n^3 \) flops.
Chapter 8

Quasi-Newton Methods

The main disadvantage of Newton’s method is to compute the Hessian $\nabla^2 f(x_k) \in \mathbb{R}^{n \times n}$ and its inverse, which is computationally challenging when the dimension $n$ is large.

In quasi-Newton methods, we use an approximation matrix $B_k$ to the true Hessian, and compute search directions by

$$p_k = -B_k^{-1}\nabla f(x_k).$$

This direction can be understood as the minimizer of a quadratic model function of $f$ with the approximate Hessian $B_k$,

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p.$$

We like to design $B_k$ so that the model function at the next iteration,

$$m_{k+1}(p) = f(x_{k+1}) + \nabla f(x_{k+1})^T p + \frac{1}{2} p^T B_{k+1} p,$$

reflect the information we have, namely $f(x_{k+1})$ and $\nabla f(x_k)$. Using the update $x_{k+1} = x_k + \alpha_k p_k$ where $\alpha_k$ is chosen by Wolfe line search, this implies that we make sure

$$\begin{cases} m_{k+1}(0) = f(x_{k+1}) \\ \nabla m_{k+1}(-\alpha_k p_k) = \nabla f(x_{k+1}) - \alpha_k B_{k+1} p_k = \nabla f(x_k) \end{cases}$$

Rearranging the second equation, we have

**Definition 8.1** (Secant Equation).

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

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

Also, $B_{k+1}$ can be positive definite only if the following condition is satisfied:
Definition 8.2 (Curvature Condition).
\[ s_k^T y_k > 0. \]

When the objective \( f \) is strongly convex, the curvature condition is satisfied for any \( x_k \) and \( x_{k+1} \). However it will not always true for nonconvex functions: in such cases we need to enforce the condition, by choosing the stepsize \( \alpha_k \) satisfying the Wolfe (or strong Wolfe) conditions. From the second Wolfe condition, we have
\[ \nabla f(x_{k+1})^T s_k \geq c_2 \nabla f(x_k)^T s_k, \quad \text{for } c_2 \in (0, 1), \]
and therefore
\[ y_k^T s_k = [\nabla f(x_{k+1}) - \nabla f(x_k)]^T s_k \geq (c_2 - 1) \alpha_k \nabla f(x_k)^T p_k. \]

Since \( c_2 < 1 \) and \( p_k \) is a descent direction, it follows that \( y_k^T s_k > 0. \)

8.1 Convergence of Quasi-Newton Method

We consider an extended form of Newton’s method, where the search direction is chosen by solving the following system of linear equations,
\[ B_k p_k = -\nabla f(x_k), \]
where \( B_k \) is a symmetric and positive definite matrix: it is easy to check \( p_k = -B_k^{-1} \nabla f(x_k) \) is a descent direction.

We first describe a global convergence result for this setting.

Theorem 8.1 (Superlinear Convergence). For \( f : \mathbb{R}^n \to \mathbb{R} \), twice continuously differentiable, consider a sequence \( \{x_k\} \) generated by \( x_{k+1} = x_k + \alpha_k p_k \) where \( p_k \) is a descent direction and \( \alpha_k \) satisfies Wolfe conditions with \( c_1 \leq 1/2 \). If \( \{x_k\} \) converges to a point \( x^* \) such that \( \nabla f(x^*) = 0 \) and \( \nabla^2 f(x^*) \) is positive definite, and if the search directions satisfy
\[ \lim_{k \to \infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k)p_k\|_2}{\|p_k\|_2} = 0, \quad (8.1) \]
then
\begin{itemize}
  \item the step size \( \alpha_k = 1 \) is admissible for all \( k > k_0 \) for some \( k_0 \).
  \item if \( \alpha_k = 1 \) for all \( k > k_0 \), then \( \{x_k\} \) converges to \( x^* \) superlinearly.
\end{itemize}

Remarks: The condition (8.1) implies that the matrix \( B_k \approx \nabla^2 f(x_k) \) for all large \( k \). Using \( p_k = -B_k^{-1} \nabla f(x_k) \), the condition becomes
\[ \lim_{k \to \infty} \frac{\|(B_k - \nabla^2 f(x_k))p_k\|_2}{\|p_k\|_2} = 0. \quad (8.2) \]
This tells us that we only need \( B_k \) becomes increasingly accurate approximations to \( \nabla^2 f(x_k) \), along the quasi-Newton directions \( p_k \).

The following shows that the Dennis & Moré condition is sufficient and necessary for (local) superlinear convergence.
The secant equation has possibly many solutions. We obtain a unique \( B \) as described above. We have

\[ \|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathcal{N}(x^*) .\]

Then \( \{x_k\} \) converges superlinearly iff the condition (8.2) holds.

**Proof:**

\[
p_k - p_k^N = [\nabla^2 f(x_k)]^{-1}(\nabla^2 f(x_k)p_k + \nabla f(x_k))
\]

Together with the condition (8.2),

\[ \|p_k - p_k^N\| \leq \|\nabla^2 f(x_k)\|^{-1}\|\nabla^2 f(x_k) - B_k\|p_k\| = o(\|p_k\|) \]

since \( \|\nabla^2 f(x_k)\|^{-1} < \infty \) for all \( x_k \) sufficiently close to \( x^* \) as \( \nabla^2 f(x_k) > 0 \). Together with the local quadratic convergence of Newton's method (Theorem 7.5), we have

\[ \|x_k + p_k - x^*\| \leq \|x_k + p_k^N - x^*\| + \|p_k - p_k^N\| = O(\|x_k - x^*\|^2) + o(\|p_k\|) \]

This also implies that \( \|p_k\| = O(\|x_k - x^*\|) \), and therefore

\[ \|x_k + p_k - x^*\| \leq o(\|x_k - x^*\|) \]

which implies superlinear convergence.

### 8.2 Davidon-Fletcher-Powell (DFP) Update

The secant equation has possibly many solutions. We obtain a unique \( B_k \) by considering the following optimization problem,

\[
\min_{B} \frac{1}{2}\|B - B_k\|_W^2,
\text{ s.t. } B = B^T, Bs_k = y_k. \tag{8.3}
\]

The norm in the objective is the weighted Frobenius norm,

\[ \|A\|_W = \|AW^{1/2}A^{1/2}\|_F \]

where the matrix square root is defined for a symmetric matrix \( W \) and its spectral decomposition \( W = Q\Sigma Q^T \) that \( W^{1/2} = Q\Sigma^{1/2}Q^T \).

The unique solution of (8.3), with an invertible symmetric matrix \( W \) satisfying \( Wy_k = s_k \), is given by

\[
B_{k+1} = (I - \rho_k y_k s_k^T)B_k(I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T, \tag{DFP} \]

where

\[ \rho_k = (y_k^T s_k)^{-1}. \]

The inverse of \( B_k \), denoted by \( H_k = B_k^{-1} \) will be more useful for implementation, which can be derived using the Sherman-Morrison-Woodbury formula,

\[
H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}, \tag{DFP} \]

where

\[ s_k = y_k - H_k y_k. \]
Algorithm 8.1: BFGS

Input: $x_0$, convergence tolerance $\epsilon$, and $H_0$.

for $k = 0, 1, 2, \ldots$ do
  if $\|\nabla f(x_k)\| \leq \epsilon$ then
    break;
  end
  Compute $p_k = -H_k \nabla f(x_k)$;
  Choose $\alpha_k$ using (strong) Wolfe linesearch (with $\alpha_0 = 1$, $c_1 = 10^{-4}$, and $c_2 = 0.9$);
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
  Compute $H_{k+1}$ by the BFGS update rule.
end

8.2.1 Broyden-Fletcher-Goldfarb-Shanno (BFGS) Update

We can instead consider the inverse $H_k$ in the secant equation,

$$H_k y_k = s_k,$$

and consider finding a unique solution by solving the following problem,

$$\min_H \frac{1}{2} \|H - H_k\|_W^2,$$

s.t. $H = H^T, H y_k = s_k$.

The unique solution, with $W$ satisfying $W s_k = y_k$, is given by

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T,$$  \hspace{1cm} (BFGS) (8.6)

for the same definition of $\rho_k = \frac{1}{y_k s_k}$. This is the most popular update rule for quasi-Newton methods.

In Algorithm 8.1, each iteration can be computed with $O(n^2)$ arithmetic operations, plus the cost for function and gradient evaluations. This is a benefit compared to $O(n^3)$ operations needed for the Newton's method, although its convergence rate is slower (superlinear in practice) than Newton's method (quadratic).

The BFGS formulation in terms of $B_k$ can be derived using the Sherman-Morrison-Woodbury formula,

$$B_{k+1} = B_k - \frac{B_k s_k y_k^T B_k}{y_k s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k s_k}.$$  \hspace{1cm} (BFGS) (8.7)

Using this form in a native way will be less efficient than the previous form, although an efficient variant is possible updating Cholesky factors of $B_k$.

Initial $H_0$

The initial matrix $H_0$ often set to $cI$, with some scalar $c$ and the identity matrix $I$. Then the first step becomes $p_0 = -c \nabla f(x_0)$, and a too large value of $c$ may require quite a few line search steps to find a suitable stepsize.
An effective heuristic is to set
\[ H_0 = \frac{y_k^T s_k}{y_k^T y_k} \]
and then apply BFGS update to obtain \( H_1 \). This is a variant of Barzilai-Borwein spectral approach (1988), to obtain \( c_k I \approx (\nabla^2 f(x_k))^{-1} \), i.e., \( c_k y_k \approx s_k \) in the least-squares sense,
\[ c_k = \arg\min_c ||c y_k - s_k||_2^2. \]
Similar idea can be applied to obtain \( d_k I \approx \nabla^2 f(x_k) \), to have
\[ d_k = \frac{y_k^T s_k}{s_k^T s_k}, \]
which leads to another heuristic,
\[ H_0 = \frac{s_k^T s_k}{y_k^T y_k} I. \]
It is hard to tell which will work better, but if we know \( \nabla^2 f(x_k) \succ 0 \), then whichever positive should be chosen.
This by itself can be considered a quasi-Newton approximating the Hessian with an identity matrix multiplied by a scalar. This approach is often called a scaled-gradient method.

### 8.2.2 Wolfe Line Search

As we see in Theorem, we should try \( \alpha_1 = 1 \) first in line search since it will be always admissible eventually (under some conditions), producing superlinear convergence. It is often suggested to perform a quite inaccurate but economical line search, with \( c_1 = 10^{-4} \) and \( c_2 = 0.9 \) in Wolfe conditions.

In other cases, we can choose the initial guess \( \alpha_0 \) so that \( \alpha_0 \nabla f(x_k)^T p_k = \alpha_{k-1} \nabla f(x_{k-1})^T p_{k-1} \), assuming that first-order change in the function will be the same as in the previous step, leading to
\[ \alpha_0 = \frac{\nabla f(x_{k-1})^T p_{k-1}}{\nabla f(x_k)^T p_k}. \]
Another way is to interpolate a quadratic function to \( f(x_{k-1}), f(x_k), \) and \( \nabla f(x_{k-1})^T p_{k-1} \) and set the initial guess to be the minimizer. This gives
\[ \alpha_0 = \frac{2(f(x_k) - f(x_{k-1}))}{\phi'(0)}. \]
If \( x_k \to x^* \) superlinearly, this ratio converges to 1. Also, we can adjust this by
\[ \alpha_0 = \min\{1, 1.01 \alpha_0\} \]
In Algorithm 8.2, we first note that \( \{\alpha_i\} \) is monotonically increasing. Also, we have the fact that the interval \( (\alpha_{i-1}, \alpha_i) \) contains step lengths satisfying the strong Wolfe condition is one of the following criterion is satisfied:
Algorithm 8.2: Line Search with Wolfe Conditions

Input: $\phi(0), \phi'(0), \alpha_0 = 0, \alpha_{\text{max}} > 0$ and $0 < c_1 < c_2 < 1$;
Choose $\alpha_1 \in (\alpha_0, \alpha_{\text{max}})$;

for $i = 1, 2, \ldots$ do

    Compute $\phi(\alpha_i)$;
    if $\phi(\alpha_i) > \phi(0) + c_1 \alpha_i \phi'(0)$ or $i > 1$ and $\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ then
        $\alpha_* = \text{zoom}(\alpha_{i-1}, \alpha_i)$ and stop;
    end

    Compute $\phi'(\alpha_i)$;
    if $|\phi'(\alpha_i)| \leq -c_2 \phi'(0)$ then
        $\alpha_* = \alpha_i$ and stop;
    end

    if $\phi'(\alpha_i) \geq 0$ then
        $\alpha_* = \text{zoom}(\alpha_i, \alpha_{i-1})$ and stop;
    end

    Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\text{max}})$;

end

• $\alpha_i$ violates the sufficient decrease condition.
• $\phi(\alpha_i) \geq \phi(\alpha_{i-1})$.
• $\phi'(\alpha_i) \geq 0$.

Otherwise, we the trial stepsize for the next iteration. Here, we can simply multiply the current value by some constant, or could use more complex procedure. In any case, it should increase the value quickly enough so that the upper limit $\alpha_{\text{max}}$ will be tried in a finite number of iterations.

In the subprocedure zoom, the input arguments satisfy the following:

• the interval $(\alpha_\ell, \alpha_u)$ contains step lengths satisfying the strong Wolfe condition.
• $\alpha_\ell$ has the smallest function value satisfying the sufficient decrease condition, amongst all steps generated so far.
• $\alpha_u$ satisfies $\phi'(\alpha_\ell)(\alpha_u - \alpha_\ell) < 0$.

In zoom, it is advised to avoid $\alpha_j$ being too close to one of the endpoints. Also, if $f(x_k)$ and $f(x_{k-1})$ is very close, then it would be hard to obtain a good stepsize in a few iterations: therefore it is advised to stop the line search procedure after trying few (e.g. 10) stepsizes.

8.3 Convergence of BFGS

8.3.1 Global Convergence

BFGS works very robustly in practice, however there is no truely global convergence results for BFGS in case of optimizing general nonlinear objective functions. Here we state a global convergence result, in a bit restricted sense.
Algorithm 8.3: zoom

Input: \( \alpha, \alpha_u \);

for \( j = 1, 2, \ldots \) do

Choose \( \alpha_j \) between \( \alpha_L \) and \( \alpha_U \): e.g. as a minimizer of a cubic function interpolating \( \phi(\alpha_L), \phi(\alpha_U), \phi'(\alpha_L), \) and \( \phi'(\alpha_U) \):

\[
\alpha_j = \alpha_u - (\alpha_u - \alpha_L) \frac{\phi'(\alpha_u) + d_2 - d_1}{\phi'(\alpha_u) - \phi'(\alpha_L) + 2d_2}
\]

with

\[
d_1 = \phi'(\alpha_L) + \phi'(\alpha_u) - 3 \frac{\phi(\alpha_L) - \phi(\alpha_U)}{\alpha_L - \alpha_U}
\]

\[
d_2 = \text{sign}(\alpha_u - \alpha_L)[d_1^2 - \phi'(\alpha_L)\phi'(\alpha_U)]^{1/2}.
\]

Compute \( \phi(\alpha_j) \);

if \( \phi(\alpha_j) > \phi(0) + c_1 \alpha_j \phi'(0) \) or \( \phi(\alpha_j) \geq \phi(\alpha_L) \) then

\[
\alpha_u = \alpha_j;
\]

else

Compute \( \phi'(\alpha_j) \);

if \( |\phi'(\alpha_j)| \leq -c_2 \phi'(0) \) then \( \alpha_\ast = \alpha_j \) and stop;

if \( \phi'(\alpha_j)(\alpha_u - \alpha_L) \geq 0 \) then \( \alpha_u = \alpha_L \);

\[
\alpha_L = \alpha_j;
\]

end

end

Theorem 8.3 (Global Convergence of BFGS). Let \( f: \mathbb{R}^n \to \mathbb{R} \) be twice continuously differentiable, \( H_0 \) be any symmetric positive definite matrix, and \( x_0 \) be a starting point for which:

- the level set \( \mathcal{L} = \{ x \in \mathbb{R}^n : f(x) \leq f(x_0) \} \) is convex, and
- there exists positive constants \( m, M \) such that

\[
mI \preceq \nabla^2 f(x) \preceq MI, \quad \forall x \in \mathcal{L}.
\]

Then the sequence \( \{ x_k \} \) generated by Algorithm 8.1 converges to the minimizer \( x^\ast \) of \( f \).

Proof. **

First we denote the average Hessian by \( \bar{G}_k \),

\[
\bar{G}_k = \int_0^1 \nabla^2 f(x_k + t \alpha_k p_k) dt,
\]

which satisfies the secant equation by Taylor’s theorem,

\[
y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k.
\]
Then the given conditions can be understood as
\[ m_k := \frac{y_k^T y_k}{y_k^T s_k} \geq m, \quad M_k := \frac{y_k^T y_k}{y_k^T s_k} \leq M, \]
where we have defined the fractions as \( m_k \) and \( M_k \).

Let us define
\[ \cos \theta_k = \frac{s_k^T B_k s_k}{\|s_k\|\|B_k s_k\|}, \quad q_k = \frac{s_k^T B_k s_k}{s_k^T s_k}. \]

Then we have
\[ \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} = \frac{\|B_k\|^2\|s_k\|^2 s_k^T B_k s_k}{(s_k^T B_k s_k)^2 \|s_k\|^2} = \frac{q_k}{\cos^2 \theta_k}. \]

Together with the BFGS update rule, we can show that
\[
\begin{align*}
\text{tr}(B_{k+1}) &= \text{tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k} = \text{tr}(B_k) - \frac{q_k}{\cos^2 \theta_k} + M_k \\
\det(B_{k+1}) &= \det(B_k) \frac{y_k^T s_k}{s_k^T B_k s_k} = \det(B_k) \left( \frac{y_k^T s_k}{s_k^T B_k s_k} \right) = \det(B_k) \frac{m_k}{q_k}.
\end{align*}
\]

We combine the trace and determinant by introducing the following function for a positive definite matrix \( B \),
\[ \psi(B) = \text{tr}(B) - \ln(\det(B)) > 0. \]

Replacing terms, we get
\[
\psi(B_{k+1}) = \psi(B_k) + M_k - \frac{q_k}{\cos^2 \theta_k} - \ln(\det(B_k)) - \ln m_k + \ln q_k
\]
\[
= \psi(B_k) + (M_k - \ln m_k - 1) + \left[ 1 - \frac{q_k}{\cos^2 \theta_k} + \ln \frac{q_k}{\cos^2 \theta_k} \right] + \ln \cos^2 \theta_k.
\]

Since \( h(t) = 1 - t + \ln t \leq 0 \) for all \( t > 0 \), the expression within the square bracket is nonpositive, and therefore
\[
0 < \psi(B_{k+1}) \leq \psi(B_0) + c(k + 1) + \sum_{j=0}^{k} \ln \cos^2 \theta_j, \quad (8.8)
\]
where we can assume \( c := M - \ln m - 1 > 0 \), WLOG.

Now, for contraction let us assume \( \cos \theta_j \to 0 \). Then there must exist \( k_1 > 0 \) s.t. for all \( j > k_1 \),
\[ \ln \cos^2 \theta_j < -2c. \]

Then from (8.8), we get
\[
0 < \psi(B_0) + c(k+1) + \sum_{j=0}^{k_1} \ln \cos^2 \theta_j + \sum_{j=k_1+1}^{k} (-2c) = \psi(B_0) + \sum_{j=0}^{k_1} \ln \cos^2 \theta_j + 2ck_1 + c - ck.
\]

The right-hand side is negative for large \( k \), however, leading to a contradiction. Therefore there exists a subsequence \( \{j_k\}_{k=1,2,...} \) such that \( \cos \theta_{j_k} \geq \delta > 0 \). Then by Zoutendijk’s result (Theorem 7.1, this implies that \( \|\nabla f(x_k)\| \to 0 \), and since the objective is strongly convex in the level set \( \mathcal{L} \), it implies in turn that \( x_k \to x^* \).
8.3. CONVERGENCE OF BFGS

This theorem can be generalized for the restricted Broyden class, i.e., using

\[ B_{k+1} = (1 - \alpha_k)B_{k+1}^{\text{BFGS}} + \alpha_k B_{k+1}^{\text{DFP}}, \quad \alpha_k \in [0, 1], \]

except for the DFP (i.e. \( \alpha_k = 1 \)).

Also, the analysis can be extended to show linear convergence of iterates under the given conditions, in particular \( \|x_k - x^*\| \) converges fast enough to make

\[
\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty. \tag{8.9}
\]

Convergence Rate of BFGS

Instead of using the rather restrictive assumptions in the previous section, we establish (local) superlinear convergence assuming that (8.9) holds.

**Theorem 8.4** (Superlinear Convergence of BFGS). Suppose that \( f \in \mathcal{C}^2 \), and \( \{x_k\} \) generated by the BFGS algorithm converges to \( x^* \) where

\[
\nabla^2 f(x^*) - \nabla f(x) \leq L\|x^* - x\|, \quad \forall x \in \mathcal{N}(x^*),
\]

for an \( L > 0 \). Also assume that (8.9) holds. Then \( \{x_k\} \) converges to \( x^* \) superlinearly.

**Proof.**

First, let us define the following,

\[
G_k = \nabla^2 f(x_k), \quad G_* = \nabla^2 f(x^*),
\]

and

\[
\tilde{s}_k = G_*^{1/2}s_k, \quad \tilde{y}_k = G_*^{-1/2}y_k, \quad \tilde{B}_k = G_*^{-1/2}B_k G_*^{-1/2}.
\]

We also define \( \cos \tilde{\theta}_k, \tilde{q}_k, \tilde{m}_k \), and \( \tilde{M}_k \) similarly to the previous proof but with \( \tilde{s}_k \), \( \tilde{y}_k \), and \( \tilde{B}_k \).

By pre- and post-multiplying \( G_*^{-1/2} \) to the BFGS update formula, we get

\[
\tilde{B}_{k+1} = \tilde{B}_k - \frac{\tilde{B}_k \tilde{s}_k \tilde{y}_k^T \tilde{B}_k}{\tilde{s}_k^T \tilde{B}_k \tilde{s}_k} + \frac{\tilde{y}_k \tilde{y}_k^T}{\tilde{y}_k^T \tilde{y}_k}.
\]

As this exactly reflects the BFGS formula, it follows from the previous proof that

\[
\psi(\tilde{B}_{k+1}) = \psi(\tilde{B}_k) + (\tilde{M}_k - \ln \tilde{m}_k - 1) + \left[ 1 - \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} + \ln \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} \right] + \ln \cos^2 \tilde{\theta}_k. \tag{8.10}
\]

From \( y_k = \tilde{G}_k \tilde{s}_k \), we get

\[
y_k - G_* s_k = (\tilde{G}_k - G_*)s_k,
\]

which implies that

\[
\tilde{y}_k - \tilde{s}_k = G_*^{-1/2}(\tilde{G}_k - G_*)G_*^{-1/2} \tilde{s}_k.
\]

Using the assumption of Lipschitz continuity, the above leads to

\[
\|\tilde{y}_k - \tilde{s}_k\| \leq \|G_*^{-1/2}\| \|\tilde{s}_k\| \|\tilde{G}_k - G_*\| \leq \|G_*^{-1/2}\|^2 \|\tilde{s}_k\| L \varepsilon_k
\]
CHAPTER 8. QUASI-NEWTON METHODS

where \( \epsilon_k = \max\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\} \). That is,

\[
\frac{\|\tilde{y}_k - \tilde{s}_k\|}{\|\tilde{s}_k\|} \leq \tilde{c}\epsilon_k
\]

(8.11)

for some constant \( \tilde{c} > 0 \).

Together with triangle inequality, the above implies that

\[
\|\tilde{y}_k\| - \|\tilde{s}_k\| \leq \tilde{c}\epsilon_k\|\tilde{s}_k\|, \quad \|\tilde{s}_k\| - \|\tilde{y}_k\| \leq \tilde{c}\epsilon_k\|\tilde{s}_k\|
\]

so that

\[
(1 - \tilde{c}\epsilon_k)\|\tilde{s}_k\| \leq \|\tilde{y}_k\| \leq (1 + \tilde{c}\epsilon_k)\|\tilde{s}_k\|.
\]

Together with the expression from squaring (8.11), the above leads to

\[
(1 - \tilde{c}\epsilon_k)^2\|\tilde{s}_k\|^2 - 2\tilde{y}_k^T\tilde{s}_k + \|\tilde{s}_k\|^2 \leq \|\tilde{y}_k\|^2 - 2\tilde{y}_k^T\tilde{s}_k + \|\tilde{s}_k\|^2 \leq \tilde{c}^2\epsilon_k^2\|\tilde{s}_k\|^2.
\]

That is,

\[
2\tilde{y}_k^T\tilde{s}_k \geq (1 - 2\tilde{c}\epsilon_k + \tilde{c}^2\epsilon_k^2 + 1 - \tilde{c}^2\epsilon_k^2)\|\tilde{s}_k\|^2 = 2(1 - \tilde{c}\epsilon_k)\|\tilde{s}_k\|^2.
\]

Also, we can show that

\[
\bar{m}_k = \frac{\tilde{y}_k^T\tilde{s}_k}{\|\tilde{s}_k\|^2} \geq 1 - \tilde{c}\epsilon_k, \quad \bar{M}_k = \frac{\|\tilde{y}_k\|^2}{\tilde{y}_k^T\tilde{s}_k} \leq 1 + \tilde{c}\epsilon_k.
\]

Since \( x_k \rightarrow x^* \), we have \( \epsilon_k \rightarrow 0 \), and therefore there must exist \( c > \tilde{c} > 0 \) such that for all sufficiently large \( k \),

\[
\bar{M}_k \leq 1 + \frac{2\bar{c}}{1 - \tilde{c}\epsilon_k} \leq 1 + c\epsilon_k.
\]

Using the fact that \( h(t) = 1 - t + \ln t \leq 0 \) for \( t > 0 \), we have

\[
\frac{-x}{1-x} - \ln(1-x) = h\left(\frac{1}{1-x}\right) \leq 0.
\]

For large \( k \), we can assume that \( \tilde{c}\epsilon_k < 1/2 \), and therefore

\[
\ln(1 - \tilde{c}\epsilon_k) \geq \frac{-\tilde{c}\epsilon_k}{1 - \tilde{c}\epsilon_k} \geq -2\tilde{c}\epsilon_k.
\]

Also, from the expression involving \( \bar{m}_k \) above, we have

\[
\ln \bar{m}_k \geq \ln(1 - \tilde{c}\epsilon_k) \geq -2\tilde{c}\epsilon_k > -2\epsilon_k.
\]

Using the conditions on \( \bar{m}_k \) and \( \bar{M}_k \) in (8.10), it leads to

\[
0 < \psi(\bar{B}_{k+1}) \leq \psi(\bar{B}_k) + 3c\epsilon_k + \ln \cos^2 \tilde{\theta}_k + \left[ 1 - \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} + \ln \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} \right].
\]

Summing up this expression and using the summability assumption of iterate distances, we get

\[
\sum_{j=0}^{\infty} \left( \ln \frac{1}{\cos^2 \theta_j} \right) \left[ 1 - \frac{\tilde{q}_j}{\cos^2 \theta_j} + \ln \frac{\tilde{q}_j}{\cos^2 \theta_j} \right] \leq \psi(\bar{B}_0) + 3c \sum_{j=0}^{\infty} \epsilon_j < +\infty.
\]
8.4. SHERMAN-MORRISON-WOODBURY FORMULA

Note that the expression in the square brackets is nonpositive and \( \ln(1/\cos^2 \tilde{\theta}_j) \geq 0 \) for all \( j \). Therefore, we get the limits

\[
\ln(1/\cos^2 \tilde{\theta}_j) \to 0, \quad 1 - \frac{\tilde{q}_j}{\cos^2 \tilde{\theta}_j} + \ln \frac{\tilde{q}_j}{\cos^2 \tilde{\theta}_j} \to 0, \quad \text{as } j \to \infty.
\]

That is,

\[
\cos^2 \tilde{\theta}_j \to 1, \quad \tilde{q}_j \to 1.
\]

Using these in terms of Dennis-Moré characterization gives the results: that is,

\[
\| G_k^{-1/2}(B_k - G_k) s_k \| = \| \tilde{B}_k s_k \| \to 0, \quad 1 - \tilde{q}_k \cos^2 \tilde{\theta}_k + \tilde{q}_k s_k^T \tilde{q}_k \to 0, \quad \text{as } j \to \infty.
\]

Since the RHS converges to 0, it implies that

\[
\| (B_k - G_k) s_k \| \to 0,
\]

and therefore from Theorem 8.1, \( \alpha_k = 1 \) will be admissible for Wolfe LS near the solution, and the convergence rate is superlinear.

8.4 Sherman-Morrison-Woodbury Formula

For matrices \( A \in \mathbb{R}^{n \times n} \), \( U \in \mathbb{R}^{n \times k} \), \( C \in \mathbb{R}^{k \times k} \), \( V \in \mathbb{R}^{k \times n} \), we have

\[
(A + UCV)^{-1} = A^{-1} - A^{-1} U(C^{-1} + VA^{-1} U)^{-1} VA^{-1}
\]

This can be shown directly, or from the blockwise matrix inversion formula.

8.4.1 Blockwise Matrix Inversion

Consider the inversion of the following block matrix,

\[
M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}
\]

If \( A \) is invertible, then this can be rewritten as

\[
M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix}
\]

Here, we can see that

\[
\det(M) = \det(A) \det(D - CA^{-1} B).
\]

The matrix \( D - CA^{-1} B \) is called the Schur complement of \( A \) in \( M \). If it is also invertible, then we can write \( M^{-1} \) in a block form,

\[
\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} I & -A^{-1} B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & (D - CA^{-1} B)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} = \begin{bmatrix} A^{-1} + A^{-1} B(D - CA^{-1} B)^{-1} CA^{-1} & -A^{-1} B(D - CA^{-1} B)^{-1} \\ (D - CA^{-1} B)^{-1} CA^{-1} & (D - CA^{-1} B)^{-1} \end{bmatrix}
\]
If $D$ and its Schur complement $A-DB^{-1}C$ are invertible, then we can play a similar trick to have
\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}^{-1} = 
\begin{bmatrix}
(A-DB^{-1}C)^{-1} & -(A-DB^{-1}C)^{-1}BD^{-1} \\
-D^{-1}C(A-DB^{-1}C)^{-1} & D^{-1} + D^{-1}C(A-DB^{-1}C)^{-1}BD^{-1}
\end{bmatrix}
\]
Comparing the above two expressions, we can deduce the Sherman-Morrison-Woodbury formula.

8.4.2 Positive Definiteness of Symmetric Block Matrices
If $M$ is symmetric, so that $A$ and $D$ are symmetric and $C = B^T$, i.e.,
\[
M = \begin{bmatrix}
A & B \\
B^T & D
\end{bmatrix} = 
\begin{bmatrix}
I & 0 \\
B^TA^{-1} & I
\end{bmatrix}
\begin{bmatrix}
A & 0 \\
0 & D - B^TA^{-1}B
\end{bmatrix}
\begin{bmatrix}
I & 0 \\
B^TA^{-1} & I
\end{bmatrix}^T.
\]
Note that the Schur complement $D - B^TA^{-1}B$ is symmetric. From this, we can deduce the following:
- $M \succ 0$ if $A \succ 0$ and $D - B^TA^{-1}B \succ 0$.
- If $A \succ 0$, then $M \succeq 0$ if $D - B^TA^{-1}B \succeq 0$.

Similarly,
- $M \succ 0$ if $D \succ 0$ and $A - BD^{-1}B^T \succ 0$.
- If $D \succ 0$, then $M \succeq 0$ if $A - BD^{-1}B^T \succeq 0$. 
8.4. SHERMAN-MORRISON-WOODBURY FORMULA  81

Exercises

8.1. Implement the quasi-Newton algorithm with BFGS update (Algorithm 8.1). The function interface should look like:

\[
[x, \text{status}] = \text{qn}(x0, f, g, \epsilon, \text{maxiter}, \text{maxLSiter}, \text{maxzoomiter}, c1, c2, \text{ls}, \text{dir})
\]

where the arguments are:

- \(x0\): a starting point.
- \(f\): the pointer to a function \(f: \mathbb{R}^n \rightarrow \mathbb{R}\) that returns the objective function value for an argument \(x \in \mathbb{R}^n\).
- \(g\): the pointer to a function that returns the objective gradient for \(x\).
- \(\epsilon\): a threshold value to stop when \(\|\nabla f(x_k)\|_\infty < \epsilon\). (Use \(\epsilon = 10^{-7}\).
- \(\text{maxiter}\): max. no. of iterations of the quasi-Newton algorithm.
- \(\text{maxLSiter}\): max. no. of line search iterations.
- \(\text{maxzoomiter}\): max. no. zoom iters (used only for Wolfe line search.)
- \(c1, c2\): params for line search (\(c2\) only for WolfeLS.)
- \(\text{ls}\): select a line search algorithm (‘bt’: backtracking, or ‘wolfe’: Wolfe LS).
- \(\text{dir}\): search direction (‘gd’: gradient descent; ‘bfgs’: quasi-Newton with BFGS).

The output arguments are

- \(x\): solution vector.
- \(\text{status}\): the status of stopping (optimal or maxiter).

It should show the information of iterations like the following,

\[
\begin{array}{ccccccc}
\text{iter} & \text{fval} & \text{opt} & \text{alp} & \text{lsiter} \\
1 & -5.000e+01 & 1.150e+01 & 1.509e-01 & 3 & 0 \\
2 & -2.503e+02 & 3.497e-01 & 5.095e-02 & 2 & 0 \\
3 & -2.503e+02 & 2.695e-01 & 1.000e+00 & 0 & 0 \\
\end{array}
\]

which shows the iteration counter \(k\), the objective function value at \(x_k\), \(\|\nabla f(x_k)\|_\infty\), the stepsize \(\alpha_k\), and the number of iterations spent for line search (show two numbers for Wolfe LS: LS iter and zoom iter).

BFGS Choose \(H_0 = I\), the \(n \times n\) identity matrix.

Wolfe LS You also need to implement the Wolfe line search, Algorithm 8.2. Use

\(\alpha_0 = 10^{-4}\) and \(\alpha_{\text{max}} = 1\) when you call it from your quasi-Newton algorithm. For quasi-Newton, use an extra argument such as \(\alpha_{\text{first}} = 1\) to check it first against the Wolfe conditions. When the line search algorithm calls the subroutine zoom, there should be an enough gap between \(\alpha_i\) and \(\alpha_{max}\); therefore just return the current \(\alpha_i\) when e.g. \(\alpha_i - \alpha_{i-1} < 10^{-8}\). Lastly, to implement \(\alpha_{i+1} \in (\alpha_i, \alpha_{\text{max}})\), one way can be:

\[
\text{alp} = \min(\text{alpmax}, \text{alp0} + i*(\text{alpmax}-\text{alp0})/\text{maxLSiter});
\]

Backtracking LS In backtracking LS, use \(c = c1\), \(\alpha_0 = 1\), and \(\eta = 0.5\).
Functions to Optimize  Implement the codes that return the objective function value and gradient of (a) logistic regression (ex. 6.2) and (b) the Rosenbrock function (ex. 3.1.), for a given iterate $x \in \mathbb{R}^n$. Let’s denote the name of these implemented functions as $f_{\text{logreg}}, g_{\text{logreg}}, f_{\text{rosenbrock}},$ and $g_{\text{rosenbrock}}$. 

Question  Run your algorithm for the following settings:
(a) Logistic regression (using the mnist data from ex. 6.2.)

```matlab
n = size(X,2); x0 = zeros(n,1);
[x, status] = qn(x0, @(x) f_logreg(x), @(x) g_logreg(x), 1e-12, 1000, 20, 20, 1e-4, .9, 'wolfe', 'bfgs');
```

(b) Minimize the Rosenbrock function on $\mathbb{R}^2$.

```matlab
x0 = zeros(2,1);
[x, status] = qn(x0, @(x) f_rosenbrock(x), @(x) g_rosenbrock(x), 1e-12, 5000, 20, 20, 1e-4, .9, 'wolfe', 'bfgs');
```

(1) Show the progress of iterations in your report for the above two runs.

(2) Repeat the above with other choices of search direction and line search algorithms. Plot the following: (i) x-axis: iter, y-axis: obj. function value, (ii) x: iter, y: optimality criterion value, each comparing gradient descent and BFGS for two different choices of line search algorithms. Generate these plots separately for (a) logistic regression and (b) Rosenbrock function, so to have four plots in total. What do you observe? Does your observation match with the their properties discussed in the lecture? Explain.

Grading:
- Correct implementation: BFGS (10), Wolfe LS (10).
- Correct behavior of BFGS with Wolfe LS (logreg: 5 + rosenbrock: 5)
- Correct plots (10).
- Correct observation & explanation of the behaviors of backtracking/Wolfe LS in regard to gradient descent/quasi-Newton (10)
- Extra (10): comparisons of different initialization, different ways of WolfeLS implementation, etc.
Chapter 9

Conjugate Gradient Method

We will discuss two types of conjugate gradient (CG) methods in this chapter:

- **Linear CG**: for minimizing a strictly convex quadratic function in large dimensions, or finding a solution for a large system of linear equations.
- **Nonlinear CG**: for finding minimizers of nonlinear functions.

Comparing to (steepest) gradient descent, conjugate gradient methods work much faster spending similar computation. Although CG does not attain as fast convergence rate of (quasi) Newton methods, but it does not require to store matrices to compute its search directions.

9.1 Linear Conjugate Gradient Method

The linear CG method is an efficient iterative procedure for solving a large system of linear equations for a symmetric and positive definite matrix $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$,

$$Ax = b.$$ 

Note that this problem can be seen as solving the following minimization problem,

$$\min_{x \in \mathbb{R}^n} \phi(x) = \frac{1}{2} x^T Ax - b^T x.$$ 

Both problems above have the same unique solution.

The solution also can be found by Gaussian elimination ($\#\text{FLOP} \sim \frac{2}{3}n^3$) or by Cholesky factorization with forward/backward substitution ($\#\text{FLOP} \sim \frac{n^3}{3} + n^2$). With linear CG, we will see that the optimal solution can be found within $n$ iterations (or faster under certain conditions), where the cost of each iteration is $\#\text{FLOP} \sim n^2$.

We define the residual of the linear system as

$$r(x) := Ax - b = \nabla \phi(x),$$

and for a particular $x_k$, we denote $r_k := r(x_k)$.

The linear CG method is capable of finding a set of nonzero vectors (search directions) \(\{p_0, p_1, \ldots, p_\ell\}\) that satisfy conjugacy. A set of vectors are conjugate with respect to $A$ if all of them satisfy the condition,

$$p_i^T Ap_j = 0, \quad \forall i \neq j.$$
Conjugate vectors are also linearly independent (exercise).

The following theorem shows that the linear CG finds the minimizer of \( \phi(x) \) in at most \( n \) steps:

**Theorem 9.1.** For any \( x_0 \in \mathbb{R}^n \), the sequence \( \{x_k\} \) generated by

\[
x_{k+1} = x_k + \alpha_k p_k,
\]

where \( \{p_0, p_1, \ldots, p_{n-1}\} \) is a set of nonzero conjugate vectors wrt a symmetric positive definite matrix \( A \), and \( \alpha_k \) is the one dimensional minimizer of \( \phi \) along \( x_k + \alpha_k p_k \) given by,

\[
\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}.
\]

Then the sequence converges to the solution of \( Ax = b \) in at most \( n \) steps.

**Proof.** Since the \( n \) vectors in \( \{p_i\}_{i=1}^{n-1} \) are linearly independent, they span \( \mathbb{R}^n \) and therefore we can write

\[
x^* - x_0 = \sum_{i=0}^{n-1} c_i p_i
\]

for some scalars \( c_0, c_1, \ldots, c_{n-1} \). Then for any \( 0 \leq k \leq n-1 \),

\[
p_k^T A (x^* - x_0) = c_k p_k^T A p_k,
\]

due to conjugacy, leading to

\[
c_k = \frac{p_k^T A (x^* - x_0)}{p_k^T A p_k}, \quad \forall k = 0, 1, \ldots, n-1. \tag{9.2}
\]

On the other hand, from the recursive generation of \( x_k \), we have

\[
x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \cdots + \alpha_{k-1} p_{k-1}. \tag{9.3}
\]

From conjugacy, it implies that

\[
p_k^T A (x_k - x_0) = 0.
\]

From the above results and the definition of \( r_k \), we see that

\[
p_k^T A (x^* - x_0) = p_k^T A (x^* - x_k) = p_k^T (b - Ax_k) = -p_k^T r_k. \tag{9.4}
\]

Comparing (9.2) and (9.4), we have \( c_k = \alpha_k \). This implies from (9.1) and (9.3) that,

\[
x^* = x_0 + \sum_{i=0}^{n-1} \alpha_i p_i = x_n.
\]

\[\square\]
9.1. LINEAR CONJUGATE GRADIENT METHOD

9.1.1 Connection to Coordinate Descent

Consider a minimization of strictly convex quadratic function,

$$\min_{x \in \mathbb{R}^n} \phi(x) := \frac{1}{2} x^T A x - b^T x$$

where $A$ is symmetric positive definite.

If $A$ is diagonal, the contour of $\phi$ aligns with the standard coordinates. If that is the case, successive minimization along the directions $e_1, e_2, \ldots, e_n$ finds the minimizer. When $A$ is not diagonal, $\phi$ is typically not aligned with the standard coordinates, and the coordinate-descent approach may not find a solution in $n$ iterations.

Conjugate gradient can be seen as a coordinate descent method operating in a transformed space. Define a matrix $S$ as

$$S = [p_0 \ p_1 \ \cdots \ p_{n-1}] \in \mathbb{R}^{n \times n}$$

where $\{p_0, p_1, \ldots, p_{n-1}\}$ is the set of conjugate directions w.r.t. $A$. Denoting the transformed optimization variable by $\hat{x}$,

$$\hat{x} := S^{-1} x,$$

we can define a new objective function on the transformed domain,

$$\hat{\phi}(\hat{x}) := \phi(S \hat{x}) = \frac{1}{2} \hat{x}^T (S^T A S) \hat{x} - (S^T b)^T \hat{x}.$$

Check that the matrix $(S^T A S)$ is diagonal by conjugacy, and therefore we can find the minimizer of $\hat{\phi}$ by performing $n$ one-dimensional minimizations along the standard coordinate directions. In other words,

$$e_i = S^{-1} p_i,$$

and applying conjugate gradient to $\phi$ is equivalent to applying coordinate descent to $\hat{\phi}$.

9.1.2 Connection to Expanding Subspace Minimization

Another interesting property of linear CG is that after $k$ steps, it minimizes the objective function $\phi$ on the subspace spanned by $\{p_0, p_1, \ldots, p_{k-1}\}$.

We use a relation that is easily verifiable,

$$r_{k+1} = r_k + \alpha_k A p_k.$$  \hspace{1cm} (9.5)

**Theorem 9.2** (Expanding Subspace Minimization). Let $x_0 \in \mathbb{R}^n$ be a starting point and suppose that $\{x_k\}$ is generated by

$$x_{i+1} = x_i + \alpha_i p_i, \quad \alpha_i = -\frac{r_i^T p_i}{p_i^T A p_i}, \quad r_i = A x_i - b,$$

for conjugate directions $p_0, p_1, \ldots, p_{k-1}$. Then,

$$r_k^T p_i = 0, \text{ for } i = 0, 1, \ldots, k-1.$$  \hspace{1cm} (9.6)
and $x_k$ is the minimizer of $\phi(x)$ over the set

\[ \{ x : x = x_0 + \text{span}\{p_0, p_1, \ldots, p_{k-1}\} \}. \quad (9.7) \]

**Proof.** We first show a point $\tilde{x}$ that minimizes $\phi$ over (9.7) if and only if $r(\tilde{x})^T p_i = 0$ for all $i = 0, 1, \ldots, k-1$. For $c = (c_0, \ldots, c_{k-1})^T$, define $h(c) = \phi(x_0 + Sc)$ for $S = [p_0 \cdots p_{k-1}]$. Since $h(c)$ is strictly convex, it has a unique minimizer $c^*$ satisfying

\[ \frac{\partial h(c^*)}{\partial c_i} = 0, \quad i = 0, 1, \ldots, k-1. \]

This implies that

\[ \nabla \phi(x_0 + c_0 p_0 + \cdots + c_{k-1} p_{k-1})^T p_i = 0, \quad i = 0, 1, \ldots, k-1. \]

Since $\nabla \phi(x) = r(x)$, this shows the claim for $\tilde{x} = x_0 + c_0 p_0 + \cdots + c_{k-1} p_{k-1}$.

To show that $x_k$ satisfies (9.6), we use induction. For the case $k = 1$, by the choice of $a_0$ we know that $x_1 = x_0 + a_0 p_0$ minimizes $\phi$ along $p_0$, and therefore $r_1^T p_0 = 0$. Suppose that $r_i^T p_0 = 0$ for $i = 0, 1, \ldots, k-2$. From (9.5), we have

\[ r_k = r_{k-1} + \alpha_{k-1} A p_{k-1}, \]

so that

\[ p_{k-1}^T r_k = p_{k-1}^T r_{k-1} + \alpha_{k-1} p_{k-1}^T A p_{k-1} = 0 \]

by the definition of $\alpha_{k-1}$. For $p_i$ with $i = 0, 1, \ldots, k-2$, we have

\[ p_i^T r_k = p_i^T r_{k-1} + \alpha_{k-1} p_i^T A p_{k-1} = 0 \]

due the the induction hypothesis and conjugacy. This shows that $r_k^T p_i = 0$ for $i = 0, 1, \ldots, k-1$, completing the proof.

This theorem tells that the residual $r_k$ is orthogonal to all previous search directions, which is an important property to be used later.

### 9.2 Generation of Conjugate Directions

So far, we have discussed the linear CG method with any (given) set of conjugate directions $\{p_0, p_1, \ldots, p_{n-1}\}$. For example, the set of eigenvectors of $A$ fits the requirements, although it would not be appealing since computing all eigenvectors requires $O(n^3)$ computation and $O(n^2)$ storage.

The practicality of CG comes from the fact that we can generate conjugate directions iteratively with $O(n^2)$ operations, without storing previous directions except for the last one. The idea is based on the fact that $p_k$ can be expressed as a linear combination of $r_k$ and $p_{k-1}$, i.e.,

\[ p_k = -r_k + \beta_k p_{k-1}, \]

where the scalar $\beta_k$ is chosen so that $p_k$ and $p_{k-1}$ will be conjugate wrt. $A$:

\[ p_{k-1}^T A p_k = 0 \]

\[ \Rightarrow p_{k-1} A (-r_k + \beta_k p_{k-1}) = 0 \]

\[ \Rightarrow \beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}. \]
9.2. GENERATION OF CONJUGATE DIRECTIONS

For the first direction $p_0$, we choose $p_0 = -\nabla \phi(x_0) = -r_0$. A preliminary version of the CG algorithm is shown in Algorithm 9.1.

**Algorithm 9.1: CG: Preliminary Version**

*Input:* $x_0, A, b$

$r_0 = Ax_0 - b$, $p_0 = -r_0$, $k = 0$;

**while** $r_k \neq 0$ **do**

\[
\alpha_k = -\frac{r_k^T r_k}{p_k^T A p_k};
\]

\[
x_{k+1} = x_k + \alpha_k p_k;
\]

\[
r_{k+1} = Ax_{k+1} - b;
\]

\[
\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k};
\]

\[
p_{k+1} = -r_{k+1} + \beta_{k+1} p_k;
\]

$k = k + 1$;

**end**

*Output:* $x^* = x_k$.

9.2.1 Conjugacy and the Krylov Subspace

Note that we have not yet shown that the directions $p_0, p_1, \ldots, p_{n-1}$ generated by Algorithms 9.1 are mutually conjugate wrt $A$. If so, then by Theorem 11.3, this algorithm will terminate in $n$ steps. The next theorem shows this property, along with two other interesting properties: (i) the residuals $r_i$ are mutually orthogonal, and (ii) each $p_k$ and $r_k$ is contained in the Krylov subspace of degree $k$ for $r_0$, defined by

\[ \mathcal{K}(r_0; k) = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}. \]

**Theorem 9.3** (Mutual Conjugacy). Suppose that $x_k$ is generated by Algorithm 9.1, which is not equal to the solution $x^*$. Then the following properties hold:

\[ r_k^T r_i = 0, \quad i = 0, 1, \ldots, k - 1, \]  
\[ \text{span}\{r_0, r_1, \ldots, r_k\} = \mathcal{K}(r_0; k), \]  
\[ \text{span}\{p_0, p_1, \ldots, p_k\} = \mathcal{K}(r_0; k), \]  
\[ p_k^T A p_i = 0, \quad i = 0, 1, \ldots, k - 1. \]

Therefore, $\{x_k\}$ converges to $x^*$ in at most $n$ steps.

**Proof.** The last three statements are proven by induction. At $k = 1$, all three statements are trivially true. Suppose that they are true for some $k$ (induction hypothesis). We show these three statements hold for $k + 1$.

From the induction hypothesis, we have

\[ r_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}, \quad p_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}. \]

Then

\[ A p_k \in \text{span}\{A r_0, A^2 r_0, \ldots, A^{k+1} r_0\}. \]
Using \( r_{k+1} = r_k + \alpha_k A p_k \), we see that
\[
 r_{k+1} \in \text{span}\{r_0, A r_0, \ldots, A^{k+1} r_0\}.
\]
Combining this with the induction hypothesis, we conclude that
\[
 \text{span}\{r_0, r_1, \ldots, r_k, r_{k+1}\} \subseteq \text{span}\{r_0, A r_0, \ldots, A^{k+1} r_0\}.
\]
To show the reverse inclusion, we start from the induction hypothesis which gives
\[
 A^k r_0 \in \text{span}\{p_0, p_1, \ldots, p_k\}
\]
and therefore
\[
 A^{k+1} r_0 \in \text{span}\{A p_0, A p_1, \ldots, A p_k\}.
\]
From \( r_{i+1} = r_i + \alpha_i A p_i \), we have \( A p_i = (r_{i+1} - r_i)/\alpha_i \) for \( i = 0, 1, \ldots, k \), and thus
\[
 A^{k+1} r_0 \in \text{span}\{r_0, r_1, \ldots, r_{k+1}\}.
\]
Combining this with the induction hypothesis, we conclude that
\[
 \text{span}\{r_0, A r_0, \ldots, A^k r_0, A^{k+1} r_0\} \subseteq \text{span}\{r_0, r_1, \ldots, r_{k+1}\}.
\]
The above two results proves (9.9).

Similarly, (9.10) can be shown for \( k + 1 \) as follows,
\[
\begin{align*}
\text{span}\{p_0, p_1, \ldots, p_k, p_{k+1}\} &= \text{span}\{p_0, p_1, \ldots, p_k, r_{k+1}\} \\
&= \text{span}\{r_0, A r_0, \ldots, A^k r_0, r_{k+1}\} \quad \text{(induction hyp for (9.10) at } k) \\
&= \text{span}\{r_0, r_1, \ldots, r_k, r_{k+1}\} \quad \text{(induction hyp for (9.9) at } k) \\
&= \text{span}\{r_0, A r_0, \ldots, A^k r_0, A^{k+1} r_0\} \quad \text{(9.9) for } k + 1).
\end{align*}
\]
Next, we show the conjugacy (9.11). From \( p_{k+1} = -r_{k+1} + \beta_{k+1} p_k \), multiplying by \( A p_i \), \( i = 0, 1, \ldots, k \) gives
\[
p_{k+1}^T A p_i = -r_{k+1}^T A p_i + \beta_{k+1} p_k^T A p_i. \tag{9.12}
\]
When \( i = k \), then \( p_{k+1}^T A p_k = 0 \) from the definition of \( \beta_{k+1} \). For \( i < k \), we first note that from the induction hypothesis, \( p_0, p_1, \ldots, p_k \) are conjugate, and by Theorem 9.2 (expanding subspace minimization),
\[
r_{k+1}^T p_i = 0, \quad i = 0, 1, \ldots, k.
\]
Also, from (9.10), for \( i = 0, 1, \ldots, k-1 \) we have,
\[
 A p_i \in \text{span}\{r_0, A r_0, \ldots, A^i r_0\} = \text{span}\{A^i r_0, A^{i+1} r_0\} \subset \text{span}\{p_0, p_1, \ldots, p_i+1\}.
\]
From the above two results, we deduce that
\[
r_{k+1}^T A p_i = 0, \quad i = 0, 1, \ldots, k-1.
\]
Therefore, the first term in (9.12) becomes zero, and the second term is also zero due to the induction hypothesis. Therefore (9.11) holds for all \( k \).

Finally (without induction), for (9.8), we first note from
\[
p_i = -r_i + \beta_i p_{i-1}
\]
that \( r_i \in \text{span}\{p_i, p_{i-1}\} \) for all \( i = 1, \ldots, k - 1 \), and \( r_0 \in \text{span}\{p_0\} \) by our initialization. Since \( r_i^T p_i = 0 \) for all \( i = 0, 1, \ldots, k - 1 \) for any \( k = 1, 2, \ldots, n - 1 \) from Theorem 9.2, we conclude that \( r_i^T r_i = 0 \) for \( i = 0, 1, \ldots, k - 1 \).

Note that the proof does require that \( p_0 = -r_0 \), the steepest descent direction.
9.3 The Conjugate Gradient Algorithm

Now we can make Algorithm 9.1 more economical. First, from the mutual conjugacy theorem (Theorem 9.3) we know that the generated directions \( p_0, p_1, \ldots, p_{n-1} \) are conjugate wrt \( A \), and therefore by the expanding subspace minimization theorem (Theorem 9.2) we have

\[
r_k^T p_i = 0, \quad i = 0, 1, \ldots, k - 1.
\]

This allows us to simplify \( \alpha_k \): together with \( p_k = -r_k + \beta_k p_{k-1} \),

\[
\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} = -\frac{r_k^T r_k}{p_k^T A p_k}.
\]

Next, with \( r_{k+1} = r_k + \alpha_k A p_k \) and \( r_k^T r_i = 0 \) for \( i = 0, \ldots, k - 1 \) from (9.8), we get

\[
\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \frac{r_{k+1}^T (r_{k+1} - r_k)}{p_k^T A p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.
\]

We also use \( r_{k+1} = r_k + \alpha_k A p_k \) for computing \( r_{k+1} \) instead. The resulting algorithm is shown in Algorithm 9.3. What are the savings?

### Algorithm 9.2: CG: Preliminary

**Input:** \( x_0, A, b \);

\( r_0 = Ax_0 - b, \; p_0 = -r_0, \; k = 0; \)

**while** \( r_k \neq 0 \) **do**

\[
\begin{align*}
\alpha_k &= -\frac{r_k^T p_k}{p_k^T A p_k}; \\
x_{k+1} &= x_k + \alpha_k p_k; \\
r_{k+1} &= Ax_{k+1} - b; \\
\beta_{k+1} &= \frac{r_{k+1}^T A p_k}{p_k^T A p_k}; \\
p_{k+1} &= -r_{k+1} + \beta_{k+1} p_k; \\
k &= k + 1;
\end{align*}
\]

**end**

**Output:** \( x^* = x_k \).

### Algorithm 9.3: CG

**Input:** \( x_0, A, b \);

\( r_0 = Ax_0 - b, \; p_0 = -r_0, \; k = 0; \)

**while** \( r_k \neq 0 \) **do**

\[
\begin{align*}
\alpha_k &= \frac{r_k^T r_k}{p_k^T A p_k}; \\
x_{k+1} &= x_k + \alpha_k p_k; \\
r_{k+1} &= r_k + \alpha_k A p_k; \\
\beta_{k+1} &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}; \\
p_{k+1} &= -r_{k+1} + \beta_{k+1} p_k; \\
k &= k + 1;
\end{align*}
\]

**end**

**Output:** \( x^* = x_k \).

9.4 Rate of Convergence

In Theorem 9.1, we showed that the CG algorithm finds the solution in at most \( n \) steps. In fact, when the eigenvalues of \( A \) are clustered, it can converges much faster than \( n \) steps.

To see this property, we first note from Theorem 9.3 that

\[
x_{k+1} = x_0 + c_0 p_0 + \cdots + c_k p_k = x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0,
\]

with some proper coefficients \( c_i \) and \( \gamma_i \). We define \( P^*_k(A) \) to be a polynomial of degree \( k \) with the coefficients \( \gamma_0, \ldots, \gamma_k \), so that

\[
P^*_k(A) = \gamma_0 I + \gamma_1 A + \cdots + \gamma_k A^k, \quad x_{k+1} = x_0 + P^*_k(A) r_0.
\]
We aim to show that among all possible methods whose first \( k \) steps are restricted to the Krylov space \( \mathcal{K}(r_0; k) \), Algorithm 9.3 performs the best for minimizing the distance to the solution after \( k \) steps, where the distance is measured by the weighted Euclidean norm:

\[
\|z\|_A^2 = z^T A z.
\]

Recall that since \( x^* \) is the minimizer of \( \phi(x) = \frac{1}{2} x^T A x - b \), we have

\[
\frac{1}{2} \|x - x^*\|_A^2 = \frac{1}{2} (x - x^*)^T A (x - x^*) = \frac{1}{2} x^T A x - b^T x + x^T (b - Ax^*) + \frac{1}{2} (x^*)^T A x^* = \phi(x) - \phi(x^*) + (b^T x^* - b^T x + x^T (A - b^T)x^*) = \phi(x) - \phi(x^*).
\]

Due to Theorem 9.2, \( x_{k+1} \) minimizes \( \phi(x) \), and hence \( \|x - x^*\|_A^2 \), over the subspace \( x_0 + \text{span}\{p_0, \ldots, p_k\} = x_0 + \text{span}\{r_0, Ar_0, \ldots, A^k r_0\} \). In other words the polynomial \( P_k^*(A) \) solves the following problem over all possible polynomials of degree \( k \),

\[
\min_{P_k} \|x_0 + P_k(A) r_0 - x^*\|_A.
\]

From \( r_0 = Ax_0 - b = A(x_0 - x^*) \), we have

\[
x_{k+1} - x^* = x_0 + P_k^*(A) r_0 - x^* = [I + P_k^*(A)A] (x_0 - x^*). \tag{9.13}
\]

On the other hand, we can write

\[
A = \sum_{i=1}^n \lambda_i v_i v_i^T
\]

for its eigenvalues \( 0 < \lambda_1 \leq \cdots \leq \lambda_n \) and the associated orthonormal eigenvectors \( v_1, v_2, \ldots, v_n \). Since the eigenvectors span \( \mathbb{R}^n \),

\[
x_0 - x^* = \sum_{i=1}^n \xi_i v_i \tag{9.14}
\]

for some coefficients \( \xi_i \). Any eigenvector of \( A \) is also an eigenvector of \( P_k(A) \), so that

\[
P_k(A) v_i = P_k(\lambda_i) v_i \quad i = 1, 2, \ldots, n.
\]

Plugging-in (9.14) into (9.13), we have

\[
x_{k+1} - x^* = \sum_{i=1}^n [1 + \lambda_i P_k^*(\lambda_i)] \xi_i v_i
\]

Using \( \|z\|_A^2 = z^T A z = \sum_{i=1}^n \lambda_i (v_i^T z)^2 \),

\[
\|x_{k+1} - x^*\|_A^2 = \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k^*(\lambda_i)]^2 \xi_i^2.
\]
The polynomial $P_k^*$ generated by the CG algorithm minimizes the LHS, and therefore,
\[
\|x_{k+1} - x^*\|^2_A = \min_{P_k} \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2
\]
\[
\leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \left(\sum_{j=1}^n \lambda_j \xi_j^2\right)
\]
\[
= \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|^2_A.
\]
(9.15)

We use the above property to show the following theorems.

**Theorem 9.4.** If $A$ has only $r$ distinct eigenvalues, then the CG algorithm will terminate at the solution in at most $r$ iterations.

**Proof.** Suppose that the eigenvalues $\lambda_i$, $i = 1, 2, \ldots, n$, of $A$ take on the $r$ distinct values $\tau_1 < \tau_2 < \cdots < \tau_r$. We define a polynomial $Q_r(\lambda)$ by
\[
Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r} (\lambda - \tau_1)(\lambda - \tau_2)\cdots(\lambda - \tau_r).
\]
Then $Q_r(\lambda_i) = 0$ for $i = 1, 2, \ldots, n$, and $Q_r(0) = 1$, and therefore $Q_r(\lambda) - 1$ is a degree-$r$ polynomial with a root at $\lambda = 0$. That is, if we define the function
\[
\tilde{P}_{r-1}(\lambda) = (Q_r(\lambda) - 1)/\lambda,
\]
then it is a polynomial of degree $r - 1$. Taking $k = r - 1$ in (9.15), we get
\[
\|x_r - x^*\|^2_A \leq \min_{P_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} \|1 + \lambda_i \tilde{P}_{r-1}(\lambda_i)\|^2 = \max_{1 \leq i \leq n} Q_r^2(\lambda_i) = 0.
\]
That is, $x_r = x^*$.

By similar arguments, we can show the following results.

**Theorem 9.5.** Let $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $A$. Then,
\[
\|x_{k+1} - x^*\|^2_A \leq \frac{\lambda_{n-k} - \lambda_2}{\lambda_{n-k} + \lambda_2} \|x_0 - x^*\|^2_A.
\]

Or, in a more approximate form, we have
\[
\|x_k - x^*\|^2_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|x_0 - x^*\|^2_A.
\]
for $\kappa(A) = \|A\|_2\|A^{-1}\|_2 = \lambda_n/\lambda_1$.

To understand this result, suppose that $A$ has $m$ large eigenvalues, where the rest $n - m$ smaller eigenvalues are clustered around 1. Defining $\epsilon = \lambda_{n-m} - \lambda_1$, this theorem tells that after $m + 1$ steps we have
\[
\|x_{m+1} - x^*\|^2_A \approx \frac{\epsilon}{2} \|x_0 - x^*\|^2_A,
\]
so that for small $\epsilon$, the CG iterate $x_{m+1}$ will provide a good estimate of the solution.
CHAPTER 9. CONJUGATE GRADIENT METHOD

9.5 Nonlinear Conjugate Gradient Method

The idea of conjugate gradient can be extended for minimizing a nonlinear objective function \( f \), instead of a strictly convex objective \( \phi \). We discuss nonlinear CG only briefly in this lecture, but it has been well-studied and quite successful in practical applications.

9.5.1 Fletcher-Reeves Algorithm

The linear CG algorithm 9.3 can be extended by making two simple changes: (i) using line search to find a stepsize \( \alpha_k \), and (ii) replacing the residual \( r_k = \nabla \phi(x_k) \) by \( \nabla f(x_k) \).

Algorithm 9.4: FR

Input: \( x_0, f, \nabla f \);
\( p_0 = -\nabla f(x_0) \), \( k = 0 \);
while \( \nabla f(x_k) \neq 0 \) do
  Compute \( \alpha_k \) by line search;
  \( x_{k+1} = x_k + \alpha_k p_k \);
  \( \beta_{FR}^{k+1} = \nabla f(x_{k+1})^T \nabla f(x_k) / \nabla f(x_k)^T \nabla f(x_k) \);
  \( p_{k+1} = -\nabla f(x_{k+1}) + \beta_{FR}^{k+1} p_k \);
  \( k = k + 1 \);
end
Output: \( x^* = x_k \).

If \( f \) is a strictly convex quadratic function and \( \alpha_k \) is computed with an exact line search, then this algorithm reverts to the linear CG algorithm. Algorithm 9.4 is appealing for solving large-scale problems, since it does not involve any matrix operation.

From the update rule of \( p_k \), we have

\[
\nabla f(x_k)^T p_k = -\|\nabla f(x_k)\|^2 + \beta_{FR}^{k+1} \nabla f(x_k)^T p_{k-1}
\]

When the line search is exact so that \( \alpha_{k-1} \) is a local minimizer of \( f \) along \( p_{k-1} \), we have \( \nabla f(x_k)^T p_{k-1} = 0 \) and therefore \( p_k \) is a descent direction. Otherwise, the line search need to satisfy the strong Wolfe conditions with \( 0 < c_1 < c_2 < \frac{1}{2} \) (we skip the proof).

9.5.2 Other Nonlinear CG Methods

Polak-Rivière (PR)

\[
\beta_{PR}^{k+1} = \frac{\nabla f(x_{k+1})^T (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla f(x_k)^T \nabla f(x_k)}
\]

PR is identical to FR when when \( f \) is a strongly convex quadratic function and the line search is exact. However, PR tends to be more robust than FR (see more discussion below). Surprisingly, PR with strong Wolfe LS does not guarantee that \( p_k \) becomes a descent direction. Therefore we use instead

\[
(\text{PR+}) \quad \beta_{k+1}^+ = \max\{\beta_{k+1}^{PR}, 0\},
\]


and use a modification of strong Wolfe conditions to have the guarantee.

On the other hand, under the assumption that $f$ is strongly convex and LS is exact, then it can be shown for PR that

$$\lim_{k \to \infty} \|\nabla f(x_k)\|_2 = 0.$$ 

**FR-PR** In fact, it is possible to show global convergence for any parameter $\beta_k$ such that

$$|\beta_k| \leq \beta^F_R, \quad \forall k \geq 2.$$ 

FR-PR is a modification of PR motivated by this, which chooses $\beta_k$ for $k \geq 2$ as follows

$$\beta_k = \begin{cases} 
-\beta^F_R & \text{if } \beta^P_R < -\beta^F_R \\
\beta^P_R & \text{if } |\beta^P_R| \leq \beta^F_R \\
\beta^F_R & \text{if } \beta^P_R > \beta^F_R
\end{cases}$$

**Recent Variants** There are attractive recent variants of nonlinear CG, with global convergence results without requiring modifications of line search based on Wolfe conditions:

(Dai & Yuan, SIOPT, 1999) $\beta_{k+1} = \frac{\|\nabla f(x_{k+1})\|^2}{(\nabla f(x_{k+1}) - \nabla f(x_k))^T p_k}$

(Hager & Zhang, SIOPT, 2005) $\beta_{k+1} = \left(y_k - 2p_k \frac{\|y_k\|^2}{y_k^T p_k}\right)^T \frac{1}{y_k^T p_k} \nabla f_{k+1},$

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

### 9.5.3 Convergence

**Theorem 9.6** (Global Convergence of FR). Suppose that the level set $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ is bounded, and that $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuously differentiable in some open neighborhood of $\mathcal{L}$. Algorithm 9.4 with a line search satisfying the strong Wolfe conditions with $0 < c_1 < c_2 < \frac{1}{2}$ satisfies that

$$\lim_{k \to \infty} \|\nabla f(x_k)\|_2 = 0.$$ 

Although FR has global convergence, it has a weakness: it can be shown that when $p_k$ is a bad direction with a very small stepsize, in other words $\nabla f(x_k) \approx \nabla f(x_{k+1}) (\beta^F_R \approx 1)$ then the next search direction and stepsize are also likely to be bad, i.e. $p_{k+1} \approx p_k$.

In comparison, PR avoids this problem since if $\nabla f(x_k) \approx \nabla f(x_{k+1}) (\beta^P_R \approx 0)$, then $p_{k+1} \approx -\nabla f(x_{k+1})$. Similar arguments hold for PR+ and FR-PR. Despite this, PR does not have a global convergence result: in fact, it can cycle infinitely even with an exact linesearch.
9.5.4 Convergence Rate

Most of the convergence rate results of nonlinear CG assumes that the line search is exact – in which case, the convergence rate is linear. With restarting ($\beta_k = 0$) in every $n$ steps, $n$-step quadratic convergence has been also shown, i.e.,

$$\|x_{k+n} - x^*\| = O(\|x_k - x^*\|^2).$$

Exercises

9.1. [5] Consider nonzero $n$-dimensional vectors $p_0, p_1, \ldots, p_{n-1}$ that are conjugate with respect to a symmetric and positive definite matrix $A \in \mathbb{R}^{n \times n}$. Show that these vectors are linearly independent.

9.2. (Truncated Newton Method) Consider a variant of Newton’s method solving

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable strictly convex function. At $k$th iteration, we use the linear CG algorithm to find $p_k$ solving the following system,

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

Then a line search procedure should follow to find an appropriate stepsize $\alpha_k$: if $x_k$ is near a solution $x^*$ and $p_k$ is accurate, then $\alpha_k = 1$ will be admissible, similarly to quasi-Newton. The update will be $x_{k+1} = x_k + \alpha_k p_k$.

The CG procedure should be stopped after $\text{maxCGiter}$ steps or

$$\frac{\|\nabla^2 f(x_k)p_k + \nabla f(x_k)\|}{\|\nabla f(x_k)\|} \leq \epsilon_{\text{CG}}$$

for a threshold $\epsilon_{\text{CG}} > 0$. Start the CG iterations from the zero vector. (Be careful of the notations in Algorithm 9.3.)

(a) Implement this algorithm and compare its performance to pure Newton and quasi-Newton (BFGS), on the penalized logistic regression problem with the MNIST data (use $\lambda = 0.01$). For CG, use $\epsilon_{\text{CG}} = 10^{-4}$, and try $\text{maxCGiter} = 5, 10, 25$. Plot information of iterates of these total five cases: (i) iteration counter vs. $\|\nabla f(x_k)\|_\infty$, and (ii) elapsed time vs. $\|\nabla f(x_k)\|_\infty$. Make only the y-axes in logarithmic scale.

- Use the optimality check $\|\nabla f(x_k)\|_\infty \leq 10^{-7}$ and $\text{maxiter} = 50$. For the rest, use the same parameter values from the previous homework.
- Start from your previous quasi-Newton method implementation.
- [5] Add a .m function for Hessian computation, and use its pointer.
- [5] Implement Algorithm 9.3 as a separate routine.

(b) [5]* Show that if we start the CG iterations from the zero vector as here, it always outputs a descent direction.
Chapter 10

Constrained Optimization

From this chapter, we will consider constrained minimization problems as described as follows in general,

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f(x) \\
\text{subject to} & \quad c_i(x) = 0, \ i \in \mathcal{E}, \\
& \quad c_i(x) \geq 0, \ i \in \mathcal{I}.
\end{align*}
\]

where \( f \) and \( c_i \)'s are all smooth, real-valued functions on a subset of \( \mathbb{R}^n \), and \( \mathcal{E} \) and \( \mathcal{I} \) are two finite sets of indices for equality and inequality constraints, resp.

**Feasible Set**  We define the set of all feasible points, or the feasible set \( \Omega \), as follows

\[
\Omega := \{ x \in \mathbb{R}^n : c_i(x) = 0, \ i \in \mathcal{E}, \ c_i(x) \geq 0, \ i \in \mathcal{I} \}.
\]

**Active Set**  The active set \( \mathcal{A}(x) \) at any feasible point \( x \) is,

\[
\mathcal{A}(x) := \mathcal{E} \cup \{ i \in \mathcal{I} : c_i(x) = 0 \}.
\]

At a feasible point \( x \), an inequality constraint \( i \in \mathcal{I} \) is called active if \( c_i(x) = 0 \), and inactive if \( c_i(x) > 0 \).

**Types of Solutions**  The types of minimizers are redefined similarly to those of unconstrained cases, but for the feasible sets in obvious senses.

- \( x^* \) is a local minimizer if \( x^* \in \Omega \) and there is a neighborhood \( \mathcal{N} \) of \( x^* \) such that \( f(x^*) \leq f(x) \) for all \( x \in \mathcal{N} \cap \Omega \).

- \( x^* \) is a strict local minimizer if \( x^* \in \Omega \) and there is neighborhood \( \mathcal{N} \) of \( x^* \) such that \( f(x^*) < f(x) \) for all \( x \in \mathcal{N} \cap \Omega \) with \( x \neq x^* \).

- \( x^* \) is an isolated local minimizer if \( x^* \in \Omega \) and there is a neighborhood \( \mathcal{N} \) of \( x^* \) such that \( x^* \) is the only local minimizer in \( \mathcal{N} \cap \Omega \).
10.1 First-Order Optimality Conditions

Consider the constrained minimization of a continuously differentiable function \( f \),

\[
\min_{x \in \Omega} f(x),
\]

where the feasible set

\[
\Omega = \{ x \in \mathbb{R}^n : c_i(x) = 0 \ \forall i \in \mathcal{E}, c_i(x) \geq 0 \ \forall i \in \mathcal{I} \}
\]

is nonempty and closed.

The following theorem provides a fundamental first-order optimality condition, which we refer to as the minimum principle in this lecture.

**Theorem 10.1 (Minimum Principle).** Suppose that \( \Omega \) is a convex set. If \( x^* \) is a local minimizer of \( f \) over \( \Omega \), then

\[
\nabla f(x^*)^T (x - x^*) \geq 0 \ \forall x \in \Omega.
\]  

(10.1)

Moreover, if \( f \) is a convex function over \( \Omega \), then the above condition is also sufficient for \( x^* \) to be a global minimizer of \( f \) over \( \Omega \).

**Proof.** Suppose for contradiction that \( \nabla f(x^*)^T (x - x^*) < 0 \) for some \( x \in \Omega \). By the mean value theorem (Theorem 6.1) applied to a function \( g(\epsilon) = f(x^* + \epsilon(x - x^*)) \) on \((0, 1)\), there exists a scalar \( \alpha \in (0, \epsilon) \) such that

\[
f(x^* + \epsilon(x - x^*)) - f(x^*) = \epsilon \nabla f(x^* + \alpha(x - x^*))^T (x - x^*).
\]

Since \( \nabla f \) is continuous, for all sufficiently small \( \epsilon > 0 \) we have \( \nabla f(x^* + \alpha(x - x^*))^T (x - x^*) < 0 \) and therefore \( f(x^* + \epsilon(x - x^*)) < f(x^*) \). The vector \( x^* + \epsilon(x - x^*) \) is feasible for all \( \epsilon \in (0, 1) \) since \( \Omega \) is convex. This contradicts the fact that \( x^* \) is a local minimizer.

If \( f \) is convex, then we have

\[
f(x) \geq f(x^*) + \nabla f(x^*)^T (x - x^*), \forall x \in \Omega.
\]

If \( \nabla f(x^*)^T (x - x^*) \geq 0 \) holds for all \( x \in \Omega \), then \( x^* \) is a global minimizer of \( f \) over \( \Omega \).

**Stationary Point** A vector \( x^* \) satisfying the condition (10.1) is called a stationary point. Check that this definition is consistent with the unconstrained case where \( \Omega = \mathbb{R}^n \).

**Ex.** Consider a positive orthant constraint

\[
\Omega = \{ x \in \mathbb{R}^n : x_i \geq 0, i = 1, 2, \ldots, n \}.
\]

From the minimum principle, if \( x^* \) is a local minimizer then

\[
\sum_{i=1}^n \frac{\partial f(x^*)}{\partial x_i} (x_i - x_i^*) \geq 0, \ \forall x = (x_1, x_2, \ldots, x_n)^T \geq 0.
\]  

(10.2)
Let us choose an index \( i \), and let \( x_j = x_j^* \) for \( j \neq i \) and \( x_i = x_i^* + 1 \). Then we obtain,

\[
\frac{\partial f(x^*)}{\partial x_i} \geq 0, \ \forall i. \tag{10.3}
\]

If \( x_i^* > 0 \), letting \( x_j = x_j^* \) for \( j \neq i \) and \( x_i = \frac{1}{2} x_i^* \) gives

\[
\frac{\partial f(x^*)}{\partial x_i} = 0, \ \text{if} \ x_i^* > 0. \tag{10.4}
\]

If \( f \) is convex, then (10.3) and (10.4) imply (10.2) and therefore these conditions are also sufficient by Theorem 10.1.

### 10.2 Tangent Cone

Given a feasible point \( x \in \Omega \), \( \{z_k\} \) is a feasible sequence approaching \( x \) if \( z_k \in \Omega \) for all sufficiently large \( k \) and \( z_k \to x \). A tangent is a limiting direction of a feasible sequence.

**Definition 10.1.** The vector \( d \) is a tangent to \( \Omega \) at a point \( x \) if there exists a feasible sequence \( \{z_k\} \) approaching \( x \) and a sequence of positive scalars \( \{t_k\} \) with \( t_k \to 0 \) such that

\[
\lim_{k \to \infty} \frac{z_k - x}{t_k} = d.
\]

The set of all tangents to \( \Omega \) at \( x^* \) is called the tangent cone, denoted by \( T_\Omega(x^*) \).

- \( T_\Omega(x^*) \) is indeed a cone\(^1\).
- \( 0 \in T_\Omega(x^*) \).
- \( T_\Omega(x^*) \) is closed.
- \( T_\Omega(x^*) \) is the closure of the set of feasible directions \( \{x - x^* : x \in \Omega\} \).

**Definition 10.2.** For a cone \( K \subseteq \mathbb{R}^n \), the polar cone of \( K \) is defined by

\[
K^\circ = \{d \in \mathbb{R}^n : d^T x \leq 0, \forall x \in K\}.
\]

- \( K^\circ \) is indeed a cone.
- (Polar Cone Theorem) For a nonempty closed convex cone \( C \), \( (C^\circ)^\circ = C \).

**Definition 10.3.** The normal cone to the set \( \Omega \) at a point \( x \in \Omega \) is defined as

\[
N_\Omega(x^*) := T_\Omega^\circ(x^*).
\]

---

\(^1\) A set \( C \subseteq \mathbb{R}^n \) is called a cone if \( x \in C \), then \( \alpha x \in C \) for all \( \alpha > 0 \).
We can restate the minimum principle in Theorem 10.1 using the tangent cone.

**Theorem 10.2 (FONC).** Suppose that \( x^* \) is a local minimizer of a continuously differentiable function \( f \) over a nonempty convex feasible set \( \Omega \). Then
\[
\nabla f(x^*)^T d \geq 0, \quad \forall d \in T_\Omega(x^*).
\]

or, equivalently,
\[
-\nabla f(x^*) \in N_\Omega(x^*).
\]

**Proof.** Suppose that \( d \in T_\Omega(x^*) \). Then there exists a sequences \( \{z_k\} \subset \Omega \) and \( \{t_k\} \) such that
\[
\frac{z_k - x^*}{t_k} = d + o(t_k).
\]

By the mean value theorem, there exists \( \alpha \in (0, 1) \) such that
\[
f(z_k) - f(x^*) = \nabla f(\tilde{x}_k)^T (z_k - x^*), \quad \tilde{x}_k = x^* + \alpha (z_k - x^*).
\]

Therefore,
\[
f(z_k) - f(x^*) = \nabla t_k (f(\tilde{x}_k)^T \tilde{d}_k), \quad \tilde{d}_k = d + o(t_k).
\]

For contradiction, suppose that there exists \( d \in T_\Omega(x^*) \) such that \( \nabla f(x^*)^T d < 0 \). Then due to the continuity of \( \nabla f \) and the facts that \( \tilde{x}_k \to x^* \) and \( \tilde{d}_k \to d \) as \( k \to \infty \), for sufficiently large \( k \) we have \( \nabla f(\tilde{x}_k)^T \tilde{d}_k < 0 \). This implies that \( f(z_k) < f(x^*) \), which contradicts the assumption that \( x^* \) is a local minimizer.

Q.E.D.

**Ex. 1**
\[
\min f(x_1, x_2) = x_1 + x_2 \quad \text{s.t.} \quad c_1(x_1, x_2) = x_1^2 + x_2^2 - 2 = 0.
\]
Consider \( \nabla f(x) \) and \( \nabla c_1(x) \) at \((-1, -1)^T, (-1, 1)^T, (1, -1)^T, \) and \((-1, -1)^T\). The solution is obviously \( x^* = (-1, -1)^T \).

We see that with \( \lambda_1^* = -1/2 \),
\[
\nabla f(x^*) = \lambda_1^* \nabla c_1(x^*).
\]

This can be derived as follows. Suppose that a feasible \( x \) is not a minimizer and therefore we find a step \( s \) which brings reduction in \( f \). We require \( s \) satisfies the constraint,
\[
0 = c_1(x + s) \Rightarrow c_1(x) + \nabla c_1(x)^T s = \nabla c_1(x)^T s.
\]

To produce a decrease in \( f \), we require that
\[
0 > f(x + s) - f(x) \approx \nabla f(x)^T s.
\]

It is easy to check that if \( \nabla f(x) \) and \( \nabla c_1(x) \) are parallel, then there exists no such direction \( s \) that satisfies the above two conditions.

We introduce the Lagrangian function,
\[
\mathcal{L}(x; \lambda_1) = f(x) - \lambda_1 c_1(x),
\]
and the above condition can be stated that
\[ \nabla_x \mathcal{L}(x^*, \lambda^*_1) = 0. \]

This condition is necessary, but clearly not sufficient for \( x^* \) being a local minimizer since \( x = (1, 1)^T \) with \( \lambda_1 = 1/2 \) also satisfies the condition, which is a maximizer. Also, the sign of \( \lambda^*_1 \) can be changed, by using \( c_1(x) = 2 - x_1^2 - x_2^2 = 0 \) instead (then \( \lambda^*_1 = 1/2 \)).

**Ex. 2**
\[
\begin{align*}
\min_{x_1, x_2} & \quad f(x_1, x_2) = x_1 + x_2 \\
\text{s.t.} & \quad c_1(x_1, x_2) = 2 - x_1^2 - x_2^2 \geq 0.
\end{align*}
\]

Suppose that a feasible point \( x \) is not a minimizer. Then we should be able to find a step \( s \) so that
\[
0 \leq c_1(x + s) \approx c_1(x) + (\nabla c_1(x))^T s.
\]

We can have two cases:

**Case I.** \( x \) lies strictly inside of the circle, so that \( c_1(x) > 0 \). When \( \nabla f(x) \neq 0 \), we can find a step \( s \) that gives a decrease in the first order, i.e.
\[
s = -\alpha \nabla f(x)
\]
for some sufficiently small \( \alpha > 0 \).

**Case II.** \( x \) lies on the boundary of the circle, so that \( c_1(x) = 0 \). Then a step decreasing \( f \) must satisfy
\[
\nabla f(x)^T s < 0, \quad (\nabla c_1(x))^T s \geq 0.
\]
The intersection of the two regions specified by these (an open half-space and a closed half-space) is empty only when \( \nabla f(x) \) and \( \nabla c_1(x) \) point to the same direction, that is,
\[
\nabla f(x) = \lambda_1 \nabla c_1(x), \quad \text{for some } \lambda_1 \geq 0.
\]
In this case the sign of \( \lambda_1 \) matters.

When an \( x^* \) is optimal in cases I and II, its optimality can be stated with the Lagrangian function, i.e.,
\[
\nabla_x \mathcal{L}(x^*; \lambda^*_1) = 0 \quad \text{for some } \lambda^*_1 \geq 0
\]
with an extra condition that
\[
\lambda^*_1 c_1(x^*) = 0.
\]
This condition is known as a complementarity condition, which implies that the Lagrange multiplier \( \lambda_1 \) can be strictly positive only when the corresponding constraint \( c_1 \) is active. This condition plays an important role: it forces that \( \lambda^*_1 = 0 \) for case I, and \( \lambda^*_1 \geq 0 \) for case II.

**Ex 3.**
\[
\begin{align*}
\min_{x_1, x_2} & \quad f(x) = x_1 + x_2 \\
\text{s.t.} & \quad c_1(x) = 2 - x_1^2 - x_2^2 \geq 0, \quad c_2(x) = x_2 \geq 0.
\end{align*}
\]
The solution is \( x = (-\sqrt{2}, 0)^T \), at which both constraints are active. If \( x \) is not a minimizer, then we expect to find a step \( s \) such that
\[
\nabla c_i(x)^T s \geq 0, \forall i \in \mathcal{E}, \quad \nabla f(x)^T s < 0.
\]
Clearly, for \( x = (-\sqrt{2}, 0)^T \) there is no such \( s \).

In terms of Lagrangian, for \( \lambda = (\lambda_1, \lambda_2)^T \) we can define
\[
\mathcal{L}(x, \lambda) = f(x) - \lambda^T \begin{bmatrix} c_1(x) \\ c_2(x) \end{bmatrix},
\]
and the optimal variables satisfy
\[
\nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \text{ for some } \lambda^* \geq 0.
\]
The complementarity conditions are
\[
\lambda_1^* c_1(x^*) = 0, \quad \lambda_2^* c_2(x^*) = 0.
\]
When \( x^* = (-\sqrt{2}, 0)^T \), we have
\[
\nabla f(x^*) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \nabla c_1(x^*) = \begin{bmatrix} 2\sqrt{2} \\ 0 \end{bmatrix}, \quad \nabla c_2(x^*) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},
\]
so it follows that
\[
\lambda^* = \begin{bmatrix} 1/(2\sqrt{2}) \\ 1 \end{bmatrix} > 0.
\]

For another point \( x = (\sqrt{2}, 0)^T \), both constraints are active. But in this case we can find \( s = (-1, 0)^T \) that satisfies (10.5). In this case the condition \( \nabla_x \mathcal{L}(x, \lambda) = 0 \) is satisfied with \( \lambda = (-1/(2\sqrt{2}), 1)^T \), whose first component is negative.

For the point \( x = (1, 0)^T \), now only the constraint \( c_2 \) is active, and any small enough step \( s \) (satisfying \( c_2 \)) will continue to satisfy \( c_1(x + s) > 0 \). So we need to consider the behavior of only \( c_2 \) to check if \( s \) is a feasible descent step: if \( s \) is descent, then
\[
\nabla c_2(x)^T s \geq 0, \quad \nabla f(x)^T s < 0.
\]
It is easy to check that \( s = (-1/2, 1/4)^T \) is such a direction. The optimality conditions in Lagrangian do not hold: from complementarity, \( \lambda_1 = 0 \) and therefore \( 0 = \nabla_x \mathcal{L}(x, \lambda) = \nabla f(x) - \lambda_2 \nabla c_2(x) \), but no such \( \lambda_2 \) exists.

### 10.4 Linearized Feasible Directions

Given a feasible point \( x \), we define the set of linearized feasible directions \( \mathcal{F}(x) \) as follows,
\[
\mathcal{F}(x) := \left\{ d : \begin{array}{l}
d^T \nabla c_i(x) = 0, \forall i \in \mathcal{E} \\
d^T \nabla c_i(x) \geq 0, \forall i \in \mathcal{A}(x) \cap \mathcal{E} \end{array} \right\}
\]
10.4. LINEARIZED FEASIBLE DIRECTIONS

Properties:

- $\mathcal{F}(x)$ is a cone.
- $T_\Omega(x) \subseteq \mathcal{F}(x)$.

Proof. Given a feasible point $x$, we reorder the constraints so that the constraints $\mathcal{C}_i, i = 1, 2, \ldots, m$ are active at $x$. Consider sequences $\{z_k\}$ and $t_k = \|z_k - x\|_2$ (in this case $\|d\|_2 = 1$), so that

$$z_k = x + t_k d + o(t_k).$$

For $i \in \mathcal{E}$, using Taylor’s theorem, we have

$$0 = \frac{1}{t_k} c_i(z_k) = \frac{1}{t_k} [c_i(x) + t_k \nabla c_i(x)^T d + o(t_k\|d\|_2)] = \nabla c_i(x)^T d + o(t_k) \frac{t_k}{t_k}$$

Taking $k \to \infty$, we have $\nabla c_i(x)^T d = 0$.

For $i \in \mathcal{F} \cap \mathcal{A}(x)$, we have similarly that

$$0 \leq \frac{1}{t_k} c_i(x) = \frac{1}{t_k} [c_i(z_k) + t_k \nabla c_i(x)^T d + o(t_k)] = \nabla c_i(x)^T d + o(t_k) \frac{t_k}{t_k}.$$

And therefore $\nabla c_i(x)^T d \geq 0$ by taking $k \to \infty$. Collecting the results show that $d \in \mathcal{F}(x)$. \qed

- $T_\Omega(x)$ does not depend on algebraic specification of $\Omega$, but $\mathcal{F}(x)$ does.

Ex 1 (revisited)

$$\min_x f(x) = x_1 + x_2, \quad c_1(x) = x_1^2 + x_2^2 - 2 = 0.$$

For a non-optimal point $x = (-\sqrt{2}, 0)^T$, we find that

$$z_k = \begin{bmatrix} -\sqrt{2} - 1/k^2 \\ -1/k \end{bmatrix}.$$

and a tangent $d = (0, -1)$ with defining $t_k = \|z_k - x\|_2$. Another feasible sequence approaching the same $x$ is

$$z_k = \begin{bmatrix} -\sqrt{2} - 1/k^2 \\ 1/k \end{bmatrix}.$$
and the tangent corresponding to this sequence are \( d = (0, \alpha)^T, \alpha > 0 \). Together with the previous one, we conclude that the tangent cone at \( x = (-\sqrt{2}, 0)^T \) is
\[
T_{\Omega}(x) = \{(0, d_2) : d_2 \in \mathbb{R}\}.
\]
A vector \( d \) in the linearized feasible directions \( \mathcal{F}(x) \) should satisfy
\[
0 = \nabla c_1(x)^T d = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}^T \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = -2\sqrt{2}d_1.
\]
Therefore, \( \mathcal{F}(x) = \{(0, d_2)^T : d_2 \in \mathbb{R}\} \) and therefore \( T_{\Omega}(x) = \mathcal{F}(x) \) in this case.

Suppose that the feasible set is defined instead of the formula,
\[
\Omega = \{x : c_1(x) = 0\}, \quad c_1(x) = (x_1^2 + x_2^2 - 2)^2 = 0.
\]
Here the set \( \Omega \) is the same as before, but its algebraic description has changed. a
vector \( d \in \mathcal{F}(x) \) if,
\[
0 = \nabla c_1(x)^T d = \begin{bmatrix} 4(x_1^2 + x_2^2 - 2)x_1 \\ 4(x_1^2 + x_2^2 - 2)x_2 \end{bmatrix}^T \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0,
\]
for any feasible \( x \). Hence, \( \mathcal{F}(x) = \mathbb{R}^2 \), and \( T_{\Omega}(x) \neq \mathcal{F}(x) \).

Ex 2 (revisted)
\[
\min_x f(x) = x_1 + x_2, \quad c_1(x) = 2 - x_1^2 - x_2^2 \geq 0.
\]
Check that for \( x = (-\sqrt{2}, 0)^T \), \( T_{\Omega}(x) = \{(w_1, w_2) : w_1 \geq 0\} \). And a vector \( d \in \mathcal{F}(x) \) satisfies
\[
0 \leq \nabla c_1(x)^T d = 2\sqrt{2}d_1,
\]
and therefore \( T_{\Omega}(x) = \mathcal{F}(x) \) in this particular algebraic specification of the feasible set.

10.5 Constraint Qualification: LICQ

Constraint qualifications are the conditions to make sure \( T_{\Omega}(x) = \mathcal{F}(x) \). One
of the simplest kind (but rather restrictive) is the linear independence constraint qualification (LICQ):

**Definition 10.4.** Given a point \( x \) and an active set \( \mathcal{A}(x) \), we say LICQ holds if the set of active constraint gradients \( \{\nabla c_i(x) : i \in \mathcal{A}(x)\} \) is linearly independent. We call \( x \) is regular if LICQ holds at \( x \).

**Lemma 10.3.** If the LICQ conditions hold at a feasible point \( x \), then \( \mathcal{F}(x) = T_{\Omega}(x) \).

**Proof.** We first define a matrix with active constraint gradients as rows, \( A(x^*)^T = [\nabla c_i(x^*)]_{i \in \mathcal{A}(x^*)} \). Suppose that the matrix has \( m \) rows. Since LICQ holds, \( A(x^*) \)
10.6. LAGRANGIAN AND THE KKT CONDITIONS

has full row rank \( m \). Let \( Z \) be a matrix whose columns are a basis for the null space of \( A(x^*) \), that is,

\[
Z \in \mathbb{R}^{n \times (n-m)}, \quad Z \text{ has full column rank, } A(x^*)Z = 0.
\]

Choose a vector \( d \in \mathcal{F}(x^*) \), and suppose that \( \{t_k\} \) is any sequence of positive scalars satisfying \( \lim_{k \to \infty} t_k = 0 \). Define the parametrized system of equations \( R : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n \) by

\[
R(z, t) = \begin{bmatrix} c(z) - tA(x^*)d \\ Z^T(z - x^* - td) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]

We want to show that the solutions \( z = z_k \) of this system for small \( t = t_k > 0 \) give a feasible sequence approaching \( x^* \) and satisfying the definition of the tangent cone.

For \( t = 0 \), we set \( z = x^* \), and the Jacobian of \( R \) is then

\[
J_zR(x^*, 0) = A(x^*)Z^T \in \mathbb{R}^{n \times n}.
\]

The Jacobian is nonsingular due to the construction of \( Z \). By the implicit function theorem, the system of equations above has a unique solution \( z_k \) for all sufficiently small values of \( t_k \). Moreover, from the system and the fact that \( d \in \mathcal{F}(x^*) \) we can check that

\[
i \in \mathcal{E} \Rightarrow c_i(z_k) = t_k \nabla c_i(x^*)^T d = 0,
\]

\[
i \in \mathcal{A}(x^*) \cap \mathcal{J} \Rightarrow c_i(z_k) = t_k \nabla c_i(x^*)^T d \geq 0,
\]

and therefore \( z_k \) is feasible. To check if this choice of \( \{z_k\} \) satisfies the definition of the tangent cone, we use the fact that \( R(z_k, t_k) = 0 \) for all \( k \) and Taylor's theorem to find,

\[
0 = R(z_k, t_k) = \begin{bmatrix} c(z_k) - t_k A(x^*)d \\ Z^T(z_k - x^* - t_k d) \end{bmatrix} = \begin{bmatrix} A(x^*)d(z_k - x^*) + o(\|z_k - x^*\|_2) - t_k A(x^*)d \\ Z^T(z_k - x^* - t_k d) \end{bmatrix} = \begin{bmatrix} A(x^*) \\ Z_T^T \end{bmatrix}(z_k - x^* - t_k d) + o(\|z_k - x^*\|_2).
\]

Dividing both sides by \( t_k \) and using the invertibility of the matrix in the first term, we get

\[
\frac{z_k - x^*}{t_k} = d + o(\|z_k - x^*\|_2 / t_k).
\]

That is, \( d \in T_{x^*}(x^*) \).

\[\square\]

10.6 Lagrangian and the KKT Conditions

We define the Lagrangian function for general problems,

\[
\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{J}} \lambda_i c_i(x).
\]
Theorem 10.4 (FONC, KKT Conditions). Suppose that $x^*$ is a local minimizer, and $f$ and $c_i$ functions are continuously differentiable, and that LICQ holds at $x^*$ (i.e., $x^*$ is regular). Then there exists a unique Lagrange multiplier vector $\lambda^*$, with components $\lambda^*_i$, $i \in \mathcal{E} \cup \mathcal{I}$, such that the following conditions (Karush-Kuhn-Tucker conditions) are satisfied at $(x^*, \lambda^*)$:

\[
\nabla_x \mathcal{L}(x^*, \lambda^*) = 0
\]

Primal Feasibility Conditions
\[
c_i(x^*) = 0, \quad \text{for all } i \in \mathcal{E},
\]
\[
c_i(x^*) \geq 0, \quad \text{for all } i \in \mathcal{I},
\]

Dual Feasibility Conditions
\[
\lambda^*_i \in \mathbb{R}, \quad \text{for all } i \in \mathcal{E}
\]
\[
\lambda^*_i \geq 0, \quad \text{for all } i \in \mathcal{I}
\]

Complementarity Conditions
\[
\lambda^*_i c_i(x^*) = 0, \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}
\]

Ex.
\[
\min_x \left( x_1 - \frac{3}{2} \right)^2 + \left( x_2 - \frac{1}{2} \right)^4, \quad \text{s.t.} \quad \begin{bmatrix} 1 - x_1 - x_2 \\ 1 - x_1 + x_2 \\ 1 + x_1 - x_2 \\ 1 + x_1 + x_2 \end{bmatrix} \geq 0.
\]

The solution is at $x^* = (1, 0)^T$, and the first two constraints are active at this point. Denoting the two constraints by $c_1$ and $c_2$, we have

\[
\nabla f(x^*) = \begin{bmatrix} -1 \\ -1/2 \end{bmatrix}, \quad \nabla c_1(x^*) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \nabla c_2(x^*) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}
\]

We can check that the KKT conditions are satisfied with

\[
\lambda^* = (3/4, 1/4, 0, 0)^T.
\]

Ex. (No Lagrange multipliers)
\[
\min f(x) = x_1 + x_2, \quad c_1(x) = (x_1 - 1)^2 + x_2^2 - 1 = 0, \quad c_2(x) = (x_1 - 2)^2 + x_2^2 - 4 = 0.
\]

At the local minimizer $x^* = (0, 0)^T$ (the only feasible solution), the objective gradient $\nabla f(x^*) = (1, 1)^T$ cannot be expressed as a linear combination of the constraint gradients $\nabla c_1(x^*) = (-2, 0)^T$ and $\nabla c_2(x^*) = (-4, 0)^T$. That is, there exists no Lagrange multipliers $\lambda^*_1$ and $\lambda^*_2$ that satisfy $\nabla_x \mathcal{L}(x^*, \lambda^*_1, \lambda^*_2) = 0$.

The difficulty comes from the fact that the linearized feasible set here $\mathcal{F}(x^*) = \{(0, w_2)^T : w_2 \in \mathbb{R}\}$ does not match with $\mathcal{F}_0(x^*) = \{(0, 0)^T\}$.

The proof of Theorem 10.4 is rather simple, using a powerful lemma due to Farkas:

Lemma 10.5 (Farkas’ Lemma). For a cone $K$ defined as follows,

\[
K = \{B y + C w \mid y \geq 0\}
\]

where $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{n \times p}$, and for a given vector $g \in \mathbb{R}^n$, we have either

(i) $g \in K$
or there exists \( d \in \mathbb{R}^n \) such that

\[
(ii) \quad g^T d < 0, \quad B^T d \geq 0, \quad C^T d = 0.
\]

But (i) and (ii) cannot hold at the same time.

**Proof of Theorem 10.4**

Proof. Suppose that \( x^* \) is a local solution where LICQ holds.

Consider a cone defined by

\[
K := \left\{ \sum_{i \in \mathcal{A}(x^* \cap \mathcal{F})} \lambda_i \nabla c_i(x^*), \mid \lambda_i \geq 0 \text{ for } i \in \mathcal{A}(x^*) \cap \mathcal{F} \right\}
\]

and \( g := \nabla f(x^*) \). By Fakas’ lemma, we have either

\[
(i) \quad \nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda_i \nabla c_i(x^*), \quad \lambda_i \geq 0 \text{ for } i \in \mathcal{A}(x^*) \cap \mathcal{F}
\]

or (ii) there exists a direction \( d \in \mathcal{F}(x^*) \) such that

\[
d^T \nabla f(x^*) < 0.
\]

Since \( x^* \) is a local minimizer, we know from Theorem 10.2 that

\[
d^T f(x^*) \geq 0 \text{ for all } d \in T_{\Omega}(x^*),
\]

and since LICQ holds at \( x^* \), we also have \( T_{\Omega}(x^*) = \mathcal{F}(x^*) \). Therefore, only (i) above holds, i.e., there exist \( \lambda_i \)'s satisfying the condition in (i). We define a vector \( \lambda^* \) by

\[
\lambda^*_i = \begin{cases} 
\lambda_i, & i \in \mathcal{A}(x^*), \\
0, & i \in \mathcal{F} \setminus \mathcal{A}(x^*).
\end{cases}
\]

Then it is straightforward to check that the KKT conditions are satisfied for \( \lambda^* \). \( \square \)

**10.7 Other Constraint Qualifications**

**Linear Active Constraints** Suppose that at some \( x^* \in \Omega \), all active constraints \( c_i(\cdot), i \in \mathcal{A}(x^*) \), are affine functions. Then \( \mathcal{F}(x^*) = T_{\Omega}(x^*) \).

This condition is neither weaker or stronger than the LICQ.

**Mangasarian-Fromovitz Constraint Qualification (MFCQ)** The MFCQ holds for \( x^* \in \Omega \) if there exists a vector \( d \in \mathbb{R}^n \) such that

\[
d^T \nabla c_i(x^*) > 0, \quad \forall \mathcal{A}(x^*) \cap \mathcal{F},
\]

\[
d^T \nabla c_i(x^*) = 0, \quad \forall i \in \mathcal{E},
\]

and the set of equality constraint gradients \( \{ \nabla c_i(x^*), \ i \in \mathcal{E} \} \) is linearly independent.

MFCQ is a weaker condition than the LICQ, and guarantees that the Lagrange multiplier vector satisfying the KKT conditions exists and bounded (but may not be unique).
**Slater’s Condition**  For a convex program, Slater’s condition holds when the program is strictly feasible, i.e., there exists a point $x_0$ such that
\[ c_i(x_0) > 0, \quad \forall i \in \mathcal{A}, \]
\[ c_i(x_0) = 0, \quad \forall i \in \mathcal{E}. \]
Moreover, it is a sufficient condition for strong duality for convex optimization.

### 10.8 Second Order Optimality Conditions

Suppose that a point $x^*$ satisfies $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$ for some $\lambda^*$, i.e.,
\[ \nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda^*_i \nabla c_i(x^*). \tag{10.6} \]

If $x^*$ is a local minimizer (at which LICQ holds), then from Theorem 10.2 we have
\[ 0 \leq d^T \nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda^*_i d^T \nabla c_i(x^*), \quad \forall d \in \mathcal{F}(x^*). \]
If $d^T \nabla f(x^*) > 0$ for some $d \in \mathcal{F}(x^*)$, it is clear that moving along $d$ away from $x^*$ will increase the objective function value. If $d^T \nabla f(x^*) = 0$, however, we cannot determine if moving towards $d$ from $x^*$ will increase or decrease the objective function. In this case, we need to check the second order conditions to make sure what happens in the Lagrangian function on such directions.

#### 10.8.1 Critical Cone

Suppose that $f$ and all $c_i$’s are twice continuously differentiable. For a given $\mathcal{F}(x^*)$ and some Lagrange multiplier $\lambda^*$ satisfying the KKT conditions, we define the critical cone $\mathcal{G}(x^*, \lambda^*)$ as follows,
\[ \mathcal{G}(x^*, \lambda^*) = \{ d \in \mathcal{F}(x^*) : d^T \nabla c_i(x^*) = 0, \quad \forall i \in \mathcal{A}(x^*) \cap \mathcal{E} \text{ with } \lambda^*_i > 0 \}. \]

Or equivalently,
\[ \mathcal{G}(x^*, \lambda^*) = \left\{ d : d^T \nabla c_i(x^*) = 0, \quad \forall i \in \mathcal{A}(x^*) \cap \mathcal{E} \text{ with } \lambda^*_i > 0, \right\}. \]

From (10.6), we see that
\[ d \in \mathcal{G}(x^*, \lambda^*) \implies d^T \nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda^*_i d^T \nabla c_i(x^*) = 0. \]
That is, the critical cone $\mathcal{G}(x^*, \lambda^*)$ contains the directions from $\mathcal{F}(x^*)$, for which it is not decidable if $f$ will increase or decrease from the first order information.
Ex.
\[ \min_{x_1, x_2} x_1, \ s.t. \ x_2 \geq 0, \ 1 - (x_1 - 1)^2 - x_2^2 \geq 0. \]

Check,
\[ x^* = (0, 0)^T, \ \mathcal{A}(x^*) = \{1, 2\}, \ \lambda^* = (0, 1/2)^T. \]

Also, at \( x^* \) the LICQ holds, so the optimal Lagrange multiplier is unique.
\[ \mathcal{F}(x^*) = \{d : d \geq 0\}. \]
And,
\[ \mathcal{C}(x^*, \lambda^*) = \{(0, w_2)^T : w_2 \geq 0\}. \]

### 10.8.2 Second Order Optimality Conditions

**Theorem 10.6 (SONC).** Suppose that \( x^* \) is a local minimizer of the constrained problem, and the LICQ is satisfied at \( x^* \). Let \( \lambda^* \) be the Lagrange multiplier vector for which the KKT conditions are satisfied. Then,
\[ d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d \geq 0, \ \forall d \in \mathcal{C}(x^*, \lambda^*). \]

The sufficient second order condition is similar.

**Theorem 10.7 (SOSC).** Suppose that for some feasible point \( x^* \in \mathbb{R}^n \) there is a Lagrange multiplier vector \( \lambda^* \) such that the KKT conditions are satisfied. If,
\[ d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d > 0, \ \forall d \in \mathcal{C}(x^*, \lambda^*), \ d \neq 0, \]
then \( x^* \) is a strict local minimizer.

Ex.
\[ \min -0.1(x_1 - 4)^2 + x_2^2 \ s.t. \ x_1^2 + x_2^2 - 1 \geq 0. \]

The objective is a nonconvex function, and the feasible set is a nonconvex set. The objective is not bounded below: for example, it tends to \(-\infty\) for the feasible sequence \((10, 0)^T, (20, 0)^T, (30, 0)^T, \ldots. \) Therefore, no global solution exists.

However, it may still be possible to identify a strict local solution on the boundary of the feasible set.
\[ \nabla_x \mathcal{L}(x, \lambda) = \begin{bmatrix} -0.2(x_1 - 4) - 2\lambda_1 x_1 \\ 2x_2 - 2\lambda_1 x_2 \end{bmatrix} \]
\[ \nabla^2_{xx} \mathcal{L}(x, \lambda) = \begin{bmatrix} -0.2 - 2\lambda_1 & 0 \\ 0 & 2 - 2\lambda_1 \end{bmatrix} \]

At point \( x^* = (1, 0)^T \), the KKT conditions are satisfied with \( \lambda^*_1 = 0.3 \) and the active set \( \mathcal{A}(x^*) = 1 \). To check SOSC, we note that
\[ \nabla c_1(x^*) = (2, 0)^T, \]
so that the critical cone is
\[ \mathcal{C}(x^*, \lambda^*) = \{(0, w_2)^T : w_2 \in \mathbb{R}\}. \]
For any $d \in C(x^*, \lambda^*)$, $d \neq 0$, we have
\[
d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*) d = \begin{bmatrix} 0 & \mu_0 - 0.4 \\
0 & 1.4 \end{bmatrix} \begin{bmatrix} 0 \\
0 \end{bmatrix} = 1.2 w_2^2 > 0.
\]
Hence, the SOSC is satisfied at $x^*$ and therefore it is a strict local minimizer.

10.8.3 General Sufficient Condition

Consider the problem,
\[
\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad c_i(x) = 0, \quad i \in \mathcal{E}, \quad c_i(x) \geq 0, \quad i \in \mathcal{I}.
\]
Here $f$ and $c_i$ are real-valued functions on $\mathbb{R}^n$.

Theorem 10.8. Let $x^*$ be a feasible point with a dual-feasible vector $\mu^*$ such that the complementary slackness holds, that is,
\[
\mu_i^* \in \mathbb{R}, \quad \forall i \in \mathcal{E},
\mu_i^* \geq 0, \quad \forall i \in \mathcal{I},
\mu^*_i c_i(x^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}.
\]
and $x^*$ minimizes the Lagrangian function $\mathcal{L}(x, \mu^*)$ over $x \in \mathbb{R}^n$, i.e.,
\[
x^* \in \text{argmin}_{x \in \mathbb{R}^n} L(x, \mu^*).
\]
Then $x^*$ is a global minimizer of the problem.

Proof. From the conditions, we have
\[
f(x^*) = f(x^*) - (\mu^*)^T c(x^*)
= \min_{x \in \mathbb{R}^n} \{f(x) - (\mu^*)^T c(x)\}
\leq \min_{x \in \mathbb{R}^n, c_i(x) = 0, i \in \mathcal{E}, c_i(x) \geq 0, i \in \mathcal{I}} \{f(x) - (\mu^*)^T c(x)\}
\leq \min_{x \in \mathbb{R}^n, c_i(x) = 0, i \in \mathcal{E}, c_i(x) \geq 0, i \in \mathcal{I}} f(x).
\]

10.8.4 Convex Optimization

Theorem 10.9 (Convex Optimization). The first order necessary condition in Theorem 10.4 is also sufficient when $c_i(\cdot)$ are linear functions for $i \in \mathcal{E}$, $c_i(\cdot)$ are concave functions for $i \in \mathcal{I}$, and $f$ is a convex function.

Proof. Suppose that $x^*$ and $\lambda^*$ satisfy the KKT conditions. From the given conditions, we can easily check that the Lagrangian
\[
\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)
\]
is a convex function, in which case $x^*$ satisfying $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$ is a global solution of $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda^*)$. Therefore, $x^*$ is a global minimizer by Theorem 10.8.
Exercise

10.1. Let \( h : \mathbb{R}^n \to \mathbb{R}^m \) be a smooth vector-valued function (each component function \( h_i : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable for \( i = 1, 2, \ldots, m \)). Consider the (in general nonsmooth) unconstrained optimization problems

\[
(i) \min_{x \in \mathbb{R}^n} \|h(x)\|_\infty, \text{ and } (ii) \min_{x \in \mathbb{R}^n} \max_{i = 1, \ldots, m} h_i(x)
\]

Reformulate each of these problems as a smooth constrained optimization problem.

10.2. The Euclidean projection of a point \( z \in \mathbb{R}^n \) onto a half-space \( H := \{x \in \mathbb{R}^n : a^T x + b \geq 0\} \) where \( a \in \mathbb{R}^n, a \neq 0, \) and \( b \in \mathbb{R}, \) can be formulated as follows,

\[
\text{Proj}_H(z) = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} \|z - x\|_2^2, \text{ subject to } a^T x + b \geq 0.
\]

Derive the expression of \( \text{Proj}_H(z) \) and discuss its optimality.

10.3. Consider the problem of finding the point \((x, y)^T \in \mathbb{R}^2\) on the parabola \( y = \frac{(x - 1)^2}{5} \) that is closest to \((1, 2)^T\), in the Euclidean norm:

\[
\min_{(x,y)^T \in \mathbb{R}^2} (x - 1)^2 + (y - 2)^2, \text{ subject to } (x - 1)^2 = 5y.
\]

(a) Find all the point satisfying the KKT conditions. Is the LICQ satisfied?

(b) Which of these are solutions?

(c) By directly substituting the constraint into the objective function and eliminating the variable \( x \), an unconstrained minimization problem can be obtained. Show that the solutions of the new problem cannot be solutions of the original problem. What was wrong with the direct substitution?

10.4. Consider minimizing a smooth function \( f : \mathbb{R}^n \to \mathbb{R} \) on a box constraint,

\[
\min_{x \in \mathbb{R}^n} f(x), \text{ s.t. } a_i \leq x_i \leq b_i \ i = 1, 2, \ldots, n,
\]

defined with scalars \( a_i \) and \( b_i \), \( i = 1, \ldots, n \), and suppose that \( x^* \) is a local minimizer. Derive the expression of necessary optimality conditions of \( x^*_i \) in terms of \( g_i^* := [\nabla f(x^*)]_i \), for the cases \( x^*_i = a_i \), \( x^*_i = b_i \), and \( a_i < x^*_i < b_i \).

10.5. Consider the problem

\[
\min_{x \in \mathbb{R}^2} f(x) = -2x_1 + x_2
\]

s.t. \((1 - x_1)^3 - x_2 \geq 0 \)

\[
x_2 + \frac{1}{4}x_1^2 - 1 \geq 0.
\]

The optimal solution is \( x^* = (0, 1)^T \). Answer the following questions:

(i) Do the LICQ hold at \( x^* \)?
(ii) Are the KKT conditions satisfied at \( x^* \)?

(iii) Show the expression of \( F(x^*) \) and \( G(x^*, \lambda^*) \).

(iv) Are the second-order necessary conditions satisfied? How about the sufficient conditions?

10.6. Find all minimizers of \( f(x) = x_1 x_2 \) on \( x_1^2 + x_2^2 = 1 \).
Chapter 11

Duality

Duality is used to develop important algorithms for constrained optimization such as the augmented Lagrangian algorithm. It also provides a tool to analyze the structure of problems. The theory of duality goes beyond nonlinear optimization, including convex nonsmooth optimization and discrete programming problems. Sometimes it leads to problems that are easier to solve than the original ones.

11.1 The Primal Problem

We consider the following problem as the primal problem:

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f(x) \\
\text{s.t.} & \quad c_i(x) \geq 0, \quad i = 1, 2, \ldots, m.
\end{align*}
\]  

(11.1)

For simplicity, we only consider the inequality constraints: equality constraints \( g_i(x) = 0 \) can be included using both \( g_i(x) \geq 0 \) and \( -g_i(x) \geq 0 \).

By collecting the constraints in a vector
\[ c(x) := (c_1(x), c_2(x), \ldots, c_m(x))^T, \]
and collecting the corresponding Lagrange multipliers in a vector
\[ \lambda := (\lambda_1, \lambda_2, \ldots, \lambda_m)^T, \]
we can write the Lagrange function as
\[ \mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x). \]

11.2 The Dual Problem

First, we define the dual objective function \( q : \mathbb{R}^m \rightarrow \mathbb{R} \) as follows,
\[ q(\lambda) := \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda). \]

Note that this requires to find the global minimizer of \( \mathcal{L} \) in \( x \) if it exists, which could be extremely difficult. However, when \( f \) is convex and \( c_i \) for \( i = 1, 2, \ldots, m \) are concave, we can show that \( \mathcal{L}(\cdot, \lambda) \) is a convex function, in which case the computation of \( q(\lambda) \) becomes far more practical.
CHAPTER 11. DUALITY

In many problems the infimum is \(-\infty\) for some values of \(\lambda\). We define the effective domain of \(q\) as
\[
\mathcal{D} := \{ \lambda \in \mathbb{R}^m : q(\lambda) > -\infty \}.
\]
Finally, we define the dual problem as follows,
\[
\sup_{\lambda \in \mathbb{R}^m} q(\lambda) \quad \text{s.t.} \quad \lambda \geq 0.
\]
(11.2)

We state an important property of the dual objective function \(q(\lambda)\), which makes the dual problem to be well-conditioned even though the corresponding primal problem is not.

**Theorem 11.1.** The dual objective function \(q(\lambda)\) is concave and its domain \(\mathcal{D}\) is convex, regardless of convexity (or concavity) of the primal objective \(f(\cdot)\) and constraints \(c_i\).

**Proof.** For any \(\lambda\) and \(\lambda'\) in \(\mathbb{R}^m\), any \(x \in \mathbb{R}^n\), and any \(\alpha \in [0,1]\), we have
\[
\mathcal{L}(x, (1-a)\lambda + a\lambda') = (1-a)\mathcal{L}(x, \lambda) + a\mathcal{L}(x, \lambda').
\]
By taking infimum of both sides and using the fact that \(\inf\{x_k\} + \inf\{y_k\} \leq \inf\{x_k + y_k\}\), and using the definition of the dual function, we have
\[
q((1-a)\lambda + a\lambda') \geq (1-a)q(\lambda) + aq(\lambda').
\]
That is, \(q\) is a concave function. If both \(\lambda\) and \(\lambda'\) belong to \(\mathcal{D}\), then the inequality also implies that \(q((1-a)\lambda + a\lambda') > -\infty\), and therefore \((1-a)\lambda + a\lambda' \in \mathcal{D}\) as well. That is, \(\mathcal{D}\) is a convex set.

This implies that the dual problem is always a convex optimization problem, even though the corresponding primal problem is not a convex optimization problem.

### 11.3 Weak Duality

**Theorem 11.2.** For any primal feasible \(\hat{x}\) and dual feasible \(\hat{\lambda}\), we have
\[
q(\hat{\lambda}) \leq f(\hat{x}).
\]

**Proof.**
\[
q(\hat{\lambda}) = \inf_x (f(x) - \hat{\lambda}^T c(x)) \leq f(\hat{x}) - \hat{\lambda}^T c(\hat{x}) \leq f(\hat{x}).
\]
The last inequality was from the fact that \(c(\hat{x}) \geq 0\) and \(\hat{\lambda} \geq 0\) due to feasibility.

The above theorem allows us to use the dual problems to find a lower bound of the primal optimal objective function value, that is,
\[
q^* := \max_{\lambda \in \mathbb{R}^m, \lambda \geq 0} q(\lambda) \leq \min_{x \in \mathbb{R}^n, c(x) \geq 0} f(x) := f^*
\]
The difference, \(f(\hat{x}) - q(\hat{\lambda}) \geq 0\), is called as the duality gap, and \(f^* - g^*\) as the optimal duality gap.
11.4 Duality in Convex Optimization

We consider convex minimization, in forms of

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

\[s.t. \ c_i(x) = 0, \ i \in \mathcal{E}\]

\[c_i(x) \geq 0, \ i \in \mathcal{I}.\]

where \(c_i(\cdot)\)'s are affine functions for \(i \in \mathcal{E}\) and \(-c_i(\cdot)\)'s are convex functions. Again, we assume that we have only inequality constraints without loss of generality.

11.4.1 Strong Duality

The following result shows that for the pair \((x^*, \lambda^*)\) that satisfies the KKT conditions of the convex primal problem, \(\lambda^*\) is the solution of the dual problem and there is no duality gap (i.e., strong duality holds).

**Theorem 11.3** (Strong Duality). Suppose that \(x^*\) is a solution of the primal problem, and that \(f\) and \(-c_i, i = 1, 2, \ldots, m\), are convex on \(\mathbb{R}^n\) and differentiable at \(x^*\). Then any \(\lambda^*\) for which \((x^*, \lambda^*)\) satisfies the KKT conditions is a solution of the dual problem. In particular, \(q(\lambda^*) = f(x^*)\).

**Proof.** Suppose that \((x^*, \lambda^*)\) satisfies the KKT conditions. Then \(\nabla_x \mathcal{L}(x^*, \lambda^*) = 0\). From the given conditions and \(\lambda^* \geq 0\), we have that \(\mathcal{L}(\cdot, \lambda^*)\) is a convex and differentiable function. Therefore, for any \(x \in \mathbb{R}^n\), we see that

\[
\mathcal{L}(x, \lambda^*) \geq \mathcal{L}(x^*, \lambda^*) + \nabla_x \mathcal{L}(x^*, \lambda^*)^T (x - x^*) = \mathcal{L}(x^*, \lambda^*),
\]

where the last equality used the fact that \(\nabla_x \mathcal{L}(x^*, \lambda^*) = 0\) from the KKT conditions. Therefore,

\[
q(\lambda^*) = \inf_x \mathcal{L}(x, \lambda^*) = \mathcal{L}(x^*, \lambda^*) = f(x^*) - (\lambda^*)^T c(x^*) = f(x^*),
\]

where the last equality is from the complementary slackness condition in the KKT conditions. From Theorem 11.2, we know that \(q(\lambda) \leq f(x^*)\) for all \(\lambda \geq 0\), and therefore the above expression tells that \(\lambda^*\) maximizes \(q(\lambda)\).

11.4.2 Sufficient Conditions for Strong Duality

In Theorem 11.3, a multiplier \(\lambda^*\) satisfying the KKT conditions will be guaranteed to exist when a constraint qualification (CQ) holds at \(x^*\). As a consequence in Theorem 11.3, CQ+KKT leads to strong duality in convex optimization.

There are other conditions that are sufficient for strong duality: i) all constraints are affine, and (ii) the Slater’s condition, i.e., there exists \(x_0\) such that

\[
c_i(x_0) > 0, \ \forall i \in \mathcal{I},
\]

\[
c_i(x_0) = 0, \ \forall i \in \mathcal{E},
\]
11.4.3 Wolfe Dual

In convex cases, we have a short cut to construct the dual problem known as the Wolfe dual:

$$\max_{x,\lambda} \mathcal{L}(x, \lambda)$$

subject to

$$\nabla_x \mathcal{L}(x, \lambda) = 0,$$

$$\lambda \geq 0.$$ 

Its relation to the general definition of dual problems is straightforward.

**Theorem 11.4.** Suppose that $f$ and $-c_i$, $i = 1, 2, \ldots, m$, are convex and continuously differentiable on $\mathbb{R}^n$. Then solving the Wolfe dual is equivalent to solving the dual problem (11.2).

**Proof.** From given conditions, $\mathcal{L}(\cdot, \lambda)$ is convex for any $\lambda \geq 0$, and therefore $\hat{x}$ satisfying $\nabla_x \mathcal{L}(\hat{x}, \lambda) = 0$ minimizes $\mathcal{L}(\cdot, \lambda)$. That is,

$$q(\lambda) := \inf_x \mathcal{L}(x, \lambda) = \mathcal{L}(\hat{x}, \lambda).$$

Therefore, the Wolfe dual can be rewritten as

$$\max_{\lambda} q(\lambda)$$

subject to

$$\lambda \geq 0.$$ 

which is the dual problem (11.2). 

**Ex. (Linear Program)** Consider the following linear program,

$$\min_{x \in \mathbb{R}^n} c^T x, \text{ s.t. } Ax \geq b,$$

with given $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$. The dual objective is (with $\lambda \in \mathbb{R}^m$),

$$q(\lambda) = \inf_x [c^T x - \lambda^T (Ax - b)] = \inf_x [(c - A^T \lambda)^T x + b^T \lambda]$$

$$= \begin{cases} 
-\infty & \text{if } c - A^T \lambda \neq 0 \\
 b^T \lambda & \text{otherwise.}
\end{cases}$$

In maximizing $q(\lambda)$, we can exclude the case $c - A^T \lambda \neq 0$, and therefore we can write the dual problem as follows,

$$\max_{\lambda \in \mathbb{R}^n} b^T \lambda \text{ s.t. } A^T \lambda = c, \lambda \geq 0.$$ 

Using Wolfe dual leads to the same dual problem.

**Ex. (Convex Quadratic Program)** Consider the following QP,

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T G x + c^T x, \text{ s.t. } Ax \geq b,$$
where $G$ is symmetric and positive definite. The dual objective is given by

$$q(\lambda) = \inf_x \mathcal{L}(x, \lambda) = \inf_x \frac{1}{2} x^T G x + c^T x - \lambda^T (Ax - b).$$

Since $G$ is p.d., $\mathcal{L}(\cdot, \lambda)$ is strictly convex and its minimum is achieve when

$$\nabla_x \mathcal{L}(x, \lambda) = Gx + c - A^T \lambda = 0.$$

Replacing $x$, we can write the dual objective as

$$q(\lambda) = -\frac{1}{2} (A^T \lambda - c)^T G^{-1} (A^T \lambda - c) + b^T \lambda.$$

Using Wolfe dual, the dual problem is written as

$$\max_{\lambda, x} -\frac{1}{2} x^T G x + b^T \lambda, \quad \text{s.t. } Gx + c - A^T \lambda = 0, \; \lambda \geq 0.$$

This is equivalent to the above expression if we do the same replacement.

**Ex. (Nonconvex QCQP with Strong Duality)** Consider the following nonconvex quadratically constrained quadratic program,

$$\min_{x \in \mathbb{R}^n} -x^T x, \quad \text{s.t. } x^T x \leq 1,$$

$$\mathcal{L}(x, \lambda) = -x^T x - \lambda(1 - x^T x) = (\lambda - 1)x^T x - \lambda.$$

And the dual obj. function is give by

$$q(\lambda) = \begin{cases} -\lambda & \lambda \geq 1 \\ -\infty & \text{o.w.} \end{cases}$$

Therefore the dual problem is

$$\max_{\lambda \in \mathbb{R}} -\lambda, \quad \text{s.t. } \lambda \geq 1.$$ 

It is easy to see that both the optimal primal and dual objective values are the same. This is an example when strong duality holds for a nonconvex problem. In fact, strong duality holds for any optimization problem with quadratic objective and a single quadratic inequality constraint, if Slater’s condition is satisfied.

**Ex. (Relaxation of Binary LP)** Consider the following binary (or boolean) linear program,

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t. } Ax \geq b, \; x_i \in \{0, 1\}, \; i = 1, 2, \ldots, n.$$ 

There are two possible ways to get lower bounds to the optimal objective of it:

(i) LP relaxation:

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t. } Ax \geq b, \; 0 \leq x_i \leq 1, \; i = 1, 2, \ldots, n.$$ 

(ii) Lagrangian relaxation: Consider a reformulation of the binary LP:

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t. } Ax \geq b, \; x_i(1 - x_i) = 0, \; i = 1, 2, \ldots, n.$$
and find the dual of this problem.

Here we show that the lower bounds from the two approaches are in fact the same. The Lagrangian of the formulation in (ii),

$$L(x, \lambda, \eta) = c^T x - \lambda^T (Ax - b) - \sum_{i=1}^{n} \eta_i x_i + \sum_{i} \eta_i x_i^2$$

Note that the Lagrangian is convex in $x$ if $\eta_i \geq 0$ for all $i$. Therefore the dual objective is given by

$$q(\lambda, \eta) = \begin{cases} 
\lambda^T b - \frac{1}{4} \sum_i (c_i - (A^T \lambda)_i - \eta_i)^2 / \eta_i & \eta > 0 \\
\lambda^T b & \eta = 0, c = A^T \lambda \\
-\infty & \text{o.w.}
\end{cases}$$

And the dual problem becomes

$$\sup_{\lambda \geq 0, \eta \geq 0} q(\lambda, \eta).$$

In case of $\eta > 0$, we note that the dual objective is separable over $\lambda$ and $\eta$, and again for each $\eta_i$: therefore we can perform maximization over each $\eta_i$ separately,

$$\sup_{\eta_i \geq 0} -(z_i - \eta_i)^2 / \eta_i$$

where $z_i := c_i - (A^T \lambda)_i$. The supremum is given by

$$\sup_{\eta_i > 0} -(z_i - \eta_i)^2 / \eta_i = \begin{cases} 
0 & z_i \geq 0 \\
4z_i & z_i < 0
\end{cases} = \min\{0, 4z_i\}.$$

Therefore the dual problem can be simplified into

$$\sup_{\lambda \geq 0} \lambda^T b + \sum_i \min\{0, c_i + (A^T \lambda)_i\}$$

(11.3)

Now we consider the Wolfe dual of the LP relaxation (i):

$$\max_{x, \lambda, \delta, \eta} c^T x - \lambda^T (Ax - b) - \delta^T x - \eta^T (1 - x)$$

s.t. $c - A^T \lambda - \delta + \eta = 0$

$$\lambda \geq 0, \delta \geq 0, \eta \geq 0.$$

It can be simplified to (replacing $-\eta$ with $\eta$ for the sake of interpretation),

$$\max_{\lambda, \eta} \lambda^T b + \eta^T \hat{x}$$

s.t. $\eta \leq c - A^T \lambda$

$$\lambda \geq 0, \eta \leq 0.$$

(11.4)

And we can see that (11.3) and (11.4) are equivalent.
11.5 Saddle-Point Interpretation of Duality

Consider the Lagrangian of the problem (11.1),
\[ \mathcal{L}(x; \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x). \]

We first note that
\[ \sup_{\lambda \geq 0} \mathcal{L}(x; \lambda) = \begin{cases} f(x), & c_i(x) \geq 0, \quad i = 1, \ldots, m, \\ +\infty, & \text{o.w.} \end{cases} \]

And therefore we can express the optimal primal objective value as
\[ f(x^*) = \inf_x \sup_{\lambda \geq 0} \mathcal{L}(x; \lambda). \]

From the definition of the dual objective, we have
\[ q(\lambda^*) = \sup_{\lambda \geq 0} \inf_x \mathcal{L}(x; \lambda). \]

And from the weak duality, we have the following max-min inequality,
\[ \sup_{\lambda \geq 0} \inf_x \mathcal{L}(x; \lambda) \leq \inf_x \sup_{\lambda \geq 0} \mathcal{L}(x; \lambda). \]

And the strong duality implies that
\[ \sup_{\lambda \geq 0} \inf_x \mathcal{L}(x; \lambda) = \inf_x \sup_{\lambda \geq 0} \mathcal{L}(x; \lambda). \]

This tells us an important fact: the order of sup and inf can be exchanged when strong duality holds.

The max-min inequality indeed holds for any function \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \) and sets \( X \subset \mathbb{R}^n \) and \( Y \subset \mathbb{R}^m \),
\[ \sup_{y \in Y} \inf_{x \in X} f(x, y) \leq \inf_{x \in X} \sup_{y \in Y} f(x, y), \]
where the equality (called the strong max-min property) holds only for special cases.

A pair of points \((x^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}_+^m\) is called a saddle-point for \( \mathcal{L} \) if
\[ \mathcal{L}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x, \lambda^*) \]
for all feasible \( x \) and \( \lambda \). That is,
\[ \mathcal{L}(x^*, \lambda^*) = \inf_x \mathcal{L}(x, \lambda^*), \quad \mathcal{L}(x^*, \lambda^*) = \sup_{\lambda \geq 0} \mathcal{L}(x^*, \lambda), \]
which also implies that the strong max-min property holds.

In terms of the duality, this tells us that if \( x^* \) and \( \lambda^* \) are the primal and the dual optimal solutions, resp., for a problem in which strong duality holds, then the pair \((x^*, \lambda^*)\) is a saddle point of the Lagrangian. The converse is also true, that if \((x, \lambda)\) is a saddle point of the Lagrangian, then \( x \) and \( \lambda \) are primal and dual optimal, respectively, and the strong duality holds.
11.6 Finding Primal Solutions from Dual Solutions

Now the question is that by solving the dual problem, if we can find the primal solution.

**Theorem 11.5.** Suppose that \( f \) and \(-c_i, i = 1, 2, \ldots, m\), are convex and continuously differentiable on \( \mathbb{R}^n \). Suppose that \( x^* \) is a solution of the primal problem at which a CQ holds. Suppose that \( \hat{\lambda} \) solves the dual problem and that the infimum in \( \inf_x \mathcal{L}(x, \hat{\lambda}) \) is attained at \( \hat{x} \). Then \( f(x^*) = \mathcal{L}(\hat{x}, \hat{\lambda}) \). If \( \mathcal{L}(\cdot, \hat{\lambda}) \) is strictly convex, then \( x^* = \hat{x} \).

**Proof.** From FONC (with LICQ at \( x^* \)), there must exist a Lagrange multiplier vector \( \lambda^* \) satisfying the KKT conditions. (Then \( \nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \) and hence \( x^* \) minimizes \( \mathcal{L}(\cdot, \lambda^*) \) due to convexity.) Also, from Theorem 11.3, \( \lambda^* \) solves the dual problem, so that
\[
\mathcal{L}(x^*, \lambda^*) = q(\lambda^*) = q(\hat{\lambda}) = \mathcal{L}(\hat{x}, \hat{\lambda}). \tag{11.5}
\]
Furthermore, we also see that
\[
f(x^*) = \mathcal{L}(x^*, \lambda^*) = \mathcal{L}(\hat{x}, \hat{\lambda}) = q(\hat{\lambda}),
\]
where the first equality comes from the complementarity condition. This tells that (without strict convexity) there is no duality gap.

Since \( \hat{x} = \arg\min_x \mathcal{L}(x, \hat{\lambda}) \), we have \( \nabla_x \mathcal{L}(\hat{x}, \hat{\lambda}) = 0 \) from the convexity of \( \mathcal{L}(x, \hat{\lambda}) \).

Suppose that \( x^* \neq \hat{x} \) for contradiction. Then from the strict convexity of \( \mathcal{L}(\cdot, \hat{\lambda}) \), we have for \( x^* \neq \hat{x} \) that
\[
\mathcal{L}(x^*, \hat{\lambda}) > \mathcal{L}(\hat{x}, \hat{\lambda}) + \nabla_x \mathcal{L}(\hat{x}, \hat{\lambda})^T(x^* - \hat{x}) = 0.
\]
Together with (11.5), this implies that
\[
\mathcal{L}(x^*, \hat{\lambda}) > \mathcal{L}(\hat{x}, \hat{\lambda}) = \mathcal{L}(x^*, \lambda^*).
\]
In particular, we have
\[
-\hat{\lambda}^T c(x^*) > -(\lambda^*)^T c(x^*) = 0,
\]
where the last equality is from the complementarity condition. Since \( \hat{\lambda} \geq 0 \) and \( c(x^*) \geq 0 \), this gives a contradiction, and therefore \( x^* = \hat{x} \).

Note that if \( f(x) \) is strictly convex, or if one of \( -c_i(x) \), \( i = 1, \ldots, m \), with \( \hat{\lambda}_i > 0 \) is strictly convex, then \( \mathcal{L}(\cdot, \hat{\lambda}) \) becomes strictly convex.
11.6. FINDING PRIMAL SOLUTIONS FROM DUAL SOLUTIONS

Exercise

11.1. Show that the dual of the following linear program

\[ \min_{x \in \mathbb{R}^n} \ c^T x, \quad \text{s.t.} \ Ax \geq b, \ Hx = g, \ x \geq 0, \]

for given \( A \in \mathbb{R}^{p \times n}, \ H \in \mathbb{R}^{q \times n}, \ c \in \mathbb{R}^n, \ b \in \mathbb{R}^p, \) and \( g \in \mathbb{R}^q \) can be expressed as

\[ \max_{\alpha \in \mathbb{R}^p, \beta \in \mathbb{R}^q} b^T \alpha + g^T \beta, \quad \text{s.t.} \ A^T \alpha + H^T \beta \leq c, \ \alpha \geq 0. \]

11.2. Derive a dual problem of

\[ \min_{x \in \mathbb{R}^n} \sum_{i=1}^N \|A_i x + b_i\|_2 + \frac{1}{2} \|x - x_0\|_2^2, \]

for \( A_i \in \mathbb{R}^{m_i \times n}, \ b_i \in \mathbb{R}^{m_i}, \) and \( x_0 \in \mathbb{R}^n. \) As a first step, introduce new variables \( y_i \in \mathbb{R}^{m_i} \) and equality constraints \( y_i = A_i x + b_i \) for \( i = 1, \ldots, N. \)