Optimization Models in Engineering

Course Reader

EECS 127/227AT

University of California, Berkeley
Department of Electrical Engineering and Computer Sciences

Spring 2024

This document contains lecture notes and supplementary material for the optimization course EECS 127/227AT.

Acknowledgments

These course notes have been developed over many years with contributions from numerous teaching assistants, graduate student instructors, and students. We gratefully acknowledge their efforts in improving the clarity and completeness of this material.

The presentation draws heavily from the excellent textbooks by Boyd and Vandenberghe [1] and Calafiore and El Ghaoui [2]. Students are encouraged to consult these references for additional depth and examples.

Note: Figure placeholders are marked with `% TODO: Figure` comments throughout the text. These indicate where diagrams would appear in the original course materials.

Contents

Chapter 1

Introduction

References

- Boyd & Vandenberghe [1]: Chapter 1
- Calafiore & El Ghaoui [2]: Chapter 1

1.1 What is Optimization?

Try to see what the following "problems" have in common.

- A statistical model, such as a neural network, trains using finite data samples.
- A robot learns a strategy using the environment, so that it does what you want.
- A major gas company decides what mixture of different fuels to process in order to get maximum profit.
- The EECS department decides how to set class sizes in order to maximize the number of credits offered subject to budget constraints.

While it might seem that these four examples are very distinct, they can all be formulated as *minimizing* an *objective function* over a *feasible set*. Thus, they can all be put into the framework of optimization.

To develop the basics of optimization, including precisely defining an objective function and a feasible set, we use some motivating examples from the third and fourth "problems". (The first and second "problems" will be discussed at the very end of the course.)

Example 1.1 (Oil and Gas)**.** Say that we are a gas company with 10^5 barrels of crude oil that we *must* refine by an expiration date. There are two refineries: one which processes crude oil into jet fuel, and one which processes crude oil into gasoline. We can sell a barrel of jet fuel to consumers for \$0.10, while we can sell a barrel of gasoline fuel for \$0.20. So, letting x_1 be a variable denoting the number of barrels of jet fuel produced, and x_2 be a variable denoting the number of barrels of

gasoline produced, we aim to solve the problem:

$$\begin{aligned} \max_{x_1,x_2} \quad & \frac{1}{10}x_1 + \frac{1}{5}x_2 \\ \text{s.t.} \quad & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_1 + x_2 = 10^5. \end{aligned} \tag{1.1}$$

That is, we aim to choose x_1 and x_2 which maximize the *objective function* $\frac{1}{10}x_1 + \frac{1}{5}x_2$, but with the caveat that they must obey the *constraints* $x_1 \geq 0$, $x_2 \geq 0$, and $x_1 + x_2 = 10^5$. The *feasible set* is the set of all (x_1, x_2) pairs which obey the constraints. As you may have noticed, constraints can be equalities or inequalities in the x_i, which we formalize shortly.

The solution to this problem can be seen to be $(x_1^\star, x_2^\star) = (0, 10^5)$, which corresponds to refining all the crude oil into gasoline. This makes sense – after all, gasoline sells for more! And with all else equal between gasoline and jet fuel, to maximize our profit, we just need to produce gasoline.

To model another constraint, say that we need at least 10^3 gallons of jet fuel and $5 \cdot 10^2$ gallons of gasoline, we can directly incorporate them into the constraint set:

$$\begin{aligned} \max_{x_1,x_2} \quad & \frac{1}{10}x_1 + \frac{1}{5}x_2 \\ \text{s.t.} \quad & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_1 \geq 10^3 \\ & x_2 \geq 5 \cdot 10^2 \\ & x_1 + x_2 = 10^5. \end{aligned} \tag{1.2}$$

We then notice that $x_1 \geq 0$ is made redundant by the constraint $x_1 \geq 10^3$. That is, no pair (x_1, x_2) which satisfies $x_1 \geq 10^3$ is not going to satisfy $x_1 \geq 0$. Thus, we can eliminate the latter constraint, since it defines the same feasible set. We can do the same thing for the constraints $x_2 \geq 0$ and $x_2 \geq 5 \cdot 10^2$, the latter making the former redundant. Thus, we can simplify the above problem to only include the redundant constraints:

$$\begin{aligned} \max_{x_1,x_2} \quad & \frac{1}{10}x_1 + \frac{1}{5}x_2 \\ \text{s.t.} \quad & x_1 \geq 10^3 \\ & x_2 \geq 5 \cdot 10^2 \\ & x_1 + x_2 = 10^5. \end{aligned} \tag{1.3}$$

Let's say that we want to incorporate one final business need. Before, we were modeling that the oil refinement is free, since we don't have an objective or constraint term which involves this cost. Now, let us say that we can transport a total of $2 \cdot 10^6$ "barrel-miles" – that is, the number of barrels times the number of miles we can transport is no greater than $2 \cdot 10^6$. Let us further say that the jet fuel

refinery is 10 miles away from the crude oil storage, and the gasoline refinery is 30 miles away from the crude oil storage. We can incorporate this further constraint into the constraint set directly:

$$\begin{aligned} \max_{x_1,x_2} \quad & \frac{1}{10}x_1 + \frac{1}{5}x_2 \\ \text{s.t.} \quad & x_1 \geq 10^3 \\ & x_2 \geq 5 \cdot 10^2 \\ & 10x_1 + 30x_2 \leq 2 \cdot 10^6 \\ & x_1 + x_2 = 10^5. \end{aligned} \tag{1.4}$$

This is a good first problem; we have a non-trivial objective function, non-trivial inequality and equality constraints, and even got to work with manipulating constraints (so as to remove redundant ones)!

This type of optimization problem is called a *linear program*. We will learn more about how to formulate and solve linear programs later in the course.

A more generic reformulation of the above optimization problem is the following "standard form".

Definition 1.2 (Standard Form of Optimization Problem)**.** We say that an optimization problem is written in *standard form* if it is of the form

$$\begin{aligned} \min_{\vec{x}\in\mathbb{R}^n} \quad & f_0(\vec{x}) \\ \text{s.t.} \quad & f_i(\vec{x}) \leq 0, \quad \forall i \in \{1,\dots,m\} \\ & h_j(\vec{x}) = 0, \quad \forall j \in \{1,\dots,p\}. \end{aligned} \tag{1.5}$$

Here:

- $\vec{x} \in \mathbb{R}^n$ is the **optimization variable**.
- $f_1, \dots, f_m$ and $h_1, \dots, h_p$ are functions $\mathbb{R}^n \to \mathbb{R}$.
- f_0 is the **objective function**.
- f_i are **inequality constraint functions**; the expression "$f_i(\vec{x}) \leq 0$" is an **inequality constraint**.
- Similarly, h_j are **equality constraint functions**, and the expression "$h_j(\vec{x}) = 0$" is an **equality constraint**.
- The **feasible set**, i.e., the set of all $\vec{x}$ that satisfy all constraints, is

$$\Omega \triangleq \left\{ \vec{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} f_i(\vec{x}) \leq 0, & \forall i \in \{1,\dots,m\} \\ h_j(\vec{x}) = 0, & \forall j \in \{1,\dots,p\} \end{array} \right\}. \tag{1.6}$$

  We can thus also write the problem (1.5) as

$$\min_{\vec{x}\in\Omega} f_0(\vec{x}). \tag{1.7}$$

- A **solution** to this optimization problem is any $\vec{x}^\star \in \Omega$ which attains the minimum value of $f(\vec{x})$ across all $\vec{x} \in \Omega$. Correspondingly, $\vec{x}^\star$ is also called a **minimizer** of f_0 over Ω.

It's perfectly fine if $m = 0$ (in which case there are no inequality constraints) and/or $p = 0$ (in which case there are no equality constraints). If there are no constraints, then $\Omega = \mathbb{R}^n$ and the problem is called *unconstrained*; otherwise it is called *constrained*.

Let us try another example now, which has vector-valued quantities.

Example 1.3. Consider the following table of EECS courses:

Class	Size	Credits	Resources per Student
127	x_1	c_1	r_1
126	x_2	c_2	r_2
182	x_3	c_3	r_3
189	x_4	c_4	r_4
162	x_5	c_5	r_5
188	x_6	c_6	r_6
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Suppose there are n classes in total. Let $\vec{x} \triangleq \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top \in \mathbb{R}^n$ be the decision variable, and let $\vec{c} \triangleq \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}^\top \in \mathbb{R}^n$ and $\vec{r} \triangleq \begin{bmatrix} r_1 & r_2 & \cdots & r_n \end{bmatrix}^\top \in \mathbb{R}^n$ be constants. Then, in order to maximize the total number of credit hours subject to a total resource budget b, we set up the linear program

$$\begin{aligned} \max_{\vec{x} \in \mathbb{R}^n} \quad & \vec{c}^\top \vec{x} \\ \text{s.t.} \quad & \vec{r}^\top \vec{x} \leq b \\ & x_i \geq 0, \quad \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{1.8}$$

As notation, instead of the last set of constraints $x_i \geq 0$, we can write the vector constraint $\vec{x} \geq \vec{0}$.

More generally, recall that if we have a vector equality constraint $\vec{h}(\vec{x}) = \vec{0}$, it can be viewed as short-hand for the several scalar equality constraints $h_1(\vec{x}) = 0, \ldots, h_p(\vec{x}) = 0$. Correspondingly, we define the vector inequality constraint $\vec{f}(\vec{x}) \leq \vec{0}$ to be short-hand for the several scalar inequality constraints $f_1(\vec{x}) \leq 0, \ldots, f_m(\vec{x}) \leq 0$.

1.2 Least Squares

We begin with one of the simplest optimization problems, that of least squares. We've probably seen this formulation before. Mathematically, we are given a data matrix $A \in \mathbb{R}^{m \times n}$ and a vector of outcomes $\vec{y} \in \mathbb{R}^m$, and attempt to find a parameter vector $\vec{x} \in \mathbb{R}^n$ which minimizes the residual $\|A\vec{x} - \vec{y}\|_2^2$. Here $\|\cdot\|_2$ is the standard Euclidean norm $\|\vec{z}\|_2 \triangleq \sqrt{\vec{z}^\top \vec{z}} = \sqrt{\sum_{i=1}^n z_i^2}$; it is labeled with the 2 for a reason we will see later in the course.

More precisely, we attempt to solve the following optimization problem:

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 . \tag{1.9}$$

Theorem 1.4 (Least Squares Solution)**.** *Let $A \in \mathbb{R}^{m\times n}$ have full column rank, and let $\vec{y} \in \mathbb{R}^m$. Then the solution to (1.9), i.e., the solution to*

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 ,$$

is given by

$$\vec{x}^\star = (A^\top A)^{-1} A^\top \vec{y}. \tag{1.10}$$

Proof. The idea is to find $A\vec{x} \in \mathcal{R}(A)$ which is closest to $\vec{y}$. Here $\mathcal{R}(A)$ is the range, or column space, or column span, of A. In general, we have no guarantee that $\vec{y} \in \mathcal{R}(A)$, so there is not necessarily an $\vec{x}$ such that $A\vec{x} = \vec{y}$. Instead, we are finding an approximate solution to the equation $A\vec{x} = \vec{y}$.

Figure: *Geometry of least squares: $\mathcal{R}(A)$ subspace and $\vec{y}$ vector*

Recall that $\mathcal{R}(A)$ is a subspace, and that $\vec{y}$ itself may not belong to $\mathcal{R}(A)$. We can now solve this problem using ideas from geometry. We claim that the closest point to $\vec{y}$ contained in $\mathcal{R}(A)$ is the orthogonal projection of $\vec{y}$ onto $\mathcal{R}(A)$; call this point $\vec{z}$. Also, define $\vec{e} \triangleq \vec{y} - \vec{z}$.

Figure: *Projection diagram: $\vec{y}$, $\vec{z}$, $\vec{e}$, and $\mathcal{R}(A)$*

From this diagram, we see that $\vec{e}$ is orthogonal to any vector in $\mathcal{R}(A)$. But remember that we still have to prove that $\vec{z}$ is the closest point to $\vec{y}$ within $\mathcal{R}(A)$. To see this, consider any other point $\vec{u} \in \mathcal{R}(A)$ and define $\vec{v} \triangleq \vec{y} - \vec{u}$.

Figure: *Extended projection: $\vec{y}$, $\vec{z}$, $\vec{e}$, $\vec{u}$, $\vec{v}$, and $\mathcal{R}(A)$*

To complete our proof, we define $\vec{w} \triangleq \vec{z} - \vec{u}$, noting that the angle $\vec{u} \to \vec{z} \to \vec{y}$ is a right angle; in other words, $\vec{w}$ and $\vec{e}$ are orthogonal.

Figure: *Complete least squares geometry with all vectors*

By the Pythagorean theorem, we see that

$$\begin{aligned}
\|\vec{y} - \vec{u}\|_2^2 &= \|\vec{v}\|_2^2 && (1.11)\\
&= \|\vec{w}\|_2^2 + \|\vec{e}\|_2^2 && (1.12)\\
&= \underbrace{\|\vec{z} - \vec{u}\|_2^2}_{>0} + \|\vec{e}\|_2^2 && (1.13)\\
&> \|\vec{e}\|_2^2 && (1.14)\\
&= \|\vec{y} - \vec{z}\|_2^2 . && (1.15)
\end{aligned}$$

Therefore, $\vec{z}$ is the closest point to $\vec{y}$ within $\mathcal{R}(A)$.

Now, we want to find $\vec{z} \in \mathcal{R}(A)$, i.e., the orthogonal projection of $\vec{y}$ onto $\mathcal{R}(A)$, such that $\vec{e} = \vec{y} - \vec{z}$ is orthogonal to all vectors in $\mathcal{R}(A)$. By the definition of $\mathcal{R}(A)$, it's equivalent to find $\vec{x}^\star \in \mathbb{R}^n$ such that $\vec{y} - A\vec{x}^\star$ is orthogonal to all vectors in $\mathcal{R}(A)$. Since the columns of A form a spanning set for $\mathcal{R}(A)$, it's equivalent to find $\vec{x}^\star \in \mathbb{R}^n$ such that $\vec{y} - A\vec{x}^\star$ is orthogonal to all columns of A. This implies

$$\begin{aligned}
\vec{0} &= A^\top(\vec{y} - A\vec{x}^\star) && (1.16)\\
&= A^\top\vec{y} - A^\top A\vec{x}^\star && (1.17)\\
\implies A^\top A\vec{x}^\star &= A^\top\vec{y} && (1.18)\\
\implies \vec{x}^\star &= (A^\top A)^{-1}A^\top\vec{y}. && (1.19)
\end{aligned}$$

Here $A^\top A$ is invertible because A has full column rank. □

We'll conclude with a statistical application of least squares to linear regression. Suppose we are given data $(x_1, y_1), \ldots, (x_n, y_n)$, and want to fit an affine model $y = mx + b$ through these data points. This corresponds to approximately solving the system

$$\begin{aligned}
mx_1 + b &= y_1\\
mx_2 + b &= y_2\\
&\vdots\\
mx_n + b &= y_n. && (1.20)
\end{aligned}$$

Formulating it in terms of vectors and matrices, we have

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \tag{1.21}$$

In the case where the data is noisy or inconsistent with the model, as in the below figure, the linear system will be overdetermined and have no solutions. Then, we find an approximate solution – a line of best fit – via least squares on the above system.

Figure: *Scatter plot with line of best fit*

As a last note, solving least squares (and similar problems) is easy because it is a so-called *convex* problem. Convex problems are easy to solve because any local optimum is a global optimum, which allows us to use a variety of simple techniques to find global optima. It is generally much more difficult to solve non-convex problems, though we solve a few during this course.

We discuss much more about convexity and convex problems later in the course.

1.3 Solution Concepts and Notation

Sometimes we assign values to our optimization problems. For example in the framework of (1.5) we may write

$$\begin{aligned} p^\star = \min_{\vec{x}\in\mathbb{R}^n} \quad & f_0(\vec{x}) \\ \text{s.t.} \quad & f_i(\vec{x}) \leq 0 \quad \forall i \in \{1,\dots,m\} \\ & h_j(\vec{x}) = 0 \quad \forall j \in \{1,\dots,p\}. \end{aligned} \tag{1.22}$$

On the other hand, in the framework of (1.7) and using the definition of Ω in (1.6), we may write[1].

$$p^\star = \min_{\vec{x}\in\Omega} f_0(\vec{x}). \tag{1.23}$$

This means that $p^\star \in \mathbb{R}$ is the minimum value of f_0 over all $\vec{x} \in \Omega$; formally,

$$p^\star = \min_{\vec{x}\in\Omega} f_0(\vec{x}) \triangleq \min\{f_0(\vec{x}) \mid \vec{x} \in \Omega\}. \tag{1.24}$$

As an example, consider the two-element set $\Omega = \{0, 1\}$ and $f_0(x) = 3x^2 + 2$. Then $p^\star = \min\{f(0), f(1)\} = \min\{2, 5\} = 2$. We emphasize that $p^\star$ is a *real number*, not a vector.

[1]For the case where the minimum does not exist, but the infimum is finite, please see Section 1.4

To extract the minimizers, i.e., the points $\vec{x} \in \Omega$ which minimize $f_0(\vec{x})$, we use the argmin notation, which gives us the set of arguments which minimize our objective function. Formally, we define:

$$\operatorname*{argmin}_{\vec{x}\in\Omega} f_0(\vec{x}) \triangleq \left\{ \vec{x} \in \Omega \;\middle|\; f_0(\vec{x}) = \min_{\vec{u}\in\Omega} f_0(\vec{u}) \right\} \tag{1.25}$$

We can thus write the set of solutions to (1.5) as

$$\begin{aligned} \operatorname*{argmin}_{\vec{x}\in\mathbb{R}^n} \quad & f_0(\vec{x}) \\ \text{s.t.} \quad & f_i(\vec{x}) \leq 0 \quad \forall i \in \{1, \ldots, m\} \\ & h_j(\vec{x}) = 0 \quad \forall j \in \{1, \ldots, p\}. \end{aligned} \tag{1.26}$$

And, as just discussed, we can write the set of solutions to (1.7) as

$$\operatorname*{argmin}_{\vec{x}\in\Omega} f_0(\vec{x}). \tag{1.27}$$

We emphasize that the argmin is a *set of vectors*, any of which are an optimal solution, i.e., a minimizer, of the optimization problem at hand. It is possible for the argmin to contain 0 vectors (in which case the minimum value is not realized and the problem has no global optima), any positive number of vectors, or an infinite number of vectors.

Let us consider the same example as before. In particular, consider the two-element set $\Omega = \{0, 1\}$ and $f_0(x) = 3x^2 + 2$. Then $\operatorname{argmin}_{x\in\Omega} f_0(x) = \{0\}$. But, in different scenarios, the argmin can have zero elements; for example, if $f_0(x) = 3x$, then $\operatorname{argmin}_{x\in\mathbb{R}} f_0(x) = \emptyset$. And it can have multiple elements; for example, if $f_0(x) = 3x^2(x-1)^2$, then $\operatorname{argmin}_{x\in\mathbb{R}} f_0(x) = \{0, 1\}$. It can even have infinitely many elements; for example, if $f_0(x) = 0$, then $\operatorname{argmin}_{x\in\mathbb{R}} f_0(x) = \mathbb{R}$.

Though we must remember to keep in mind that technically argmin is a set, in the problems we study, it usually contains exactly one element. Thus, instead of writing, for example, $\vec{x}^\star \in \operatorname{argmin}_{\vec{x}\in\Omega} f_0(\vec{x})$, we may also write $\vec{x}^\star = \operatorname{argmin}_{\vec{x}\in\Omega} f_0(\vec{x})$. The former expression is technically more correct, but both usages are fine, if — and only if — the argmin in question contains exactly one element.

1.4 (OPTIONAL) Infimum Versus Minimum

There is one remaining issue with our formulation, which we can conceptually consider as a "corner case". What happens if the minimum does not exist? This may seem like a very esoteric case, yet one can construct a straightforward example, such as the following. We know that the minimum of any set of numbers must be contained in the set. But what happens if we try to find the minimum of the open interval $(0, 1)$? For any $x \in (0, 1)$ which we claim to be our minimum, we see that $\frac{x}{2}$ is also contained in $(0, 1)$ and is smaller than x, which is a contradiction to our claim. Thus the set $(0, 1)$ has no minimum.

It seems like 0 is a useful notion of "minimum" for this set — that is, it's the largest number which is $\leq$ all numbers in the set, i.e., its "greatest lower bound" — but it isn't contained in the set and thus cannot be the minimum. Fortunately, this notion of greatest lower bound of a set is formalized in real analysis as the concept of an "infimum", denoted inf. For our purposes, we can think of the infimum as a generalization of the minimum which takes care of these corner cases and always exists. When the minimum exists, it is always equal to the infimum.

Based on this discussion, we can write our optimization problems as

$$\begin{aligned} p^\star = \inf_{\vec{x}\in\mathbb{R}^n} \quad & f_0(\vec{x}) \\ \text{s.t.} \quad & f_i(\vec{x}) \leq 0 \quad \forall i \in \{1,\ldots,m\} \\ & h_j(\vec{x}) = 0 \quad \forall j \in \{1,\ldots,p\}. \end{aligned} \tag{1.28}$$

and

$$p^\star = \inf_{\vec{x}\in\Omega} f_0(\vec{x}). \tag{1.29}$$

However, the argmin retains the same definition. In fact, one can prove that if we replaced the min in the argmin definition (1.25) with inf, that this "new" argmin would be exactly equivalent in every case to the "old" argmin, which we use henceforth. The analogous quantity to infimum for maximization — that is, the appropriate generalization of max — is the supremum, denoted sup.

Interested readers are encouraged to consult a real analysis textbook for a more comprehensive coverage. Though we have gone over the technical details here, for the rest of the course we will omit them for simplicity, and stick to using min and max (meaning inf and sup when the minimum and maximum do not exist).

Chapter 2

Linear Algebra Review

References

- Boyd & Vandenberghe [1]: Appendix A
- Calafiore & El Ghaoui [2]: Chapters 2–5

2.1 Norms

2.1.1 Definitions

Definition 2.1 (Norm)**.** Let $\mathcal{V}$ be a vector space over $\mathbb{R}$. A function $f : \mathcal{V} \to \mathbb{R}$ is a **norm** if:

- **Positive definiteness**: $f(\vec{x}) \geq 0$ for all $\vec{x} \in \mathcal{V}$, and $f(\vec{x}) = 0$ if and only if $\vec{x} = \vec{0}$.
- **Positive homogeneity**: $f(\alpha\vec{x}) = |\alpha| f(\vec{x})$ for all $\alpha \in \mathbb{R}$ and $\vec{x} \in \mathcal{V}$.
- **Triangle inequality**: $f(\vec{x} + \vec{y}) \leq f(\vec{x}) + f(\vec{y})$ for all $\vec{x}, \vec{y} \in \mathcal{V}$.

We can check that the familiar Euclidean norm $\|\cdot\|_2 : \vec{x} \mapsto \sqrt{\sum_{i=1}^n x_i^2}$ satisfies these properties. A generalization of the Euclidean norm is the following very useful class of norms.

Definition 2.2 (ℓ^p Norms)**.** Let $1 \leq p < \infty$. The ℓ^p-norm on $\mathbb{R}^n$ is given by

$$\|\vec{x}\|_p \triangleq \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}. \tag{2.1}$$

The ℓ^∞-norm on $\mathbb{R}^n$ is given by

$$\|\vec{x}\|_\infty \triangleq \max_{i \in \{1,\ldots,n\}} |x_i|. \tag{2.2}$$

Example 2.3 (Examples of ℓ^p Norms)**.** (a) The Euclidean norm, given by $\|\vec{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$, is an ℓ^p-norm for $p = 2$. (This is why we gave the subscript 2 to the Euclidean norm previously).

(b) The ℓ^1-norm is given by $\|\vec{x}\|_1 = \sum_{i=1}^n |x_i|$.

(c) The ℓ^∞-norm, given by $\|\vec{x}\|_\infty = \max_{i\in\{1,\dots,n\}} |x_i|$, is the limit of the ℓ^p norms as $p \to \infty$:

$$\|\vec{x}\|_\infty = \lim_{p\to\infty} \|\vec{x}\|_p . \tag{2.3}$$

We do not prove this here; it is left as an exercise.

2.1.2 Inequalities

There are a variety of useful inequalities which are associated with the ℓ^p norms. Before we provide them, we will take a second to discuss the importance of inequalities for optimization.

A priori, it may not be clear why we need to care about inequalities; why does it matter whether one arrangement of variables is always greater or less than another arrangement? It turns out that such inequalities are very helpful for characterizing the minimum and maximum of a given set of things; we can obtain upper bounds and lower bounds for things using these inequalities. This is definitely very helpful for optimization.

With that out of the way, let us get to the first major inequality.

Theorem 2.4 (Cauchy-Schwarz Inequality)**.** *For any $\vec{x}, \vec{y} \in \mathbb{R}^n$, we have*

$$\left|\vec{x}^\top \vec{y}\right| \le \|\vec{x}\|_2 \|\vec{y}\|_2 . \tag{2.4}$$

Proof. Let θ be the angle between $\vec{x}$ and $\vec{y}$. We write

$$\left|\vec{x}^\top \vec{y}\right| = |\|\vec{x}\|_2 \|\vec{y}\|_2 \cos\theta| \tag{2.5}$$

$$= \|\vec{x}\|_2 \|\vec{y}\|_2 |\cos\theta| \tag{2.6}$$

$$\le \|\vec{x}\|_2 \|\vec{y}\|_2 . \tag{2.7}$$

□

We can get this result for ℓ^2 norms. A natural next question is whether we can generalize it to ℓ^p norms for $p \neq 2$. It turns out that we can, as we demonstrate shortly.

Theorem 2.5 (Hölder's Inequality)**.** *Let $1 \le p, q \le \infty$ such that $\frac{1}{p} + \frac{1}{q} = 1$.*[1] *Then for any $\vec{x}, \vec{y} \in \mathbb{R}^n$, we have*

$$\left|\vec{x}^\top \vec{y}\right| \le \sum_{i=1}^n |x_i y_i| \le \|\vec{x}\|_p \|\vec{y}\|_q . \tag{2.8}$$

This inequality collapses to the Cauchy-Schwarz Inequality when $p = q = 2$. The proof is out of scope for now since it uses convexity.

Example 2.6 (Dual Norms)**.** Fix $\vec{y} \in \mathbb{R}^n$. Let us solve the problem:

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_p \le 1}} \vec{x}^\top \vec{y}. \tag{2.9}$$

[1] Such pairs (p, q) are called *Hölder conjugates*.

It is initially difficult to see how to proceed, so let us simplify the problem to get back onto familiar territory. We start with $p = 2$, so that the problem becomes:

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_2\leq 1}} \vec{x}^\top \vec{y}. \tag{2.10}$$

Figure: *Unit ball for ℓ_2 norm with vector $\vec{y}$*

For any $\vec{x} \in \mathbb{R}^n$, and θ the angle between $\vec{x}$ and $\vec{y}$, we have

$$\vec{x}^\top \vec{y} = \|\vec{x}\|_2 \|\vec{y}\|_2 \cos\theta. \tag{2.11}$$

This term is maximized when $\cos\theta = 1$, or equivalently $\theta = 0$. Thus $\vec{x}$ and $\vec{y}$ must point in the same direction, i.e., $\vec{x}$ is a scalar multiple of $\vec{y}$. And since we want to maximize this dot product, we must choose $\vec{x}$ to maximize $\|\vec{x}\|_2$ subject to the constraint $\|\vec{x}\|_2 \leq 1$. Thus, we choose an $\vec{x}$ which has $\|\vec{x}\|_2 = 1$ and points in the same direction as $\vec{y}$. This gives $\vec{x}^\star = \vec{y}/\|\vec{y}\|_2$. Thus,

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_2\leq 1}} \vec{x}^\top \vec{y} = (\vec{x}^\star)^\top \vec{y} = \left(\frac{\vec{y}}{\|\vec{y}\|_2}\right)^\top \vec{y} = \frac{\vec{y}^\top \vec{y}}{\|\vec{y}\|_2} = \frac{\|\vec{y}\|_2^2}{\|\vec{y}\|_2} = \|\vec{y}\|_2 . \tag{2.12}$$

Now let us try $p = \infty$. The problem becomes

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_\infty\leq 1}} \vec{x}^\top \vec{y}. \tag{2.13}$$

Figure: *Unit ball for ℓ_∞ norm (square) with vector $\vec{y}$*

Motivated by this diagram, we see that the constraint $\|\vec{x}\|_\infty \leq 1$ is equivalent to the $2n$ constraints $-1 \leq x_i$ and $x_i \leq 1$. Also, writing out the objective function

$$\vec{x}^\top \vec{y} = \sum_{i=1}^{n} x_i y_i = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n, \tag{2.14}$$

we see that the problem is

$$\begin{aligned} \max_{\vec{x}\in\mathbb{R}^n} \quad & (x_1 y_1 + x_2 y_2 + \cdots + x_n y_n) \\ \text{s.t.} \quad & -1 \leq x_i \leq 1, \quad \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{2.15}$$

This problem has an interesting structure that will be repeated several times in the problems we discuss in this class. Namely, the objective function is the sum of several terms, each of which involves only one x_i. And the constraints are able to be partitioned into some groups, where the constraints in each group constrain only one x_i. Thus, this problem is *separable* into n different scalar problems, such that the optimal solutions for each scalar problem form an optimal solution for the vector problem. Namely, the problems are

$$\max_{\substack{x_i \in \mathbb{R} \\ -1 \leq x_i \leq 1}} \quad x_i y_i \tag{2.16}$$

We solve this much simpler problem by hand. If $y_i > 0$ then $x_i^\star = 1$; on the other hand, if $y_i \leq 0$ then $x_i^\star = -1$. To summarize, $x_i^\star = \operatorname{sign}(y_i)$, so that $x_i^\star y_i = |y_i|$.

Putting all the scalar problems together, we see that $\vec{x}^\star = \operatorname{sign}(\vec{y})$, and the vector problem's optimal value is given by

$$\max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_\infty \leq 1}} \vec{x}^\top \vec{y} = (\vec{x}^\star)^\top \vec{y} = \sum_{i=1}^n x_i^\star y_i = \sum_{i=1}^n \operatorname{sign}(y_i) y_i = \sum_{i=1}^n |y_i| = \|\vec{y}\|_1 . \tag{2.17}$$

As a final exercise, we consider $p = 1$, so that the problem becomes

$$\max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_1 \leq 1}} \vec{x}^\top \vec{y}. \tag{2.18}$$

Figure: *Unit ball for ℓ_1 norm (diamond) with vector $\vec{y}$*

We now bound the objective as

$$\vec{x}^\top \vec{y} \leq \left|\vec{x}^\top \vec{y}\right| \tag{2.19}$$

$$= \left|\sum_{i=1}^{n} x_i y_i\right| \tag{2.20}$$

$$\leq \sum_{i=1}^{n} |x_i y_i| \quad \text{by triangle inequality} \tag{2.21}$$

$$= \sum_{i=1}^{n} |x_i||y_i| \tag{2.22}$$

$$\leq \sum_{i=1}^{n} |x_i| \left(\max_{i\in\{1,\ldots,n\}} |y_i|\right) \tag{2.23}$$

$$= \left(\max_{i\in\{1,\ldots,n\}} |y_i|\right) \sum_{i=1}^{n} |x_i| \tag{2.24}$$

$$= \|\vec{y}\|_\infty \|\vec{x}\|_1 \tag{2.25}$$

$$\leq \|\vec{y}\|_\infty . \tag{2.26}$$

Thus we have

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_1\leq 1}} \vec{x}^\top \vec{y} \leq \|\vec{y}\|_\infty . \tag{2.27}$$

This inequality is actually an equality. To show this, we need to show the reverse inequality

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_1\leq 1}} \vec{x}^\top \vec{y} \geq \|\vec{y}\|_\infty . \tag{2.28}$$

And showing this inequality amounts to choosing, for our fixed $\vec{y}$, an $\vec{x}$ such that $\|\vec{x}\|_1 \leq 1$ and $\vec{x}^\top \vec{y} \geq \|\vec{y}\|_\infty$. This is also called "showing the maximum is attained". To do this, we can find an $\vec{x}$ such that $\|\vec{x}\|_p \leq 1$ and all the inequalities in the chain are met with equality.

To meet all the constraints, we can construct $\vec{x}^\star$ via the following process:

- For each $i \notin \operatorname{argmax}_{j\in\{1,\ldots,n\}} |y_j|$, set $\tilde{x}_i = 0$.
- For each $i \in \operatorname{argmax}_{j\in\{1,\ldots,n\}} |y_j|$, set $\tilde{x}_i = \operatorname{sign}(y_i)$.
- To get the true solution vector $\vec{x}^\star$, divide $\tilde{\vec{x}}$ by $\left\|\tilde{\vec{x}}\right\|_1$; that is, $\vec{x}^\star = \tilde{\vec{x}} / \left\|\tilde{\vec{x}}\right\|_1$. This ensures that $\|\vec{x}^\star\|_1 = 1$.

This $\vec{x}^\star$ "achieves the maximum", showing that

$$\max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_1\leq 1}} \vec{x}^\top \vec{y} = \|\vec{y}\|_\infty . \tag{2.29}$$

This notion where the ℓ^2-norm constraint leads to the ℓ^2-norm objective, the ℓ^∞-norm constraint leads to the ℓ^1-norm objective, and the ℓ^1-norm constraint leads to the ℓ^∞-norm objective, hints

at a greater pattern. Indeed, one can show that for $1 \leq p, q \leq \infty$ such that $\frac{1}{p} + \frac{1}{q} = 1$, an ℓ^p-norm constraint leads to an ℓ^q-norm objective:

$$\max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_p \leq 1}} \vec{x}^\top \vec{y} = \|\vec{y}\|_q . \tag{2.30}$$

As before, we can prove this equality by proving the two constituent inequalities:

$$\max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_p \leq 1}} \vec{x}^\top \vec{y} \leq \|\vec{y}\|_q \quad \text{and} \quad \max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_p \leq 1}} \vec{x}^\top \vec{y} \geq \|\vec{y}\|_q . \tag{2.31}$$

The proof of the first inequality ($\leq$) follows from applying Hölder's inequality to the objective function:

$$\max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_p \leq 1}} \vec{x}^\top \vec{y} \leq \max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_p \leq 1}} \|\vec{x}\|_p \|\vec{y}\|_q = \|\vec{y}\|_q \cdot \max_{\substack{\vec{x} \in \mathbb{R}^n \\ \|\vec{x}\|_p \leq 1}} \|\vec{x}\|_p = \|\vec{y}\|_q . \tag{2.32}$$

The second inequality ($\geq$) can follow if, for our fixed choice of $\vec{y}$, we produce some $\vec{x}$ such that $\|\vec{x}\|_p \leq 1$ and $\vec{x}^\top \vec{y} \geq \|\vec{y}\|_q$, i.e., "the maximum is attained". This is more complicated to do, and we won't do it here.

The above equality (2.30) means that the norms $\|\cdot\|_p$ and $\|\cdot\|_q$ are so-called *dual norms*. We will explore aspects of duality later in the course, though frankly we are just scratching the surface.

These problems, which are short and easy to state, contain a couple of core ideas within their solutions, which are broadly generalizable to a lot of optimization problems. For your convenience, we discuss these explicitly below.

Problem Solving Strategy (Separating Vector Problems into Scalar Problems). *When trying to simplify an optimization problem, try to see if you can simplify it into several independent scalar problems. Then solve each scalar problem — this is usually much easier than solving the whole vector problem at once. The optimal solutions to each scalar problem will then form the optimal solution to the whole vector problem.*

Problem Solving Strategy (Proving Optimality in an Optimization Problem). *To solve an optimization problem, you can use inequalities to bound the objective function, and then try to show that this bound is tight by finding a feasible choice of optimization variable which makes all the inequalities into equalities.*

2.2 Gram-Schmidt and QR Decomposition

The Gram-Schmidt algorithm is a way to turn a linearly independent set $\{\vec{a}_1, \ldots, \vec{a}_k\}$ of vectors into an orthonormal set $\{\vec{q}_1, \ldots, \vec{q}_k\}$ which spans the same space. To reiterate, an orthonormal set is a set of vectors in which each vector has norm 1 and is orthogonal to all others in the basis.

Suppose for simplicity that $n = k = 2$, and that we have the following vectors.

Figure: *Gram-Schmidt Step 1: Two vectors $\vec{a}_1$ and $\vec{a}_2$*

We begin with $\vec{a}_1$. We want to construct a vector $\vec{q}_1$ such that

- it's orthogonal to all the $\vec{q}_i$ which came before it — which is none of them, so we don't have to worry; and
- it has unit norm, so $\|\vec{q}_1\|_2 = 1$.

To achieve this, the simplest choice is

$$\vec{q}_1 \triangleq \frac{\vec{a}_1}{\|\vec{a}_1\|_2}. \tag{2.33}$$

Figure: *Gram-Schmidt Step 2: $\vec{a}_1$, $\vec{a}_2$, and $\vec{q}_1$*

Then we go to $\vec{a}_2$. To find $\vec{q}_2$ which is orthogonal to all the $\vec{q}_i$ before it — that is, $\vec{q}_1$ — we subtract off the orthogonal projection of $\vec{a}_2$ onto $\vec{q}_1$ from $\vec{a}_2$. The orthogonal projection of $\vec{a}_2$ onto $\vec{q}_1$ is given by

$$\vec{p}_2 \triangleq \vec{q}_1(\vec{q}_1^\top \vec{a}_2) \tag{2.34}$$

and so the projection residual is given by

$$\vec{s}_2 \triangleq \vec{a}_2 - \vec{p}_2 = \vec{a}_2 - \vec{q}_1(\vec{q}_1^\top \vec{a}_2). \tag{2.35}$$

Note that these formulas only hold because $\vec{q}_1$ is normalized, i.e., has norm 1.

Figure: *Gram-Schmidt Step 3: $\vec{a}_1$, $\vec{a}_2$, $\vec{q}_1$, $\vec{p}_2$, $\vec{s}_2$*

While $\vec{s}_2$ is orthogonal to $\vec{q}_1$, because we want a $\vec{q}_2$ that is normalized, we normalize $\vec{s}_2$ to get $\vec{q}_2$:

$$\vec{q}_2 \triangleq \frac{\vec{s}_2}{\|\vec{s}_2\|_2}. \tag{2.36}$$

Figure: Complete Gram-Schmidt orthogonalization process

If we had a vector $\vec{q}_3$ (and weren't limited by drawing in 2D space), we would ensure that $\vec{q}_3$ were orthogonal to $\vec{q}_1$ and $\vec{q}_2$, as well as normalized, in a similar way as before. First we would compute the projection

$$\vec{p}_3 \triangleq \vec{q}_1(\vec{q}_1^\top \vec{a}_3) + \vec{q}_2(\vec{q}_2^\top \vec{a}_3). \tag{2.37}$$

and the residual

$$\vec{s}_3 \triangleq \vec{a}_3 - \vec{p}_3 = \vec{a}_3 - \vec{q}_1(\vec{q}_1^\top \vec{a}_3) - \vec{q}_2(\vec{q}_2^\top \vec{a}_3). \tag{2.38}$$

These projection formulas only hold because $\{\vec{q}_1, \vec{q}_2\}$ is an orthonormal set. And then we could compute

$$\vec{q}_3 \triangleq \frac{\vec{s}_3}{\|\vec{s}_3\|_2}. \tag{2.39}$$

And so on. The general algorithm goes similar.

Algorithm 1 Gram-Schmidt algorithm.

1. **function** GramSchmidtAlgorithm(linearly independent set $\{\vec{a}_1, \ldots, \vec{a}_k\}$)
2. $\quad \vec{q}_1 \triangleq \vec{a}_1 / \|\vec{a}_1\|_2$.
3. $\quad$ **for** $i \in \{2, 3, \ldots, k\}$ **do**
4. $\quad\quad \vec{p}_i \triangleq \sum_{j=1}^{i-1} \vec{q}_j(\vec{q}_j^\top \vec{a}_i)$
5. $\quad\quad \vec{s}_i \triangleq \vec{a}_i - \vec{p}_i$
6. $\quad\quad \vec{q}_i \triangleq \vec{s}_i / \|\vec{s}_i\|_2$
7. $\quad$ **end for**
8. $\quad$ **return** orthonormal set $\{\vec{q}_1, \ldots, \vec{q}_k\}$
9. **end function**

This algorithm has the following two properties, which you can formally prove as an exercise.

Theorem 2.7 (Gram-Schmidt Algorithm)**.** *Algorithm 1 has the following properties:*

1. *For each $i \in \{1, \ldots, k\}$, we have*

$$span(\vec{a}_1, \ldots, \vec{a}_i) = span(\vec{q}_1, \ldots, \vec{q}_i). \tag{2.40}$$

   In particular, $\{\vec{a}_1, \ldots, \vec{a}_k\}$ spans the same subspace as $\{\vec{q}_1, \ldots, \vec{q}_k\}$, as was stated in our original goal.

2. *$\{\vec{q}_1, \ldots, \vec{q}_k\}$ is an orthonormal set.*

The Gram-Schmidt algorithm leads to something called the QR decomposition. Because, for each i, we have $\text{span}(\vec{a}_1, \ldots, \vec{a}_i) = \text{span}(\vec{q}_1, \ldots, \vec{q}_i)$, it means that we can write $\vec{a}_i$ as a linear combination of the $\vec{q}_j$:

$$\vec{a}_i = r_{1i}\vec{q}_1 + r_{2i}\vec{q}_2 + \cdots + r_{ii}\vec{q}_i = \sum_{j=1}^{i} r_{ji}\vec{q}_j \tag{2.41}$$

Putting all the k equations in a matrix form, we can write

$$\begin{bmatrix} \vec{a}_1 & \cdots & \vec{a}_k \end{bmatrix} = \begin{bmatrix} \vec{q}_1 & \cdots & \vec{q}_k \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ 0 & r_{22} & \cdots & r_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{kk} \end{bmatrix}. \tag{2.42}$$

More generally, we can decompose every tall matrix with full column rank into a product of a tall matrix with orthonormal columns Q and an upper-triangular matrix R.

Theorem 2.8 (QR Decomposition)**.** *Let $A \in \mathbb{R}^{n\times k}$ where $k \leq n$ (so A is tall). Suppose A has full column rank. Then there is a matrix $Q \in \mathbb{R}^{n\times k}$ with orthonormal columns, and a matrix $R \in \mathbb{R}^{k\times k}$ which is upper triangular, such that $A = QR$.*

As a final note, there are various alterations to the QR decomposition that work for matrices which are wide and/or do not have full column rank. Those are out of scope, but the idea is the same.

The QR decomposition is also relevant in numerical linear algebra, where it can be used to solve tall linear systems $A\vec{x} = \vec{y}$ efficiently, especially if the underlying matrix A has special structure. All such connections are out of scope.

2.3 Fundamental Theorem of Linear Algebra

The fundamental theorem of linear algebra is a tool for understanding what happens to vectors and vector spaces under a linear transformation. Matrix multiplication transforms one vector space into another. This is helpful for allowing us to change our coordinate system, which tells us more about the problem.

Definition 2.9 (Direct Sum)**.** Let $U, V \subseteq \mathbb{R}^n$ be subspaces. We say that U and V *direct sum* to $\mathbb{R}^n$, denoted $U \oplus V = \mathbb{R}^n$, if and only if:

- Every vector $\vec{x} \in \mathbb{R}^n$ can be written as $\vec{x} = \vec{x}_1 + \vec{x}_2$, where $\vec{x}_1 \in U$ and $\vec{x}_2 \in V$.
- Furthermore, this decomposition is unique, in the sense that if $\vec{x} = \vec{x}_1 + \vec{x}_2 = \vec{y}_1 + \vec{y}_2$ are two instances of the above decomposition, then $\vec{x}_1 = \vec{y}_1$ and $\vec{x}_2 = \vec{y}_2$.

Theorem 2.10 (Fundamental Theorem of Linear Algebra)**.** *Let $A \in \mathbb{R}^{m\times n}$. Then*

$$\mathcal{N}(A) \oplus \mathcal{R}(A^\top) = \mathbb{R}^n. \tag{2.43}$$

Note that we cannot replace $\mathcal{R}(A^\top)$ by $\mathcal{R}(A)$, since vectors in $\mathcal{R}(A)$ and $\mathcal{N}(A)$ do not even have the same number of entries or lie in the same Euclidean space. If we want to make a statement about $\mathcal{R}(A)$, we can replace A by $A^\top$ in the above theorem to get the following corollary.

Corollary 2.11. *Let $A \in \mathbb{R}^{m\times n}$. Then*

$$\mathcal{N}(A^\top) \oplus \mathcal{R}(A) = \mathbb{R}^m. \tag{2.44}$$

To prove the fundamental theorem of linear algebra, we use a tool called the orthogonal decomposition theorem.

Definition 2.12 (Orthogonal Complement)**.** Let $S \subseteq \mathbb{R}^n$ be a subspace. The *orthogonal complement* of S, denoted $S^\perp$, is

$$S^\perp \triangleq \{\vec{x} \in \mathbb{R}^n \mid \vec{s}^\top \vec{x} = 0 \text{ for all } \vec{s} \in S\} \tag{2.45}$$

Theorem 2.13 (Orthogonal Decomposition Theorem)**.** *Let $S \subseteq \mathbb{R}^n$ be a subspace. Then*

$$S \oplus S^\perp = \mathbb{R}^n. \tag{2.46}$$

Proof. To prove this, we first need to prove the following claim:

Let $U, V \subseteq \mathbb{R}^n$ be subspaces. Then $U \oplus V = \mathbb{R}^n$ if and only if every vector $\vec{x} \in \mathbb{R}^n$ can be written as $\vec{x} = \vec{x}_1 + \vec{x}_2$, where $\vec{x}_1 \in U$ and $\vec{x}_2 \in V$, and $U \cap V = \{\vec{0}\}$.

To prove this claim, suppose first that $U \oplus V = \mathbb{R}^n$. Then every vector $\vec{x} \in \mathbb{R}^n$ can be written as $\vec{x} = \vec{x}_1 + \vec{x}_2$, where $\vec{x}_1 \in U$ and $\vec{x}_2 \in V$. It remains to prove that $U \cap V = \{\vec{0}\}$. Suppose for the sake of contradiction that there exists $\vec{y} \neq \vec{0}$ such that $\vec{y} \in U \cap V$. Then

$$\vec{x} = (\vec{x}_1 + \vec{y}) + (\vec{x}_2 - \vec{y}). \tag{2.47}$$

Since $\vec{y} \in U$, we have $\vec{x}_1 + \vec{y} \in U$; since $\vec{y} \in V$, we have $\vec{x}_2 - \vec{y} \in V$. Thus

$$\vec{x} = \vec{x}_1 + \vec{x}_2 = (\vec{x}_1 + \vec{y}) + (\vec{x}_2 - \vec{y}) \tag{2.48}$$

are two distinct ways to write $\vec{x}$ as the sum of vectors from U and V, so it cannot be true that $U \oplus V = \mathbb{R}^n$, a contradiction.

Towards the other direction, suppose that every vector $\vec{x} \in \mathbb{R}^n$ can be written as $\vec{x} = \vec{x}_1 + \vec{x}_2$, where $\vec{x}_1 \in U$ and $\vec{x}_2 \in V$, and $U \cap V = \{\vec{0}\}$. The only thing remaining to prove is that if $\vec{x} = \vec{x}_1 + \vec{x}_2 = \vec{z}_1 + \vec{z}_2$ where $\vec{x}_1, \vec{z}_1 \in U$ and $\vec{x}_2, \vec{z}_2 \in V$, then we must have $\vec{x}_1 = \vec{z}_1$ and $\vec{x}_2 = \vec{z}_2$. Suppose again for the sake of contradiction that there exists $\vec{x} \in \mathbb{R}^n$, $\vec{x}_1, \vec{z}_1 \in U$, and $\vec{x}_2, \vec{z}_2 \in V$ such that $\vec{x} = \vec{x}_1 + \vec{x}_2 = \vec{z}_1 + \vec{z}_2$ but $\vec{x}_1 \neq \vec{z}_1$ or $\vec{x}_2 \neq \vec{z}_2$. Then we have

$$\vec{0} = \vec{x} - \vec{x} = \vec{x}_1 + \vec{x}_2 - \vec{z}_1 - \vec{z}_2 = (\vec{x}_1 - \vec{z}_1) + (\vec{x}_2 - \vec{z}_2). \tag{2.49}$$

Thus, we have that $\vec{x}_1 - \vec{z}_1 = \vec{z}_2 - \vec{x}_2 \neq \vec{0}$. Since $\vec{x}_1, \vec{z}_1 \in U$, we have $\vec{x}_1 - \vec{z}_1 \in U$, and since $\vec{x}_2, \vec{z}_2 \in V$, we have $\vec{z}_2 - \vec{x}_2 \in V$. Since they are equal, we have $\vec{x}_1 - \vec{z}_1 \in U \cap V$ and nonzero. Thus $U \cap V \neq \{\vec{0}\}$, a contradiction.

This proves the above claim. Now to prove the actual theorem, we note that every vector $\vec{x} \in \mathbb{R}^n$ can be written as

$$\vec{x} = \text{proj}_S(\vec{x}) + (\vec{x} - \text{proj}_S(\vec{x})). \tag{2.50}$$

By definition, $\text{proj}_S(\vec{x}) \in S$, and because the projection residual is orthogonal to the subspace, we have $\vec{x} - \text{proj}_S(\vec{x}) \in S^\perp$. Thus every vector in $\mathbb{R}^n$ can be written as the sum of a vector in S and $S^\perp$. It is an exercise to show that $S \cap S^\perp = \{\vec{0}\}$. Invoking the quoted claim completes the proof. □

Using this theorem, the only thing we need to show to prove the fundamental theorem of linear algebra is that $\mathcal{N}(A)$ and $\mathcal{R}(A^\top)$ are orthogonal complements. We do this below.

Proof of Theorem 2.10. By the Orthogonal Decomposition Theorem, the only thing we need to show is that $\mathcal{N}(A) = \mathcal{R}(A^\top)^\perp$. This is a set equality; we show it by showing that $\mathcal{N}(A) \subseteq \mathcal{R}(A^\top)^\perp$ and that $\mathcal{N}(A) \supseteq \mathcal{R}(A^\top)^\perp$.

We first want to show that $\mathcal{N}(A) \subseteq \mathcal{R}(A^\top)^\perp$. That is, we want to show that for any $\vec{x} \in \mathcal{N}(A)$ we have $\vec{x} \in \mathcal{R}(A^\top)^\perp$. That is, for any $\vec{y} \in \mathcal{R}(A^\top)$, we want to show that $\vec{y}^\top \vec{x} = 0$.

Since $\vec{y} \in \mathcal{R}(A^\top)$ we can write $\vec{y} = A^\top \vec{w}$ for some $\vec{w} \in \mathbb{R}^m$. Then, since $\vec{x} \in \mathcal{N}(A)$ we have $A\vec{x} = \vec{0}$, so

$$\begin{aligned} \vec{y}^\top \vec{x} &= (A^\top \vec{w})^\top \vec{x} && (2.51)\\ &= \vec{w}^\top A\vec{x} && (2.52)\\ &= \vec{w}^\top \vec{0} && (2.53)\\ &= 0. && (2.54) \end{aligned}$$

Thus $\vec{x}$ and $\vec{y}$ are orthogonal, so $\vec{x} \in \mathcal{R}(A^\top)^\perp$, which shows that $\mathcal{N}(A) \subseteq \mathcal{R}(A^\top)^\perp$.

We now want to show that $\mathcal{R}(A^\top)^\perp \subseteq \mathcal{N}(A)$. That is, we want to show that for any $\vec{x} \in \mathcal{R}(A^\top)^\perp$, we want to show that $\vec{x} \in \mathcal{N}(A)$. That is, we want to show that $A\vec{x} = \vec{0}$.

By definition, for every $\vec{y} \in \mathcal{R}(A^\top)$, we have $\vec{y}^\top \vec{x} = 0$. By writing $\vec{y} = A^\top \vec{w}$ for arbitrary $\vec{w} \in \mathbb{R}^m$, we get that for every $\vec{w} \in \mathbb{R}^m$ we have $(A^\top \vec{w})^\top \vec{x} = 0$. But the left-hand side is $\vec{w}^\top A\vec{x}$, so we have that $\vec{w}^\top A\vec{x} = 0$ for every $\vec{w} \in \mathbb{R}^m$. This is true for all $\vec{w} \in \mathbb{R}^m$, so it is true for the specific choice of $\vec{w} = A\vec{x}$, which yields

$$\begin{aligned} 0 &= \vec{w}^\top A\vec{x} && (2.55)\\ &= (A\vec{x})^\top A\vec{x} && (2.56)\\ &= \|A\vec{x}\|_2^2 && (2.57)\\ \implies A\vec{x} &= \vec{0}. && (2.58) \end{aligned}$$

This implies that $\vec{x} \in \mathcal{N}(A)$ as desired, so $\mathcal{R}(A^\top)^\perp \subseteq \mathcal{N}(A)$.

Thus, we have shown that $\mathcal{N}(A) = \mathcal{R}(A^\top)^\perp$, and so by the Orthogonal Decomposition Theorem we have $\mathcal{N}(A) \oplus \mathcal{R}(A^\top) = \mathbb{R}^n$. □

This will help us solve a very important optimization problem, which is considered "dual" to least squares in some sense. Recall that least squares helps us find an approximate solution to the linear system $A\vec{x} = \vec{y}$, when A is a tall matrix with full column rank. In other words, the linear system is over-determined, there are many more equations than unknowns, and there are generally no exact solutions, so we pick the solution with minimum squared error.

What about when A is a wide matrix with full row rank? There are now more unknowns than equations, and infinitely many exact solutions. So how do we pick one solution in particular? It really depends on which engineering problem we are solving. One common solution is to pick the

minimum-energy or minimum-norm problem, which is the solution to the optimization problem:

$$\min_{\vec{x}\in\mathbb{R}^n} \|\vec{x}\|_2^2 \quad \text{s.t.} \quad A\vec{x} = \vec{y}. \tag{2.59}$$

Note that this principle of choosing the smallest or simplest solution — the "Occam's Razor" principle — is much more broadly generalized beyond the case of finding solutions to linear systems, and is used within control theory and machine learning. But we deal with just this linear system case for now.

Theorem 2.14 (Minimum-Norm Solution)**.** *Let $A \in \mathbb{R}^{m\times n}$ have full row rank, and let $\vec{y} \in \mathbb{R}^m$. Then the solution to Equation (2.59) is given by*

$$\vec{x}^\star = A^\top (AA^\top)^{-1}\vec{y}. \tag{2.60}$$

Proof. Observe that the constraint $A\vec{x} = \vec{y}$ under-specifies the $\vec{x}$ — in particular, any component of $\vec{x}$ in $\mathcal{N}(A)$ will not affect the constraint and only the objective. In this sense, it is "wasteful", and we should intuitively remove it. This motivates using Theorem 2.10 to decompose $\vec{x}$ into a component inside $\mathcal{N}(A)$ — which we want to remove — and a component inside $\mathcal{R}(A^\top)$ — which we will optimize over.

Indeed, write $\vec{x} = \vec{u} + \vec{v}$, where $\vec{u} \in \mathcal{N}(A)$ and $\vec{v} \in \mathcal{R}(A^\top)$. Thus, there exists $\vec{w} \in \mathbb{R}^m$ such that $\vec{v} = A^\top \vec{w}$. The constraint becomes

$$\begin{aligned}
\vec{y} &= A\vec{x} && (2.61)\\
&= A(\vec{u} + \vec{v}) && (2.62)\\
&= A\vec{u} + A\vec{v} && (2.63)\\
&= \vec{0} + AA^\top \vec{w} && (2.64)\\
&= AA^\top \vec{w}. && (2.65)
\end{aligned}$$

And the objective function becomes

$$\begin{aligned}
\|\vec{x}\|_2^2 &= \|\vec{u} + \vec{v}\|_2^2 && (2.66)\\
&= \vec{u}^\top \vec{u} + 2\vec{u}^\top \vec{v} + \vec{v}^\top \vec{v} && (2.67)\\
&= \|\vec{u}\|_2^2 + 2\vec{v}^\top \vec{u} + \|\vec{v}\|_2^2 && (2.68)\\
&= \|\vec{u}\|_2^2 + 2(A^\top \vec{w})^\top \vec{u} + \|\vec{v}\|_2^2 && (2.69)\\
&= \|\vec{u}\|_2^2 + 2\vec{w}^\top A\vec{u} + \|\vec{v}\|_2^2 && (2.70)\\
&= \|\vec{u}\|_2^2 + 2\vec{w}^\top \vec{0} + \|\vec{v}\|_2^2 && (2.71)\\
&= \|\vec{u}\|_2^2 + \|\vec{v}\|_2^2 && (2.72)\\
&= \|\vec{u}\|_2^2 + \left\|A^\top \vec{w}\right\|_2^2 && (2.73)
\end{aligned}$$

Thus, the minimum-norm problem can be reformulated in terms of $\vec{u}$ and $\vec{w}$:

$$\min_{\substack{\vec{u}\in\mathbb{R}^n \\ \vec{w}\in\mathbb{R}^m}} \quad \|\vec{u}\|_2^2 + \left\|A^\top \vec{w}\right\|_2^2 \tag{2.74}$$

$$\text{s.t.} \quad \vec{y} = AA^\top \vec{w} \tag{2.75}$$

$$A\vec{u} = \vec{0}. \tag{2.76}$$

Now, because A has full row rank, $AA^\top$ is invertible, so the first constraint implies that $\vec{w}^\star = (AA^\top)^{-1}\vec{y}$, so $\vec{v}^\star = A^\top \vec{w}^\star = A^\top (AA^\top)^{-1}\vec{y}$. And because we are trying to minimize the objective, which only involves $\vec{u}$ through $\|\vec{u}\|_2^2$, the ideal solution is to set $\vec{u}^\star = \vec{0}$, which also satisfies the second constraint and so is feasible. Thus $\vec{x}^\star = \vec{v}^\star = A^\top (AA^\top)^{-1}\vec{y}$ as desired. □

2.4 Symmetric Matrices

Symmetric matrices are a sub-class of matrices which have many special properties, and in engineering applications one usually tries to work with symmetric matrices as much as possible.

Definition 2.15 (Symmetric Matrix)**.** Let $A \in \mathbb{R}^{n\times n}$ be a square matrix. We say that A is *symmetric* if $A = A^\top$. The set of all symmetric matrices is denoted $\mathbb{S}^n$.

Equivalently, $A_{ij} = A_{ji}$ for all i and j.

Example 2.16. The 2×2 matrix $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is symmetric.

Example 2.17 (Covariance Matrices)**.** Any matrix of the form $A = BB^\top$, such as the covariance matrices we will discuss in the next section, is a symmetric matrix, since

$$A^\top = (BB^\top)^\top = (B^\top)^\top (B)^\top = BB^\top = A. \tag{2.77}$$

Example 2.18 (Adjacency Matrix)**.** Consider an undirected connected graph $G = (V, E)$. Its adjacency matrix A has coordinate $A_{ij} = 1$ if $(i, j) \in E$, and $A_{ij} = 0$ otherwise. Since the graph is undirected, $(i, j) \in E$ if and only if $(j, i) \in E$, so $A_{ij} = A_{ji}$, and so A is a symmetric matrix.

Why do we care about symmetric matrices? Symmetric matrices have two nice properties: real eigenvalues, and guaranteed diagonalizability.

In general, a (non-symmetric) matrix need not be diagonalizable. For example, the matrix $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ is not diagonalizable. How can we characterize the diagonalizability of a matrix, then?

First, we will need the following definitions.

Definition 2.19 (Multiplicities)**.** Let $A \in \mathbb{R}^{n\times n}$, and let λ be an eigenvalue of A.

(a) The *algebraic multiplicity* μ of eigenvalue λ in A is the number of times λ is a root of the characteristic polynomial $p_A(x) \triangleq \det(xI - A)$ of A, i.e., it is the power of $(x - \lambda)$ in the factorization of $p_A(x)$.

(b) The *geometric multiplicity* ϕ of eigenvalue λ in A is the dimension of the null space $\Phi \triangleq \mathcal{N}(\lambda I - A)$.

Theorem 2.20 (Diagonalizability)**.** *A square matrix $A \in \mathbb{R}^{n\times n}$ is diagonalizable if and only if every eigenvalue of A has equal algebraic and geometric multiplicities.*

Example 2.21 (Multiplicities of Degenerate Matrix)**.** We were earlier told that the matrix $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ is not diagonalizable. To check this, let us compute its eigenvalues, algebraic multiplicities, and geometric multiplicities.

First, its characteristic polynomial is

$$p_A(x) = \det(xI - A) \tag{2.78}$$

$$= \det\left(\begin{bmatrix} x-1 & -1 \\ 0 & x-1 \end{bmatrix}\right) \tag{2.79}$$

$$= (x-1)^2. \tag{2.80}$$

Thus, A has only one eigenvalue $\lambda = 1$. Since $(x-1)$ has power 2 in the factorization of p_A, the eigenvalue $\lambda = 1$ has algebraic multiplicity $\mu = 2$.

The corresponding null space is

$$\Phi = \mathcal{N}(\lambda I - A) \tag{2.81}$$

$$= \mathcal{N}\left(\begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}\right) \tag{2.82}$$

$$= \operatorname{span}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \tag{2.83}$$

which has dimension $\phi = 1$. Thus, for $\lambda = 1$, we have $\mu \neq \phi$ and the matrix is indeed not diagonalizable.

This allows us to formally state the spectral theorem.

Theorem 2.22 (Spectral Theorem)**.** *Let $A \in \mathbb{S}^n$ have eigenvalues λ_i with algebraic multiplicities μ_i, eigenspaces $\Phi_i \triangleq \mathcal{N}(\lambda_i I - A)$, and geometric multiplicities $\phi_i \triangleq \dim(\Phi_i)$.*

(a) All eigenvalues are real: $\lambda_i \in \mathbb{R}$ for each i.

(b) Eigenspaces corresponding to different eigenvalues are orthogonal: Φ_i and Φ_j are orthogonal subspaces, i.e., for every $\vec{p}_i \in \Phi_i$ and $\vec{p}_j \in \Phi_j$ we have $\vec{p}_i^{\top}\vec{p}_j = 0$.

(c) A is diagonalizable: $\mu_i = \phi_i$ for each i.

(d) A is orthonormally diagonalizable*; there exists an* orthonormal *matrix $U \in \mathbb{R}^{n\times n}$ and* diagonal *matrix $\Lambda \in \mathbb{R}^{n\times n}$ such that $A = U\Lambda U^{\top}$.*

Recall that orthonormal matrices are matrices whose columns are orthonormal, i.e., are pairwise orthogonal and unit-norm. Orthonormal matrices U have the nice property that $U^{\top}U = I$, and if U is square, then $U^{\top} = U^{-1}$.

One nice thing about diagonalization is that we can read off the eigenvalues and eigenvectors from the components of the diagonalization.

Theorem 2.23. *Let $A \in \mathbb{S}^n$ have orthonormal diagonalization $A = U\Lambda U^\top$, where $U = \begin{bmatrix}\vec{u}_1 & \cdots & \vec{u}_n\end{bmatrix} \in \mathbb{R}^{n\times n}$ is square orthonormal, and $\Lambda = diag(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ is diagonal. Then for each i, the pair $(\lambda_i, \vec{u}_i)$ is an eigenvalue-eigenvector pair for A.*

Using this, we can work with another nice property of the orthonormal diagonalization. Namely, we can read off bases for $\mathcal{N}(A)$ and $\mathcal{R}(A)$. That is, a basis for $\mathcal{N}(A)$ is the set of eigenvectors $\vec{u}_i$ corresponding to the eigenvalues λ_i of A which are equal to 0. Since U is orthonormal, the remaining eigenvectors $\vec{u}_i$ span the orthogonal complement to $\mathcal{N}(A)$. But by the fundamental theorem of linear algebra, we have $\mathcal{N}(A)^\perp = \mathcal{R}(A^\top) = \mathcal{R}(A)$, so these eigenvectors form a basis for $\mathcal{R}(A)$.

Before we get into those, we will first state and solve a quick optimization problem which yields the eigenvalues of a symmetric matrix. This optimization problem turns out to be quite useful for further study of optimization.

Theorem 2.24 (Variational Characterization of Eigenvalues)**.** *Let $A \in \mathbb{S}^n$. Let $\lambda_{\min}\{A\}$ and $\lambda_{\max}\{A\}$ be the maximum and minimum eigenvalues of A (which is well-defined since by the spectral theorem, all eigenvalues of A are real). Then*

$$\lambda_{\max}\{A\} = \max_{\substack{\vec{x}\in\mathbb{R}^n \\ \vec{x}\neq\vec{0}}} \frac{\vec{x}^\top A\vec{x}}{\vec{x}^\top\vec{x}} = \max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_2=1}} \vec{x}^\top A\vec{x} \tag{2.84}$$

$$\lambda_{\min}\{A\} = \min_{\substack{\vec{x}\in\mathbb{R}^n \\ \vec{x}\neq\vec{0}}} \frac{\vec{x}^\top A\vec{x}}{\vec{x}^\top\vec{x}} = \min_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_2=1}} \vec{x}^\top A\vec{x}. \tag{2.85}$$

The term $\frac{\vec{x}^\top A\vec{x}}{\vec{x}^\top\vec{x}}$ is called the Rayleigh quotient *of A; it is a function of $\vec{x} \in \mathbb{R}^n$.*

This characterization motivates defining a new sub-class (or really several new sub-classes) of matrices.

Definition 2.25 (Positive Semidefinite and Positive Definite Matrices)**.** Let $A \in \mathbb{S}^n$. We say that A is *positive semidefinite* (PSD), denoted $A \in \mathbb{S}^n_+$, if $\vec{x}^\top A\vec{x} \geq 0$ for all $\vec{x}$. We say that A is *positive definite* (PD), denoted $A \in \mathbb{S}^n_{++}$, if $\vec{x}^\top A\vec{x} > 0$ for all nonzero $\vec{x}$.

There are also negative semidefinite (NSD) and negative definite (ND) symmetric matrices, defined analogously. There are also indefinite symmetric matrices, which are none of the above. It is clear to see that PD matrices are themselves PSD.

Theorem 2.26. *We have $A \in \mathbb{S}^n_+$ if and only if each eigenvalue of A is non-negative. Also, $A \in \mathbb{S}^n_{++}$ if and only if each eigenvalue of A is positive.*

The final construction we discuss is that of the positive semidefinite square root.

Theorem 2.27. *Let $A \in \mathbb{S}^n_+$. Then there exists a unique symmetric PSD matrix $B \in \mathbb{S}^n_+$, usually denoted $B = A^{1/2}$, such that $A = B^2$.*

2.5 Principal Component Analysis

Principal components analysis is a way to recover the eponymous principal components of the data. These principal components are those that are most representative of the data structure. Formally, if we have data in $\mathbb{R}^d$, we want to find an underlying p-dimensional linear structure, where $p \ll d$.

This idea has many, many use cases. For example, in modern machine learning, most data has thousands or millions of dimensions. In order to visualize it properly, we need to reduce its dimension to a reasonable number, in order to get an idea about the underlying structure of the data.

Let us first lay out some notation and definitions. Suppose we have the data points $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$. We organize these into a *data matrix* X where data points form the rows:

$$X \triangleq \begin{bmatrix} \vec{x}_1^\top \\ \vdots \\ \vec{x}_n^\top \end{bmatrix} \in \mathbb{R}^{n \times d} \quad \text{so that} \quad X^\top = \begin{bmatrix} \vec{x}_1 & \cdots & \vec{x}_n \end{bmatrix} \in \mathbb{R}^{d \times n}. \tag{2.86}$$

We define the *covariance matrix* $C \in \mathbb{R}^{d \times d}$ by

$$C \triangleq \frac{1}{n} X^\top X = \frac{1}{n} \begin{bmatrix} \vec{x}_1 & \cdots & \vec{x}_n \end{bmatrix} \begin{bmatrix} \vec{x}_1^\top \\ \vdots \\ \vec{x}_n^\top \end{bmatrix} = \frac{1}{n} \sum_{i=1}^n \vec{x}_i \vec{x}_i^\top. \tag{2.87}$$

We see that C is symmetric since $X^\top X$ is symmetric, so really $C \in \mathbb{S}^d$.

We now discuss how to choose the first principal component $\vec{w}_1 \in \mathbb{R}^d$. To preserve the structure of the underlying data as much as possible, we want the vectors $\vec{x}_i$ projected onto the span of $\vec{w}_1$ to be as close as possible to the original vectors $\vec{x}_i$. We also want $\|\vec{w}_1\|_2 = 1$. Thus, the error of the projection across all data points is

$$\text{err}(\vec{w}_1) = \frac{1}{n} \sum_{i=1}^n \left\| \vec{x}_i - \vec{w}_1(\vec{w}_1^\top \vec{x}_i) \right\|_2^2. \tag{2.88}$$

Expanding, we have

$$\text{err}(\vec{w}_1) = \frac{1}{n} \sum_{i=1}^n \left\| \vec{x}_i - \vec{w}_1(\vec{w}_1^\top \vec{x}_i) \right\|_2^2 \tag{2.89}$$

$$= \frac{1}{n} \sum_{i=1}^n (\|\vec{x}_i\|_2^2 - (\vec{x}_i^\top \vec{w}_1)^2). \tag{2.90}$$

Now solving the principal components optimization problem gives

$$\min_{\substack{\vec{w}_1 \in \mathbb{R}^d \\ \|\vec{w}_1\|_2 = 1}} \text{err}(\vec{w}_1) = \frac{1}{n} \sum_{i=1}^n \|\vec{x}_i\|_2^2 - \max_{\substack{\vec{w}_1 \in \mathbb{R}^d \\ \|\vec{w}_1\|_2 = 1}} \vec{w}_1^\top C \vec{w}_1 \tag{2.91}$$

$$= \frac{1}{n} \sum_{i=1}^n \|\vec{x}_i\|_2^2 - \lambda_{\max}\{C\} \tag{2.92}$$

with the $\vec{w}_1$ achieving this upper bound being the eigenvector $\vec{u}_{\max}$ corresponding to the eigenvalue $\lambda_{\max}\{C\}$. Thus, the first principal component is exactly an eigenvector corresponding to the largest eigenvalue of the covariance matrix $C = X^\top X/n$.

This computation is a special case of the singular value decomposition, which is used in practice to compute the PCA of a dataset; understanding this decomposition will allow us to neatly compute the other principal components (i.e., second, third, fourth,...), as well.

2.6 Singular Value Decomposition

Definition 2.28 (SVD)**.** Let $A \in \mathbb{R}^{m\times n}$ have rank r. A *singular value decomposition* (SVD) of A is a decomposition of the form

$$A = U\Sigma V^\top \tag{2.93}$$

$$= \begin{bmatrix} U_r & U_{m-r} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix} \begin{bmatrix} V_r^\top \\ V_{n-r}^\top \end{bmatrix} \tag{2.94}$$

$$= U_r \Sigma_r V_r^\top \tag{2.95}$$

$$= \sum_{i=1}^{r} \sigma_i \vec{u}_i \vec{v}_i^\top, \tag{2.96}$$

where:

- $U \in \mathbb{R}^{m\times m}$, $U_r \in \mathbb{R}^{m\times r}$, $U_{m-r} \in \mathbb{R}^{m\times(m-r)}$, $V \in \mathbb{R}^{n\times n}$, $V_r \in \mathbb{R}^{n\times r}$, and $V_{n-r} \in \mathbb{R}^{n\times(n-r)}$ are orthonormal matrices, where $U = \begin{bmatrix} U_r & U_{m-r} \end{bmatrix}$ has columns $\vec{u}_1, \ldots, \vec{u}_m$ (*left singular vectors*) and $V = \begin{bmatrix} V_r & V_{n-r} \end{bmatrix}$ has columns $\vec{v}_1, \ldots, \vec{v}_n$ (*right singular vectors*).
- $\Sigma_r = \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}$ is a diagonal matrix with ordered positive entries $\sigma_1 \geq \cdots \geq \sigma_r > 0$ (*singular values*), and the zero matrices in Σ are shaped to ensure that $\Sigma \in \mathbb{R}^{m\times n}$.

Suppose that A is tall (so $m > n$) with full column rank n. Then the SVD looks like:

$$A = U \begin{bmatrix} \Sigma_n \\ 0_{(m-n)\times n} \end{bmatrix} V^\top. \tag{2.97}$$

On the other hand, if A is wide (so $m < n$) with full row rank m, then the SVD looks like:

$$A = U \begin{bmatrix} \Sigma_m & 0_{m\times(n-m)} \end{bmatrix} V^\top. \tag{2.98}$$

The last (summation) form of the SVD is called the *dyadic SVD*; this is because terms of the form $\vec{p}\vec{q}^\top$ are called dyads, and the dyadic SVD expresses the matrix A as the sum of dyads.

All forms of the SVD are useful conceptually and computationally, depending on the problem we are working on.

We now discuss a method to construct the SVD. Suppose $A \in \mathbb{R}^{m\times n}$ has rank r. We consider the symmetric matrix $A^\top A$ which has rank r and thus r nonzero eigenvalues, which are positive. We can order its eigenvalues as $\lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_n = 0$, say with corresponding orthonormal eigenvectors $\vec{v}_1, \ldots, \vec{v}_n$.

Then, for $i \in \{1, \ldots, r\}$, we define $\sigma_i \triangleq \sqrt{\lambda_i} > 0$, and $\vec{u}_i \triangleq A\vec{v}_i/\sigma_i$. This only gives us r vectors $\vec{u}_i$, but we need m of them to construct $U \in \mathbb{R}^{m\times m}$. To find the remaining $\vec{u}_i$ we use Gram-Schmidt on the matrix $\begin{bmatrix} \vec{u}_1 & \cdots & \vec{u}_r & I \end{bmatrix} \in \mathbb{R}^{m\times(r+m)}$, throwing out the r vectors whose projection residual onto previously processed vectors is 0.

Theorem 2.29. *In the context of the SVD construction algorithm, $\{\vec{u}_1, \ldots, \vec{u}_m\}$ is an orthonormal set.*

Theorem 2.30. *In the context of the SVD construction algorithm, we have $A = U\Sigma V^\top$.*

The SVD is not unique: the Gram-Schmidt process could have used any basis for $\mathbb{R}^m$ that wasn't the columns of I and still have been valid; if you had multiple eigenvectors of $A^\top A$ with the same eigenvalue then the choice of eigenvectors in the diagonalization would not be unique; and even if you didn't have multiple eigenvectors with the same eigenvalue, the eigenvectors would only be determined up to a sign change $\vec{v} \mapsto -\vec{v}$ anyways.

We now discuss the geometry of the SVD, especially how each component of the SVD acts on vectors. The key insight is to interpret U as a rotation or reflection, Σ as a scaling, and $V^\top$ as another rotation or reflection.

Since $V^\top$ is an orthonormal matrix, it represents a rotation and/or reflection, and so it maps the unit circle to the unit circle. The diagonal matrix Σ will scale each coordinate, obtaining an ellipse. Finally, the orthonormal matrix U will map this axis-aligned ellipse to an ellipse which isn't necessarily axis-aligned.

To understand the impact of A on any general vector $\vec{x}$, we write it in the V basis: $\vec{x} = \alpha_1\vec{v}_1 + \alpha_2\vec{v}_2$, and use linearity to obtain $A\vec{x} = \alpha_1\sigma_1\vec{u}_1 + \alpha_2\sigma_2\vec{u}_2$.

This perspective also says that σ_1 is the maximum scaling of any vector obtained by multiplication by A, and σ_r is the minimum nonzero scaling. (If $r < n$, i.e., A is not full column rank, then there are some nonzero vectors in $\mathbb{R}^n$ which are sent to $\vec{0}$ by A, so the minimum scaling is 0.)

2.7 Low-Rank Approximation

Sometimes, in real datasets, matrices have billions or trillions of entries. Storing all of them would be prohibitively expensive, and we would need a way to compress them down into their most important parts. This turns out to be doable via the SVD, as we will see.

To formally talk about a compression algorithm that stores a compressed version of the data with minimal error, we need to talk about what kind of errors are appropriate to discuss in the context of matrices. This motivates thinking about matrix norms.

There are two ways to think about a matrix. The first way is as a block of numbers. This norm is called the *Frobenius norm*, and it corresponds to unrolling an $m \times n$ matrix into a length $m \cdot n$ vector and taking its ℓ_2-norm.

Definition 2.31 (Frobenius Norm)**.** For a matrix $A \in \mathbb{R}^{m\times n}$, its Frobenius norm is defined as

$$\|A\|_F \triangleq \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}. \tag{2.99}$$

Theorem 2.32. *For a matrix $A \in \mathbb{R}^{m\times n}$, we have $\|A\|_F^2 = tr(A^\top A)$.*

Theorem 2.33. *For a matrix $A \in \mathbb{R}^{m\times n}$ and orthonormal matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$, we have*

$$\|UAV\|_F = \|UA\|_F = \|AV\|_F = \|A\|_F. \tag{2.100}$$

Theorem 2.34. *For a matrix $A \in \mathbb{R}^{m\times n}$ with rank r and singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$, we have*

$$\|A\|_F^2 = \sum_{i=1}^{r} \sigma_i^2. \tag{2.101}$$

The second way to think about a matrix is as a linear transformation. A suitable notion of size in this case is the largest scaling factor of the matrix on any unit vector; this is called the *spectral norm* or the matrix ℓ_2-norm.

Definition 2.35 (Spectral Norm)**.** For a matrix $A \in \mathbb{R}^{m\times n}$, its spectral norm is defined by

$$\|A\|_2 \triangleq \max_{\substack{\vec{x}\in\mathbb{R}^n \\ \|\vec{x}\|_2=1}} \|A\vec{x}\|_2 . \tag{2.102}$$

Theorem 2.36. *For a matrix $A \in \mathbb{R}^{m\times n}$ with rank r and singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$, we have*

$$\|A\|_2 = \sigma_1. \tag{2.103}$$

To present our main theorems about how to approximate the matrix well under these norms, we define notation. Fix a matrix $A \in \mathbb{R}^{m\times n}$. For convenience, let $p \triangleq \min\{m, n\}$. Suppose that A has rank $r \leq p$, and that A has SVD $A = \sum_{i=1}^{p} \sigma_i \vec{u}_i \vec{v}_i^\top$ where $\sigma_1 \geq \cdots \geq \sigma_r$ and define $\sigma_{r+1} = \sigma_{r+2} = \cdots = 0$. Then, for $k \leq p$, we can define

$$A_k \triangleq \sum_{i=1}^{k} \sigma_i \vec{u}_i \vec{v}_i^\top . \tag{2.104}$$

Note that if $k \ll p$, then A_k can be stored much more efficiently than A. It turns out that A_k indeed well-approximates A in the sense of the two norms. The two results are collectively known as the *Eckart-Young* (sometimes Eckart-Young-Mirsky) theorem(s).

Theorem 2.37 (Eckart-Young Theorem for Spectral Norm)**.** *We have*

$$A_k \in \operatorname*{argmin}_{\substack{B\in\mathbb{R}^{m\times n} \\ \mathit{rank}(B)\leq k}} \|A - B\|_2 , \tag{2.105}$$

or, equivalently,

$$\|A - A_k\|_2 \leq \|A - B\|_2 , \quad \forall B \in \mathbb{R}^{m\times n} : \mathit{rank}(B) \leq k. \tag{2.106}$$

Theorem 2.38 (Eckart-Young Theorem for Frobenius Norm)**.** *We have*

$$A_k \in \operatorname*{argmin}_{\substack{B\in\mathbb{R}^{m\times n} \\ \mathit{rank}(B)\leq k}} \|A - B\|_F , \tag{2.107}$$

or, equivalently,

$$\|A - A_k\|_F^2 \leq \|A - B\|_F^2 , \quad \forall B \in \mathbb{R}^{m\times n} : \mathit{rank}(B) \leq k. \tag{2.108}$$

2.8 (OPTIONAL) Block Matrix Identities

In this section, we list many ways to manipulate block matrices. Since each fact in here is something you can derive yourself using definitions, you may use any of them without proof.

2.8.1 Transposes of Block Matrices

$$\begin{bmatrix}\vec{x}_1 & \cdots & \vec{x}_n\end{bmatrix}^\top = \begin{bmatrix}\vec{x}_1^\top \\ \vdots \\ \vec{x}_n^\top\end{bmatrix} \tag{2.109}$$

$$\begin{bmatrix}A & B\end{bmatrix}^\top = \begin{bmatrix}A^\top \\ B^\top\end{bmatrix} \tag{2.110}$$

$$\begin{bmatrix}A & B \\ C & D\end{bmatrix}^\top = \begin{bmatrix}A^\top & C^\top \\ B^\top & D^\top\end{bmatrix} \tag{2.111}$$

2.8.2 Block Matrix Products

$$\begin{bmatrix}\vec{x}_1 & \cdots & \vec{x}_n\end{bmatrix}\begin{bmatrix}\vec{y}_1^\top \\ \vdots \\ \vec{y}_n^\top\end{bmatrix} = \sum_{i=1}^{n}\vec{x}_i\vec{y}_i^\top \tag{2.112}$$

$$A\begin{bmatrix}\vec{x}_1 & \cdots & \vec{x}_n\end{bmatrix} = \begin{bmatrix}A\vec{x}_1 & \cdots & A\vec{x}_n\end{bmatrix} \tag{2.113}$$

$$A\begin{bmatrix}B & C\end{bmatrix} = \begin{bmatrix}AB & AC\end{bmatrix} \tag{2.114}$$

$$\begin{bmatrix}A \\ B\end{bmatrix}C = \begin{bmatrix}AC \\ BC\end{bmatrix} \tag{2.115}$$

2.8.3 Quadratic Forms

$$\vec{x}^\top A\vec{y} = \sum_i\sum_j A_{ij}x_iy_j \tag{2.116}$$

$$\begin{bmatrix}\vec{x} \\ \vec{y}\end{bmatrix}^\top\begin{bmatrix}A & B \\ C & D\end{bmatrix}\begin{bmatrix}\vec{x} \\ \vec{y}\end{bmatrix} = \vec{x}^\top A\vec{x} + \vec{x}^\top B\vec{y} + \vec{y}^\top C\vec{x} + \vec{y}^\top D\vec{y} \tag{2.117}$$

Chapter 3

Vector Calculus

3.1 Gradient, Jacobian, and Hessian

To motivate this section, we start with a familiar concept: the derivatives of a scalar function $f : \mathbb{R} \to \mathbb{R}$ which takes in scalar input and produces a scalar output. The derivative quantifies the (instantaneous) rate of change of the function due to the change of its input.

Definition 3.1 (Derivative for Scalar Functions)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be differentiable. The *derivative* of f with respect to x is the function $\frac{\mathrm{d}f}{\mathrm{d}x} : \mathbb{R} \to \mathbb{R}$ defined by

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \tag{3.1}$$

In this section, we aim to generalize the concept of derivatives beyond scalar functions. We will focus on two types of functions:

1. *Multivariate functions* $f : \mathbb{R}^n \to \mathbb{R}$ which take a vector $\vec{x} \in \mathbb{R}^n$ as input and produce a scalar $f(\vec{x}) \in \mathbb{R}$ as output.
2. *Vector-valued functions* $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$ which take a vector $\vec{x} \in \mathbb{R}^n$ as input and produce another vector $\vec{f}(\vec{x}) \in \mathbb{R}^m$ as output.

Theorem 3.2 (Chain Rule for Scalar Functions)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ be two differentiable scalar functions, and let $h : \mathbb{R} \to \mathbb{R}$ be defined as $h(x) = f(g(x))$ for all $x \in \mathbb{R}$. Then h is differentiable, and*

$$\frac{\mathrm{d}h}{\mathrm{d}x}(x) = \frac{\mathrm{d}f}{\mathrm{d}g}(g(x)) \cdot \frac{\mathrm{d}g}{\mathrm{d}x}(x). \tag{3.2}$$

3.1.1 Partial Derivatives

For multivariate functions $f : \mathbb{R}^n \to \mathbb{R}$, when we talk about the rate of change of the function with respect to its input, we need to specify which input we are talking about. Partial derivatives quantify this and give us the rate of change of the function due to the change of one of its inputs, say x_i, while keeping all other inputs fixed.

Definition 3.3 (Partial Derivative)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. The *partial derivative* of f with respect to x_i is the function $\frac{\partial f}{\partial x_i} : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\frac{\partial f}{\partial x_i}(\vec{x}) = \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots, x_n) - f(\vec{x})}{h}, \tag{3.3}$$

or equivalently,

$$\frac{\partial f}{\partial x_i}(\vec{x}) = \lim_{h \to 0} \frac{f(\vec{x} + h \cdot \vec{e}_i) - f(\vec{x})}{h} \tag{3.4}$$

where $\vec{e}_i$ is the ith standard basis vector.

Problem Solving Strategy. *To compute the partial derivative $\frac{\partial f}{\partial x_i}$, pretend that all x_j for $j \neq i$ are constants, then take the ordinary derivative in x_i.*

Theorem 3.4 (Chain Rule For Multivariate Functions)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $\vec{g} : \mathbb{R} \to \mathbb{R}^n$ be differentiable functions. Define the function $h : \mathbb{R} \to \mathbb{R}$ by $h(x) = f(\vec{g}(x))$ for all $x \in \mathbb{R}$. Then h is differentiable and has derivative*

$$\frac{\mathrm{d}h}{\mathrm{d}x}(x) = \sum_{i=1}^{n} \frac{\partial f}{\partial g_i}(\vec{g}(x)) \cdot \frac{\mathrm{d}g_i}{\mathrm{d}x}(x). \tag{3.5}$$

3.1.2 Gradient

We will now use the definition of partial derivatives to introduce the gradient of multivariate functions.

Definition 3.5 (Gradient)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. The *gradient* of f is the function $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\nabla f(\vec{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\vec{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\vec{x}) \end{bmatrix}. \tag{3.6}$$

Note that the gradient is a column vector. We will now list two important geometric properties of the gradient.

Theorem 3.6. *Let $\vec{x} \in \mathbb{R}^n$. The gradient $\nabla f(\vec{x})$ points in the direction of* steepest ascent *at $\vec{x}$, i.e., the direction around $\vec{x}$ in which f has the maximum rate of change. Furthermore, this rate of change is quantified by the norm $\|\nabla f(\vec{x})\|_2$.*

Definition 3.7 (Level Set)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, and $\alpha \in \mathbb{R}$ be a scalar.

- The *α-level set* of f is the set of points $\vec{x}$ such that $f(\vec{x}) = \alpha$:

$$L_\alpha(f) = \{\vec{x} \in \mathbb{R}^n \mid f(\vec{x}) = \alpha\}. \tag{3.7}$$

- The *α-sublevel set* of f is the set of points $\vec{x}$ such that $f(\vec{x}) \leq \alpha$:

$$L_{\leq \alpha}(f) = \{\vec{x} \in \mathbb{R}^n \mid f(\vec{x}) \leq \alpha\}. \tag{3.8}$$

- The *α-superlevel set* of f is the set of points $\vec{x}$ such that $f(\vec{x}) \geq \alpha$:

$$L_{\geq \alpha}(f) = \{\vec{x} \in \mathbb{R}^n \mid f(\vec{x}) \geq \alpha\}. \tag{3.9}$$

Theorem 3.8. *Let $\vec{x} \in \mathbb{R}^n$ and suppose $f(\vec{x}) = \alpha$. Then $\nabla f(\vec{x})$ is orthogonal to the hyperplane which is tangent at $\vec{x}$ to the α-level set of f.*

Example 3.9 (Gradient of the Squared ℓ_2 Norm)**.** Consider the function $f(\vec{x}) = \|\vec{x}\|_2^2$ where $\vec{x} \in \mathbb{R}^2$. We have

$$\nabla f(\vec{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = 2\vec{x}. \tag{3.10}$$

Example 3.10 (Gradient of Linear Function)**.** For the linear function $f(\vec{x}) = \vec{a}^\top \vec{x}$ where $\vec{a} \in \mathbb{R}^n$ is fixed, we have

$$\nabla f(\vec{x}) = \vec{a}. \tag{3.11}$$

Example 3.11 (Gradient of the Quadratic Form)**.** Let $A \in \mathbb{R}^{n \times n}$. For the quadratic function $f(\vec{x}) = \vec{x}^\top A\vec{x}$, we have

$$\nabla f(\vec{x}) = (A + A^\top)\vec{x}. \tag{3.12}$$

If A is symmetric, then $\nabla f(\vec{x}) = 2A\vec{x}$.

3.1.3 Jacobian

We now have the tools to generalize the notion of derivatives to vector-valued functions $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$.

Definition 3.12 (Jacobian)**.** Let $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable function. The *Jacobian* of $\vec{f}$ is the function $D\vec{f} : \mathbb{R}^n \to \mathbb{R}^{m \times n}$ defined as

$$D\vec{f}(\vec{x}) = \begin{bmatrix} \nabla f_1(\vec{x})^\top \\ \vdots \\ \nabla f_m(\vec{x})^\top \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\vec{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\vec{x}) & \cdots & \frac{\partial f_m}{\partial x_n}(\vec{x}) \end{bmatrix}. \tag{3.13}$$

One big thing to note is that the Jacobian is different from the gradient! If $f : \mathbb{R}^n \to \mathbb{R}^1 = \mathbb{R}$, then its Jacobian $Df : \mathbb{R}^n \to \mathbb{R}^{1 \times n}$ is a function which outputs a row vector. This row vector is the transpose of the gradient.

Theorem 3.13 (Chain Rule for Vector-Valued Functions)**.** *Let $\vec{f} : \mathbb{R}^p \to \mathbb{R}^m$ and $\vec{g} : \mathbb{R}^n \to \mathbb{R}^p$ be differentiable functions. Let $\vec{h} : \mathbb{R}^n \to \mathbb{R}^m$ be defined as $\vec{h}(\vec{x}) = \vec{f}(\vec{g}(\vec{x}))$ for all $\vec{x} \in \mathbb{R}^n$. Then $\vec{h}$ is differentiable, and*

$$D\vec{h}(\vec{x}) = [D\vec{f}(\vec{g}(\vec{x}))][D\vec{g}(\vec{x})]. \tag{3.14}$$

Corollary 3.14. *Let $f : \mathbb{R}^p \to \mathbb{R}$ and $\vec{g} : \mathbb{R}^n \to \mathbb{R}^p$ be differentiable functions. Let $h : \mathbb{R}^n \to \mathbb{R}$ be defined as $h(\vec{x}) = f(\vec{g}(\vec{x}))$ for all $\vec{x} \in \mathbb{R}^n$. Then h is differentiable, and*

$$\nabla h(\vec{x}) = [D\vec{g}(\vec{x})]^\top \nabla f(\vec{g}(\vec{x})). \tag{3.15}$$

Example 3.15. Using the chain rule to compute the gradient of $h(\vec{x}) = \|A\vec{x} - \vec{y}\|_2^2$:

It can be written as $h(\vec{x}) = f(\vec{g}(\vec{x}))$ where $f(\vec{x}) = \|\vec{x}\|_2^2$ and $\vec{g}(\vec{x}) = A\vec{x} - \vec{y}$. We have $D\vec{g}(\vec{x}) = A$ and $\nabla f(\vec{x}) = 2\vec{x}$. Thus

$$\nabla h(\vec{x}) = [D\vec{g}(\vec{x})]^\top \nabla f(\vec{g}(\vec{x})) = 2A^\top(A\vec{x} - \vec{y}). \tag{3.16}$$

3.1.4 Hessian

For multivariate functions $f : \mathbb{R}^n \to \mathbb{R}$, defining a second derivative as a particular matrix becomes possible; this matrix, called the Hessian, has great conceptual and computational importance.

The Hessian is exactly the Jacobian of the gradient.

Definition 3.16 (Hessian)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable. The *Hessian* of f is the function $\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ defined by

$$\nabla^2 f(\vec{x}) = D(\nabla f)(\vec{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\vec{x}) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(\vec{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(\vec{x}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\vec{x}) \end{bmatrix}. \tag{3.17}$$

Theorem 3.17 (Clairaut's Theorem)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, and fix $\vec{x} \in \mathbb{R}^n$. Then $\nabla^2 f(\vec{x})$ is a symmetric matrix, i.e., for every $1 \le i, j \le n$ we have*

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(\vec{x}) = \frac{\partial^2 f}{\partial x_j \partial x_i}(\vec{x}). \tag{3.18}$$

Example 3.18 (Hessian of Squared ℓ_2 Norm)**.** For the function $f(\vec{x}) = \|\vec{x}\|_2^2$ where $\vec{x} \in \mathbb{R}^2$, the gradient is $\nabla f(\vec{x}) = 2\vec{x}$. The Hessian is

$$\nabla^2 f(\vec{x}) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = 2I. \tag{3.19}$$

Example 3.19 (Hessian of $\log(1+\|\vec{x}\|_2^2)$)**.** Consider the function $h(\vec{x}) = \log(1+\|\vec{x}\|_2^2)$. The gradient is

$$\nabla h(\vec{x}) = \frac{2\vec{x}}{1+\|\vec{x}\|_2^2}. \tag{3.20}$$

For the Hessian, computing componentwise:

$$[\nabla^2 h(\vec{x})]_{j,k} = \frac{\partial}{\partial x_k}\left(\frac{2x_j}{1+\|\vec{x}\|_2^2}\right) = -\frac{4x_j x_k}{(1+\|\vec{x}\|_2^2)^2} + \frac{2}{1+\|\vec{x}\|_2^2}\cdot \mathbf{1}_{j=k}. \tag{3.21}$$

In matrix form:

$$\nabla^2 h(\vec{x}) = \frac{2}{1+\|\vec{x}\|_2^2} I - \frac{4\vec{x}\vec{x}^\top}{(1+\|\vec{x}\|_2^2)^2}. \tag{3.22}$$

3.2 Taylor's Theorems

Taylor approximation is a tool to find polynomial approximations of functions using information about the function value at a point along with the value of its first, second, and higher order derivatives.

Definition 3.20 (Taylor Approximation)**.** Let $f : \mathbb{R} \to \mathbb{R}$ be a k-times continuously differentiable function, and fix $x_0 \in \mathbb{R}$. The kth degree Taylor approximation around x_0 is the function $\hat{f}_k(\cdot; x_0) : \mathbb{R} \to \mathbb{R}$ given by

$$\hat{f}_k(x; x_0) = f(x_0) + \frac{1}{1!}\frac{\mathrm{d}f}{\mathrm{d}x}(x_0)(x-x_0) + \cdots + \frac{1}{k!}\frac{\mathrm{d}^k f}{\mathrm{d}x^k}(x_0)(x-x_0)^k = \sum_{i=0}^{k} \frac{1}{i!}\frac{\mathrm{d}^i f}{\mathrm{d}x^i}(x_0)(x-x_0)^i. \tag{3.23}$$

In particular, the first-order and second-order Taylor approximations are

$$\hat{f}_1(x; x_0) = f(x_0) + \frac{\mathrm{d}f}{\mathrm{d}x}(x_0)(x - x_0) \tag{3.24}$$

$$\hat{f}_2(x; x_0) = f(x_0) + \frac{\mathrm{d}f}{\mathrm{d}x}(x_0)(x - x_0) + \frac{1}{2}\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(x_0)(x - x_0)^2. \tag{3.25}$$

Example 3.21 (Taylor Approximation of Cubic Function)**.** Let us approximate $f(x) = x^3$ around $x_0 = 1$:

$$\hat{f}_1(x; 1) = 1 + 3(x - 1) = 3x - 2 \tag{3.26}$$

$$\hat{f}_2(x; 1) = 3x - 2 + 3(x - 1)^2 = 3x^2 - 3x + 1 \tag{3.27}$$

$$\hat{f}_3(x; 1) = 3x^2 - 3x + 1 + (x - 1)^3 = x^3. \tag{3.28}$$

The first-order approximation is the tangent line, the second-order captures local curvature, and the third-degree exactly equals f since f is cubic.

Theorem 3.22 (Taylor's Theorem)**.** *Let $f : \mathbb{R} \to \mathbb{R}$ be k-times continuously differentiable, and fix $x_0 \in \mathbb{R}$. Then for all $x \in \mathbb{R}$:*

$$f(x) = \hat{f}_k(x; x_0) + o(|x - x_0|^k) \tag{3.29}$$

where $o(|x - x_0|^k)$ denotes a remainder $R_k(x; x_0)$ such that $\lim_{x \to x_0} \frac{R_k(x;x_0)}{|x-x_0|^k} = 0$.

A more useful form:

$$f(x + \delta) = f(x) + \frac{\mathrm{d}f}{\mathrm{d}x}(x) \cdot \delta + o(|\delta|) \tag{3.30}$$

$$= f(x) + \frac{\mathrm{d}f}{\mathrm{d}x}(x) \cdot \delta + \frac{1}{2}\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}(x) \cdot \delta^2 + o(\delta^2). \tag{3.31}$$

3.2.1 Taylor Approximation of Multivariate Functions

Definition 3.23 (Multivariate Taylor Approximations)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ and fix $\vec{x}_0 \in \mathbb{R}^n$.

- If f is continuously differentiable, its first-order Taylor approximation around $\vec{x}_0$ is

$$\hat{f}_1(\vec{x}; \vec{x}_0) = f(\vec{x}_0) + [\nabla f(\vec{x}_0)]^\top (\vec{x} - \vec{x}_0). \tag{3.32}$$

- If f is twice continuously differentiable, its second-order Taylor approximation is

$$\hat{f}_2(\vec{x}; \vec{x}_0) = f(\vec{x}_0) + [\nabla f(\vec{x}_0)]^\top (\vec{x} - \vec{x}_0) + \frac{1}{2}(\vec{x} - \vec{x}_0)^\top [\nabla^2 f(\vec{x}_0)](\vec{x} - \vec{x}_0). \tag{3.33}$$

The graph of the first-order Taylor approximation is the hyperplane tangent to the graph of f at $(\vec{x}_0, f(\vec{x}_0))$.

Theorem 3.24 (Taylor's Theorem (Multivariate))**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be k-times continuously differentiable, and fix $\vec{x}_0 \in \mathbb{R}^n$. Then for all $\vec{x} \in \mathbb{R}^n$:*

$$f(\vec{x}) = \hat{f}_k(\vec{x}; \vec{x}_0) + o(\|\vec{x} - \vec{x}_0\|_2^k). \tag{3.34}$$

In more workable form for $k = 1, 2$:

$$f(\vec{x} + \vec{\delta}) = f(\vec{x}) + [\nabla f(\vec{x})]^\top \vec{\delta} + o(\left\|\vec{\delta}\right\|_2) \tag{3.35}$$

$$= f(\vec{x}) + [\nabla f(\vec{x})]^\top \vec{\delta} + \frac{1}{2}\vec{\delta}^\top [\nabla^2 f(\vec{x})]\vec{\delta} + o(\left\|\vec{\delta}\right\|_2^2). \tag{3.36}$$

Example 3.25 (Taylor Approximation of Squared ℓ_2 Norm)**.** For $f(\vec{x}) = \|\vec{x}\|_2^2$ with $\nabla f(\vec{x}) = 2\vec{x}$ and $\nabla^2 f(\vec{x}) = 2I$:

$$\hat{f}_1(\vec{x}; \vec{x}_0) = \|\vec{x}_0\|_2^2 + 2\vec{x}_0^\top(\vec{x} - \vec{x}_0) = 2\vec{x}_0^\top \vec{x} - \|\vec{x}_0\|_2^2 \tag{3.37}$$

$$\hat{f}_2(\vec{x}; \vec{x}_0) = \hat{f}_1(\vec{x}; \vec{x}_0) + \frac{1}{2}(\vec{x} - \vec{x}_0)^\top [2I](\vec{x} - \vec{x}_0) = \|\vec{x}\|_2^2 . \tag{3.38}$$

Thus $\hat{f}_2 = f$ independently of $\vec{x}_0$, since f is quadratic.

Example 3.26 (Computing Gradients via Pattern Matching)**.** For $f(\vec{x}) = \vec{x}^\top A\vec{x}$, we perturb and expand:

$$f(\vec{x} + \vec{\delta}) = (\vec{x} + \vec{\delta})^\top A(\vec{x} + \vec{\delta}) \tag{3.39}$$

$$= \vec{x}^\top A\vec{x} + \vec{\delta}^\top A\vec{x} + \vec{x}^\top A\vec{\delta} + \vec{\delta}^\top A\vec{\delta} \tag{3.40}$$

$$= f(\vec{x}) + ((A + A^\top)\vec{x})^\top \vec{\delta} + \frac{1}{2}\vec{\delta}^\top (A + A^\top)\vec{\delta}. \tag{3.41}$$

By pattern matching with Taylor's theorem: $\nabla f(\vec{x}) = (A + A^\top)\vec{x}$ and $\nabla^2 f(\vec{x}) = A + A^\top$.

Definition 3.27 (Vector-Valued Taylor Approximation)**.** Let $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$ and fix $\vec{x}_0 \in \mathbb{R}^n$. If $\vec{f}$ is continuously differentiable, its first-order Taylor approximation around $\vec{x}_0$ is

$$\hat{\vec{f}}_1(\vec{x}; \vec{x}_0) = \vec{f}(\vec{x}_0) + [D\vec{f}(\vec{x}_0)](\vec{x} - \vec{x}_0). \tag{3.42}$$

Theorem 3.28 (Vector-Valued Taylor's Theorem)**.** *Let $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$ be continuously differentiable, and fix $\vec{x}_0 \in \mathbb{R}^n$. Then for all $\vec{x} \in \mathbb{R}^n$:*

$$\vec{f}(\vec{x}) = \hat{\vec{f}}_1(\vec{x}; \vec{x}_0) + \vec{o}(\|\vec{x} - \vec{x}_0\|_2). \tag{3.43}$$

In workable form: $\vec{f}(\vec{x} + \vec{\delta}) = \vec{f}(\vec{x}) + [D\vec{f}(\vec{x})]\vec{\delta} + o(\left\|\vec{\delta}\right\|_2)$.

3.3 The Main Theorem

Theorem 3.29 (The Main Theorem)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, and let $\Omega \subseteq \mathbb{R}^n$ be an open set. Consider the optimization problem*

$$\min_{\vec{x} \in \Omega} f(\vec{x}). \tag{3.44}$$

Let $\vec{x}^\star$ be a solution to this problem. Then $\nabla f(\vec{x}^\star) = \vec{0}$.

This gives a *necessary* condition for optimality: any optimal point must have gradient equal to zero.

Proof. We prove for scalar functions $f : \mathbb{R} \to \mathbb{R}$. Using Taylor approximation around $x^\star$:

$$f(x) = f(x^\star) + \frac{\mathrm{d}f}{\mathrm{d}x}(x^\star)(x - x^\star) + o(|x - x^\star|). \tag{3.45}$$

Since $f(x^\star) \leq f(x)$ for all $x \in \Omega$:

$$0 \leq \frac{\mathrm{d}f}{\mathrm{d}x}(x^\star)(x - x^\star) + o(|x - x^\star|). \tag{3.46}$$

Since Ω is open, there exists $r > 0$ such that $B_r(x^\star) \subseteq \Omega$. Partitioning into B_+ (where $x - x^\star \geq 0$) and B_- (where $x - x^\star < 0$):

For $x \in B_+$, taking $x \to x^\star$ gives $0 \leq \frac{\mathrm{d}f}{\mathrm{d}x}(x^\star)$.

For $x \in B_-$, taking $x \to x^\star$ gives $0 \geq \frac{\mathrm{d}f}{\mathrm{d}x}(x^\star)$.

Thus $\frac{\mathrm{d}f}{\mathrm{d}x}(x^\star) = 0$. □

3.4 Directional Derivatives

Definition 3.30 (Directional Derivative)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable, and fix $\vec{u} \in \mathbb{R}^n$ with $\|\vec{u}\|_2 = 1$. The *directional derivative* of f along $\vec{u}$ is

$$Df(\vec{x})[\vec{u}] = \lim_{h \to 0} \frac{f(\vec{x} + h \cdot \vec{u}) - f(\vec{x})}{h}. \tag{3.47}$$

Proposition 3.31. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable, and fix $\vec{u} \in \mathbb{R}^n$ with $\|\vec{u}\|_2 = 1$. Then*

$$Df(\vec{x})[\vec{u}] = \vec{u}^\top [\nabla f(\vec{x})]. \tag{3.48}$$

In particular, $Df(\vec{x})[\vec{e}_i] = \frac{\partial f}{\partial x_i}(\vec{x})$.

3.5 (OPTIONAL) Matrix Calculus

We generalize derivatives to functions $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ that take a matrix X as input and produce a scalar.

Definition 3.32 (Gradient for Matrix Functions)**.** Let $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ be differentiable. The *gradient* of f is $\nabla f : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ defined as

$$\nabla f(X) = \begin{bmatrix} \frac{\partial f}{\partial X_{11}}(X) & \cdots & \frac{\partial f}{\partial X_{1n}}(X) \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial X_{m1}}(X) & \cdots & \frac{\partial f}{\partial X_{mn}}(X) \end{bmatrix}. \tag{3.49}$$

Theorem 3.33 (Chain Rule for Matrix Functions)**.** *Let $F : \mathbb{R}^{p \times q} \to \mathbb{R}^{r \times s}$ and $G : \mathbb{R}^{m \times n} \to \mathbb{R}^{p \times q}$ be differentiable. Let $H(X) = F(G(X))$. Then*

$$\frac{\partial H_{ij}}{\partial X_{k\ell}}(X) = \sum_a \sum_b \frac{\partial F_{ij}}{\partial G_{ab}}(G(X)) \frac{\partial G_{ab}}{\partial X_{k\ell}}(X). \tag{3.50}$$

Definition 3.34 (Matrix Taylor Approximation)**.** Let $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ and fix $X_0 \in \mathbb{R}^{m \times n}$. If f is continuously differentiable, its first-order Taylor approximation is

$$\hat{f}_1(X; X_0) = f(X_0) + \operatorname{tr}\left([\nabla f(X_0)]^\top (X - X_0) \right). \tag{3.51}$$

Chapter 4

Linear and Ridge Regression

4.1 Impact of Perturbations on Linear Regression

Let $A \in \mathbb{R}^{n\times n}$ be invertible and $\vec{y} \in \mathbb{R}^n$. Consider the linear system $A\vec{x} = \vec{y}$ with unique solution $\vec{x} = A^{-1}\vec{y}$. We want to understand how sensitive this system is to perturbations in the output.

If $\vec{y}$ is perturbed by $\vec{\delta}_{\vec{y}}$, then $\vec{x}$ is also perturbed by $\vec{\delta}_{\vec{x}}$:

$$A(\vec{x} + \vec{\delta}_{\vec{x}}) = \vec{y} + \vec{\delta}_{\vec{y}}. \tag{4.1}$$

From $A\vec{\delta}_{\vec{x}} = \vec{\delta}_{\vec{y}}$, we get $\vec{\delta}_{\vec{x}} = A^{-1}\vec{\delta}_{\vec{y}}$, and thus:

$$\left\|\vec{\delta}_{\vec{x}}\right\|_2 = \left\|A^{-1}\vec{\delta}_{\vec{y}}\right\|_2 \leq \left\|A^{-1}\right\|_2 \left\|\vec{\delta}_{\vec{y}}\right\|_2. \tag{4.2}$$

Also, from $A\vec{x} = \vec{y}$: $\|\vec{x}\|_2 \geq \frac{\|\vec{y}\|_2}{\|A\|_2}$.

Combining:

$$\frac{\left\|\vec{\delta}_{\vec{x}}\right\|_2}{\|\vec{x}\|_2} \leq \|A\|_2 \left\|A^{-1}\right\|_2 \cdot \frac{\left\|\vec{\delta}_{\vec{y}}\right\|_2}{\|\vec{y}\|_2}. \tag{4.3}$$

Definition 4.1 (Condition Number)**.** Let $A \in \mathbb{R}^{n\times n}$. The *condition number* of A is

$$\kappa(A) \triangleq \frac{\sigma_1\{A\}}{\sigma_n\{A\}}. \tag{4.4}$$

If $\kappa(A)$ is large, small changes in $\vec{y}$ cause huge changes in $\vec{x}$. If $\kappa(A)$ is small, the system is robust.

For least-squares systems, we use the normal equations $A^\top A\vec{x} = A^\top\vec{y}$, with condition number:

$$\kappa(A^\top A) = \frac{\lambda_{\max}\{A^\top A\}}{\lambda_{\min}\{A^\top A\}}. \tag{4.5}$$

4.2 Ridge Regression

When $\kappa(A^\top A)$ is infinite or very large, we can improve conditioning by adding λI to $A^\top A$.

Theorem 4.2 (Ridge Regression)**.** *Let $A \in \mathbb{R}^{m\times n}$, $\vec{y} \in \mathbb{R}^m$, and $\lambda > 0$. The unique solution to the ridge regression problem*

$$\min_{\vec{x}\in\mathbb{R}^n} \left\{ \|A\vec{x} - \vec{y}\|_2^2 + \lambda \|\vec{x}\|_2^2 \right\} \tag{4.6}$$

is given by

$$\vec{x}^\star = (A^\top A + \lambda I)^{-1} A^\top \vec{y}. \tag{4.7}$$

Proof. Let $f(\vec{x}) = \|A\vec{x} - \vec{y}\|_2^2 + \lambda \|\vec{x}\|_2^2$. Taking the gradient:

$$\nabla_{\vec{x}} f(\vec{x}) = 2A^\top A\vec{x} - 2A^\top \vec{y} + 2\lambda\vec{x} = 2(A^\top A + \lambda I)\vec{x} - 2A^\top \vec{y}. \tag{4.8}$$

Setting to zero: $(A^\top A + \lambda I)\vec{x} = A^\top \vec{y}$. Since $A^\top A + \lambda I$ is PD (thus invertible), $\vec{x}^\star = (A^\top A + \lambda I)^{-1} A^\top \vec{y}$. □

An alternative derivation uses the augmented system:

$$\begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix} \vec{x} = \begin{bmatrix} \vec{y} \\ \vec{0} \end{bmatrix}. \tag{4.9}$$

The term $\lambda \|\vec{x}\|_2^2$ is called a *regularizer*—it regularizes the problem by making it better-conditioned.

4.3 Principal Components Regression

Using the SVD $A = U\Sigma V^\top$, the ridge regression solution becomes:

$$\vec{x}^\star = (A^\top A + \lambda I)^{-1} A^\top \vec{y} \tag{4.10}$$

$$= V \begin{bmatrix} (\Sigma_r^2 + \lambda I)^{-1}\Sigma_r & 0 \\ 0 & 0 \end{bmatrix} U^\top \vec{y} \tag{4.11}$$

$$= \sum_{i=1}^{r} \frac{\sigma_i\{A\}}{\sigma_i\{A\}^2 + \lambda} (\vec{u}_i^\top \vec{y}) \cdot \vec{v}_i. \tag{4.12}$$

For large λ, this performs "soft thresholding" of singular values: terms with smaller singular values are nearly zeroed out while terms with larger singular values are preserved. Thus ridge regression behaves qualitatively similar to a soft form of PCA.

4.4 Tikhonov Regression

Tikhonov regression generalizes ridge regression by allowing different weights and a prior $\vec{x}_0$:

Theorem 4.3 (Tikhonov Regression)**.** *Let $A \in \mathbb{R}^{m\times n}$, $\vec{x}_0 \in \mathbb{R}^n$, $\vec{y} \in \mathbb{R}^m$, and let $W_1 \in \mathbb{R}^{m\times m}$, $W_2 \in \mathbb{R}^{n\times n}$ be diagonal. The unique solution to*

$$\min_{\vec{x}\in\mathbb{R}^n} \left\{ \|W_1(A\vec{x} - \vec{y})\|_2^2 + \|W_2(\vec{x} - \vec{x}_0)\|_2^2 \right\} \tag{4.13}$$

is given by

$$\vec{x}^\star = (A^\top W_1^2 A + W_2^2)^{-1}(A^\top W_1^2 \vec{y} + W_2^2 \vec{x}_0). \tag{4.14}$$

Setting $W_1 = I$, $W_2 = \sqrt{\lambda} I$, and $\vec{x}_0 = \vec{0}$ recovers ridge regression.

4.5 Maximum Likelihood Estimation (MLE)

Suppose we have the probabilistic model $y_i = \vec{a}_i^\top \vec{x} + w_i$ where $w_i \sim \mathcal{N}(0, \sigma_i^2)$ are independent. In short: $\vec{y} = A\vec{x} + \vec{w}$ where $\vec{w} \sim \mathcal{N}(\vec{0}, \Sigma_{\vec{w}})$ with $\Sigma_{\vec{w}} = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_m^2)$.

Proposition 4.4 (MLE as Tikhonov Regression)**.** *In the above model:*

$$\operatorname*{argmax}_{\vec{x}\in\mathbb{R}^n} p_{\vec{x}}(\vec{y}) = \operatorname*{argmin}_{\vec{x}\in\mathbb{R}^n} \left\| \Sigma_{\vec{w}}^{-1/2}(A\vec{x} - \vec{y}) \right\|_2^2. \tag{4.15}$$

Proof. Since log is monotonically increasing:

$$\operatorname*{argmax}_{\vec{x}} p_{\vec{x}}(\vec{y}) = \operatorname*{argmax}_{\vec{x}} \sum_{i=1}^m \log\left(\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - \vec{a}_i^\top \vec{x})^2}{2\sigma_i^2} \right) \right) \tag{4.16}$$

$$= \operatorname*{argmin}_{\vec{x}} \sum_{i=1}^m \frac{(y_i - \vec{a}_i^\top \vec{x})^2}{\sigma_i^2} = \operatorname*{argmin}_{\vec{x}} \left\| \Sigma_{\vec{w}}^{-1/2}(A\vec{x} - \vec{y}) \right\|_2^2. \tag{4.17}$$

□

4.6 Maximum A Posteriori Estimation (MAP)

Now suppose $\vec{x}$ is also random: $x_j = \mu_j + v_j$ where $v_j \sim \mathcal{N}(0, \tau_j^2)$. In short: $\vec{x} = \vec{x}_0 + \vec{v}$ where $\vec{v} \sim \mathcal{N}(\vec{0}, \Sigma_{\vec{v}})$ with $\Sigma_{\vec{v}} = \operatorname{diag}(\tau_1^2, \ldots, \tau_n^2)$.

Theorem 4.5 (MAP as Tikhonov Regression)**.** *In the above model:*

$$\operatorname*{argmax}_{\vec{x}\in\mathbb{R}^n} p(\vec{x}|\vec{y}) = \operatorname*{argmin}_{\vec{x}\in\mathbb{R}^n} \left\{ \left\| \Sigma_{\vec{w}}^{-1/2}(A\vec{x} - \vec{y}) \right\|_2^2 + \left\| \Sigma_{\vec{v}}^{-1/2}(\vec{x} - \vec{x}_0) \right\|_2^2 \right\}. \tag{4.18}$$

Proof. Using Bayes' rule $p(\vec{x}|\vec{y}) = \frac{p(\vec{y}|\vec{x})p(\vec{x})}{p(\vec{y})}$ and taking log:

$$\operatorname*{argmax}_{\vec{x}} p(\vec{x}|\vec{y}) = \operatorname*{argmax}_{\vec{x}} \{\log p(\vec{y}|\vec{x}) + \log p(\vec{x})\} \tag{4.19}$$

$$= \operatorname*{argmax}_{\vec{x}} \left\{ \sum_{i=1}^m \left(-\frac{(y_i - \vec{a}_i^\top \vec{x})^2}{2\sigma_i^2} \right) + \sum_{j=1}^n \left(-\frac{(x_j - (\vec{x}_0)_j)^2}{2\tau_j^2} \right) \right\} \tag{4.20}$$

$$= \operatorname*{argmin}_{\vec{x}} \left\{ \left\| \Sigma_{\vec{w}}^{-1/2}(A\vec{x} - \vec{y}) \right\|_2^2 + \left\| \Sigma_{\vec{v}}^{-1/2}(\vec{x} - \vec{x}_0) \right\|_2^2 \right\}. \tag{4.21}$$

□

Chapter 5

Convexity

5.1 Convex Sets

5.1.1 Basics

Definition 5.1 (Convex Combination)**.** Let $\vec{x}_1, \ldots, \vec{x}_k \in \mathbb{R}^n$. The sum

$$\vec{x} = \sum_{i=1}^{k} \theta_i \vec{x}_i \tag{5.1}$$

is a *convex combination* of $\vec{x}_1, \ldots, \vec{x}_k$ if each $\theta_i \geq 0$ and $\sum_{i=1}^{k} \theta_i = 1$.

We can interpret each θ_i as a weight or probability.

Definition 5.2 (Convex Set)**.** Let $C \subseteq \mathbb{R}^n$. We say C is a *convex set* if it is closed under convex combinations: for all $\vec{x}_1, \vec{x}_2 \in C$ and all $\theta \in [0, 1]$, we have $\theta\vec{x}_1 + (1-\theta)\vec{x}_2 \in C$.

Geometrically, C is convex if for every two points $\vec{x}_1, \vec{x}_2 \in C$, the line segment $\{\theta\vec{x}_1 + (1-\theta)\vec{x}_2 \mid \theta \in [0, 1]\}$ is contained in C.

Algebraically, a set C is convex if for any $\vec{x}_1, \ldots, \vec{x}_k \in C$, any convex combination of $\vec{x}_1, \ldots, \vec{x}_k$ is contained in C.

Definition 5.3 (Convex Hull)**.** Let $S \subseteq \mathbb{R}^n$ be a set. The *convex hull* of S, denoted $\text{conv}(S)$, is the set of all convex combinations of points in S:

$$\text{conv}(S) = \left\{ \sum_{i=1}^{k} \theta_i \vec{x}_i \;\middle|\; k \in \mathbb{N}, \theta_1, \ldots, \theta_k \geq 0, \sum_{i=1}^{k} \theta_i = 1, \vec{x}_1, \ldots, \vec{x}_k \in S \right\}. \tag{5.2}$$

Proposition 5.4. *Let $S \subseteq \mathbb{R}^n$ be a set.*

(a) conv(S) is a convex set.

(b) conv(S) is the minimal convex set containing S: conv$(S) = \bigcap_{C \text{ convex}, C \supseteq S} C$.

(c) conv(S) is the union of convex hulls of all finite subsets of S.

Thus if S is convex then $conv(S) = S$.

Theorem 5.5 (Carathéodory's Theorem)**.** *Let $S \subseteq \mathbb{R}^n$ be a set. Then $conv(S)$ is the union of convex hulls of all finite subsets of S of size at most $n+1$:*

$$conv(S) = \bigcup_{\substack{A \subseteq S \\ |A| \leq n+1}} conv(A). \tag{5.3}$$

5.1.2 Hyperplanes and Half-Spaces

Definition 5.6 (Hyperplane)**.** Let $\vec{a}, \vec{x}_0 \in \mathbb{R}^n$ and $b \in \mathbb{R}$. A *hyperplane* is a set of the form

$$\{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top \vec{x} = b\} \quad \text{or equivalently} \quad \{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top (\vec{x} - \vec{x}_0) = 0\}. \tag{5.4}$$

Example 5.7 (Hyperplanes are Convex)**.** Consider a hyperplane $H = \{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top \vec{x} = b\}$. Let $\vec{x}_1, \vec{x}_2 \in H$ and $\theta \in [0, 1]$. Then

$$\vec{a}^\top(\theta \vec{x}_1 + (1-\theta)\vec{x}_2) = \theta \vec{a}^\top \vec{x}_1 + (1-\theta)\vec{a}^\top \vec{x}_2 = \theta b + (1-\theta)b = b, \tag{5.5}$$

so $\theta \vec{x}_1 + (1-\theta)\vec{x}_2 \in H$. Thus H is convex.

Definition 5.8 (Half-Space)**.** Let $\vec{a}, \vec{x}_0 \in \mathbb{R}^n$ and $b \in \mathbb{R}$. A *positive half-space* is a set of the form

$$\{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top \vec{x} \geq b\} \quad \text{or} \quad \{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top (\vec{x} - \vec{x}_0) \geq 0\}. \tag{5.6}$$

A *negative half-space* is a set of the form

$$\{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top \vec{x} \leq b\} \quad \text{or} \quad \{\vec{x} \in \mathbb{R}^n \mid \vec{a}^\top (\vec{x} - \vec{x}_0) \leq 0\}. \tag{5.7}$$

The positive and negative half-spaces partition $\mathbb{R}^n$. A vector $\vec{x}$ is in the positive half-space if $\vec{x} - \vec{x}_0$ forms an acute angle with $\vec{a}$ (positive dot product), and in the negative half-space if it forms an obtuse angle (negative dot product).

Example 5.9 (Set of PSD Matrices is Convex)**.** Consider $\mathbb{S}^n_+$, the set of all symmetric positive semidefinite matrices. Take $A_1, A_2 \in \mathbb{S}^n_+$ and $\theta \in [0, 1]$. For any $\vec{x} \in \mathbb{R}^n$:

$$\vec{x}^\top(\theta A_1 + (1-\theta)A_2)\vec{x} = \theta \underbrace{\vec{x}^\top A_1 \vec{x}}_{\geq 0} + (1-\theta) \underbrace{\vec{x}^\top A_2 \vec{x}}_{\geq 0} \geq 0. \tag{5.8}$$

Thus $\mathbb{S}^n_+$ is convex.

Theorem 5.10 (Separating Hyperplane Theorem)**.** *Let $C, D \subseteq \mathbb{R}^n$ be two nonempty disjoint convex sets, i.e., $C \cap D = \emptyset$. Then there exists a hyperplane that separates C and D, i.e., there exists $\vec{a}, \vec{x}_0 \in \mathbb{R}^n$ such that*

$$\vec{a}^\top(\vec{x} - \vec{x}_0) \geq 0, \quad \forall \vec{x} \in C \tag{5.9}$$

$$\vec{a}^\top(\vec{x} - \vec{x}_0) \leq 0, \quad \forall \vec{x} \in D. \tag{5.10}$$

Moreover, if C is closed and D is closed and bounded, then there exists a hyperplane that separates C and D with strict inequalities.

Proof (sketch). Since C and D are disjoint and compact, define $\operatorname{dist}(C,D) = \min_{\vec{c}\in C,\vec{d}\in D} \left\|\vec{c}-\vec{d}\right\|_2 > 0$. Let $\vec{c}\in C$ and $\vec{d}\in D$ achieve this minimum. Set

$$\vec{a} = \vec{c}-\vec{d}, \quad \vec{x}_0 = \frac{\vec{c}+\vec{d}}{2}. \tag{5.11}$$

The hyperplane passing through $\vec{x}_0$ with normal $\vec{a}$ separates C and D. To verify, suppose for contradiction there exists $\vec{u}\in D$ with $\vec{a}^\top(\vec{u}-\vec{x}_0)\geq 0$. One can show that the line segment from $\vec{d}$ to $\vec{u}$ contains a point closer to $\vec{c}$ than $\vec{d}$, contradicting the minimality of $\left\|\vec{c}-\vec{d}\right\|_2$. □

5.1.3 (OPTIONAL) Cones

Definition 5.11 (Cones and Proper Cones)**.** Let $K\subseteq\mathbb{R}^n$.

(a) K is a *cone* if for any $\vec{v}\in K$ and $\alpha\geq 0$, we have $\alpha\vec{v}\in K$.

(b) K is a *convex cone* if it is both a cone and a convex set.

(c) K is a *pointed cone* if it contains no line through the origin.

(d) K is a *solid cone* if it has non-empty interior.

(e) K is a *closed cone* if it contains its boundary points.

(f) K is a *proper cone* if it is convex, pointed, solid, and closed.

Definition 5.12 (Dual Cone)**.** Let $K\subseteq\mathbb{R}^n$ be a cone. The *dual cone* of K is

$$K^\star = \{\vec{y}\in\mathbb{R}^n \mid \vec{y}^\top\vec{x}\geq 0 \text{ for each } \vec{x}\in K\}. \tag{5.12}$$

Proposition 5.13. *Let $K\subseteq\mathbb{R}^n$ be a cone. Then $K^\star$ is a closed convex cone.*

Example 5.14. (a) The non-negative orthant $\mathbb{R}^n_+ = \{\vec{x}\in\mathbb{R}^n \mid x_i\geq 0, \forall i\}$ is a proper cone, and its dual cone is itself.

(b) Let $S\subseteq\mathbb{R}^n$ be a subspace. Then S is a convex cone, and $S^\perp$ is its dual cone.

Definition 5.15 (Second Order Cone)**.** The *second-order cone* (or Lorentz cone) in $\mathbb{R}^{n+1}$ is

$$K = \{(\vec{x},t)\in\mathbb{R}^n\times\mathbb{R} \mid \|\vec{x}\|_2\leq t\}. \tag{5.13}$$

Proposition 5.16. *The second-order cone K is a proper cone, and its dual cone in $\mathbb{R}^{n+1}$ is itself.*

Proposition 5.17. *Let $\mathbb{S}^n$ be the vector space of $n\times n$ symmetric matrices with the Frobenius inner product $\langle A,B\rangle_F = tr(AB)$.*

(a) $\mathbb{S}^n_+$ is a proper cone in $\mathbb{S}^n$.

(b) The dual cone of $\mathbb{S}^n_+$ in $\mathbb{S}^n$ is itself.

Theorem 5.18. *Let $K\subseteq\mathbb{R}^n$ be a closed convex cone. Then $(K^\star)^\star = K$.*

5.2 Convex Functions

Definition 5.19 (Convex and Concave Functions)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$. We say f is *convex* if $\text{dom}(f)$ is convex and for all $\vec{x}_1, \vec{x}_2 \in \text{dom}(f)$ and all $\theta \in [0, 1]$:

$$f(\theta\vec{x}_1 + (1-\theta)\vec{x}_2) \leq \theta f(\vec{x}_1) + (1-\theta)f(\vec{x}_2). \tag{5.14}$$

We say f is *concave* if $-f$ is convex.

Geometrically, f is convex if the line segment connecting any two points on the graph of f lies above the graph.

Theorem 5.20 (Jensen's Inequality)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex. Then for any $\vec{x}_1, \ldots, \vec{x}_k \in dom(f)$ and any $\theta_1, \ldots, \theta_k \geq 0$ with $\sum_{i=1}^k \theta_i = 1$:*

$$f\left(\sum_{i=1}^k \theta_i \vec{x}_i\right) \leq \sum_{i=1}^k \theta_i f(\vec{x}_i). \tag{5.15}$$

Definition 5.21 (Epigraph)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$. The *epigraph* of f is

$$\text{epi}(f) = \{(\vec{x}, t) \in \mathbb{R}^{n+1} \mid \vec{x} \in \text{dom}(f), t \geq f(\vec{x})\}. \tag{5.16}$$

The epigraph is the set of points lying on or above the graph of f.

Proposition 5.22. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if epi(f) is a convex set.*

Theorem 5.23 (First-Order Condition for Convexity)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then f is convex if and only if dom(f) is convex and for all $\vec{x}, \vec{y} \in dom(f)$:*

$$f(\vec{y}) \geq f(\vec{x}) + [\nabla f(\vec{x})]^\top (\vec{y} - \vec{x}). \tag{5.17}$$

This says that for a convex function, the first-order Taylor approximation is a global underestimator.

Theorem 5.24 (Second-Order Condition for Convexity)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable. Then f is convex if and only if dom(f) is convex and $\nabla^2 f(\vec{x}) \succeq 0$ for all $\vec{x} \in dom(f)$.*

Example 5.25. (a) $f(\vec{x}) = \|\vec{x}\|_2^2$ is convex since $\nabla^2 f(\vec{x}) = 2I \succ 0$.

(b) $f(\vec{x}) = \left\|A\vec{x} - \vec{b}\right\|_2^2$ is convex since $\nabla^2 f(\vec{x}) = 2A^\top A \succeq 0$.

(c) $f(x) = e^x$ is convex since $f''(x) = e^x > 0$.

(d) $f(x) = \log(x)$ is concave since $f''(x) = -1/x^2 < 0$ for $x > 0$.

Definition 5.26 (Strictly Convex Function)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$. We say f is *strictly convex* if $\text{dom}(f)$ is convex and for all $\vec{x}_1 \neq \vec{x}_2 \in \text{dom}(f)$ and all $\theta \in (0, 1)$:

$$f(\theta\vec{x}_1 + (1-\theta)\vec{x}_2) < \theta f(\vec{x}_1) + (1-\theta)f(\vec{x}_2). \tag{5.18}$$

Theorem 5.27 (Conditions for Strict Convexity)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable. If $\nabla^2 f(\vec{x}) \succ 0$ for all $\vec{x} \in dom(f)$, then f is strictly convex.*

The converse is not true: $f(x) = x^4$ is strictly convex but $f''(0) = 0$.

Theorem 5.28. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then f is strictly convex if and only if dom(f) is convex and for all $\vec{x} \neq \vec{y} \in dom(f)$:*

$$f(\vec{y}) > f(\vec{x}) + [\nabla f(\vec{x})]^\top (\vec{y} - \vec{x}). \tag{5.19}$$

5.2.1 Affine Functions

Definition 5.29 (Affine Function)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is *affine* if it can be written as $f(\vec{x}) = \vec{a}^\top \vec{x} + b$ for some $\vec{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$.

Proposition 5.30. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is affine if and only if it is both convex and concave.*

5.3 Convex Optimization Problems

Definition 5.31 (Convex Optimization Problem)**.** An optimization problem is a *convex optimization problem* if:

(a) The objective function f is convex.

(b) The equality constraints are affine: $h_i(\vec{x}) = \vec{a}_i^\top \vec{x} - b_i$.

(c) The inequality constraints g_j are convex.

Theorem 5.32. *Let $\mathcal{F}$ be the feasible set of a convex optimization problem. Then $\mathcal{F}$ is a convex set.*

Proof. The feasible set is

$$\mathcal{F} = \mathrm{dom}(f) \cap \bigcap_{i=1}^{p} \{\vec{x} \mid h_i(\vec{x}) = 0\} \cap \bigcap_{j=1}^{q} \{\vec{x} \mid g_j(\vec{x}) \leq 0\}. \tag{5.20}$$

Each equality constraint defines a hyperplane (convex), each inequality constraint defines a sublevel set of a convex function (convex), and intersections of convex sets are convex. □

Theorem 5.33 (Local Minima are Global Minima)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function, and let $\mathcal{F} \subseteq \mathbb{R}^n$ be a convex set. Consider the optimization problem $\min_{\vec{x} \in \mathcal{F}} f(\vec{x})$. If $\vec{x}^\star$ is a local minimum, then $\vec{x}^\star$ is a global minimum.*

Proof. Suppose $\vec{x}^\star$ is a local but not global minimum. Then there exists $\vec{y} \in \mathcal{F}$ with $f(\vec{y}) < f(\vec{x}^\star)$. Consider the line segment from $\vec{x}^\star$ to $\vec{y}$. For $\theta \in (0, 1)$:

$$f(\theta \vec{y} + (1-\theta)\vec{x}^\star) \leq \theta f(\vec{y}) + (1-\theta) f(\vec{x}^\star) < f(\vec{x}^\star). \tag{5.21}$$

Taking θ small, $\theta\vec{y} + (1-\theta)\vec{x}^\star$ is arbitrarily close to $\vec{x}^\star$ but has smaller objective value, contradicting local minimality. □

Theorem 5.34 (Uniqueness for Strictly Convex Problems)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a strictly convex function, and let $\mathcal{F} \subseteq \mathbb{R}^n$ be a convex set. Consider the optimization problem $\min_{\vec{x} \in \mathcal{F}} f(\vec{x})$. If a solution exists, it is unique.*

Theorem 5.35 (First-Order Optimality Condition)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex and differentiable function. Consider $\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$. Then $\vec{x}^\star$ is a global minimum if and only if $\nabla f(\vec{x}^\star) = \vec{0}$.*

5.4 Solving Convex Optimization Problems

Definition 5.36 (Active and Inactive Constraints)**.** Consider the problem

$$\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) \quad \text{s.t.} \quad g_j(\vec{x}) \leq 0, \quad j = 1, \ldots, q. \tag{5.22}$$

At a feasible point $\vec{x}$, constraint g_j is *active* if $g_j(\vec{x}) = 0$ and *inactive* if $g_j(\vec{x}) < 0$.

Problem Solving Strategy 5.37 (Solving Convex Problems by Cases). Consider a convex optimization problem with inequality constraints. For each subset S of constraints:

1. Assume constraints in S are active (equality) and constraints not in S are inactive.
2. Solve the equality-constrained problem.
3. Check if the solution satisfies the original inequality constraints.
4. Among valid solutions, select the one with smallest objective value.

5.5 Problem Transformations

Proposition 5.38 (Monotone Transformations)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ and let $\phi : \mathbb{R} \to \mathbb{R}$ be a strictly monotonically increasing function. Then*

$$\operatorname*{argmin}_{\vec{x}\in\mathcal{F}} f(\vec{x}) = \operatorname*{argmin}_{\vec{x}\in\mathcal{F}} \phi(f(\vec{x})). \tag{5.23}$$

If ϕ is strictly monotonically decreasing, then

$$\operatorname*{argmin}_{\vec{x}\in\mathcal{F}} f(\vec{x}) = \operatorname*{argmax}_{\vec{x}\in\mathcal{F}} \phi(f(\vec{x})). \tag{5.24}$$

Example 5.39 (Logistic Regression)**.** In logistic regression, we maximize the log-likelihood:

$$\max_{\vec{\theta}} \sum_{i=1}^{n} \Big[y_i \log(\sigma(\vec{\theta}^\top \vec{x}_i)) + (1 - y_i) \log(1 - \sigma(\vec{\theta}^\top \vec{x}_i)) \Big] \tag{5.25}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function. This is equivalent to minimizing the negative log-likelihood, which is a convex function.

Definition 5.40 (Slack Variables)**.** Consider the problem

$$\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) \quad \text{s.t.} \quad g_j(\vec{x}) \leq 0, \quad j = 1, \ldots, q. \tag{5.26}$$

Introducing *slack variables* $s_j \geq 0$, this is equivalent to

$$\min_{\vec{x}\in\mathbb{R}^n, \vec{s}\in\mathbb{R}^q} f(\vec{x}) \quad \text{s.t.} \quad g_j(\vec{x}) + s_j = 0, \quad s_j \geq 0, \quad j = 1, \ldots, q. \tag{5.27}$$

Definition 5.41 (Epigraph Reformulation)**.** Consider $\min_{\vec{x}\in\mathcal{F}} f(\vec{x})$. The *epigraph reformulation* is

$$\min_{(\vec{x},t)\in\mathbb{R}^{n+1}} t \quad \text{s.t.} \quad \vec{x} \in \mathcal{F}, \quad f(\vec{x}) \leq t. \tag{5.28}$$

These problems have the same optimal value, and if $(\vec{x}^\star, t^\star)$ solves the epigraph form, then $\vec{x}^\star$ solves the original problem.

Example 5.42 (Elastic-Net Regression)**.** Consider the elastic-net problem:

$$\min_{\vec{x}\in\mathbb{R}^n} \left\{ \|A\vec{x} - \vec{y}\|_2^2 + \lambda_1 \|\vec{x}\|_1 + \lambda_2 \|\vec{x}\|_2^2 \right\}. \tag{5.29}$$

Using slack variables $\vec{t} \in \mathbb{R}^n$ with $|x_i| \leq t_i$, this becomes:

$$\min_{\vec{x},\vec{t}} \left\{ \|A\vec{x} - \vec{y}\|_2^2 + \lambda_1 \vec{1}^\top \vec{t} + \lambda_2 \|\vec{x}\|_2^2 \right\} \quad \text{s.t.} \quad -\vec{t} \leq \vec{x} \leq \vec{t}. \tag{5.30}$$

Chapter 6

Gradient Descent

6.1 Strong Convexity and Smoothness

Definition 6.1 (μ-Strongly Convex)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. We say f is *μ-strongly convex* (with parameter $\mu > 0$) if for all $\vec{x}, \vec{y} \in \text{dom}(f)$:

$$f(\vec{y}) \geq f(\vec{x}) + [\nabla f(\vec{x})]^\top (\vec{y} - \vec{x}) + \frac{\mu}{2} \|\vec{y} - \vec{x}\|_2^2 . \tag{6.1}$$

Strong convexity strengthens the first-order condition by adding a quadratic lower bound. It implies strict convexity.

Theorem 6.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable. Then f is μ-strongly convex if and only if $\nabla^2 f(\vec{x}) \succeq \mu I$ for all $\vec{x} \in dom(f)$.*

Theorem 6.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be μ-strongly convex with minimizer $\vec{x}^\star$. Then for all $\vec{x} \in dom(f)$:*

$$f(\vec{x}) - f(\vec{x}^\star) \geq \frac{\mu}{2} \|\vec{x} - \vec{x}^\star\|_2^2 . \tag{6.2}$$

Definition 6.4 (L-Smooth)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. We say f is *L-smooth* (with parameter $L > 0$) if for all $\vec{x}, \vec{y} \in \text{dom}(f)$:

$$f(\vec{y}) \leq f(\vec{x}) + [\nabla f(\vec{x})]^\top (\vec{y} - \vec{x}) + \frac{L}{2} \|\vec{y} - \vec{x}\|_2^2 . \tag{6.3}$$

L-smoothness provides a quadratic upper bound. For twice-differentiable functions, f is L-smooth if and only if $\nabla^2 f(\vec{x}) \preceq L \cdot I$ for all $\vec{x}$.

Example 6.5. For $f(\vec{x}) = \frac{1}{2}\vec{x}^\top A \vec{x}$ with $A \succ 0$:

- f is $\lambda_{\min}(A)$-strongly convex.
- f is $\lambda_{\max}(A)$-smooth.

6.2 Gradient Descent

Consider the unconstrained optimization problem $\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$ where f is convex and differentiable.

Gradient Descent Algorithm:

1. Initialize $\vec{x}^{(0)} \in \mathbb{R}^n$.
2. For $k = 0, 1, 2, \ldots$:
$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \eta_k \nabla f(\vec{x}^{(k)}) \tag{6.4}$$
where $\eta_k > 0$ is the *step size* (or *learning rate*).

The direction $-\nabla f(\vec{x}^{(k)})$ is the direction of steepest descent at $\vec{x}^{(k)}$.

6.2.1 Search Direction

Theorem 6.6 (Negative Gradient is Direction of Steepest Descent)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, and let $\vec{x} \in \mathbb{R}^n$. Then*

$$-\frac{\nabla f(\vec{x})}{\|\nabla f(\vec{x})\|_2} \in \operatorname*{argmin}_{\substack{\vec{v}\in\mathbb{R}^n \\ \|\vec{v}\|_2=1}} Df(\vec{x})[\vec{v}]. \tag{6.5}$$

6.2.2 Convergence Analysis of Gradient Descent

Example 6.7 (Gradient Descent for Least Squares)**.** For the least squares problem $\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2$ with A full column rank, the gradient descent update becomes:

$$\vec{x}_{t+1} = (I - 2\eta A^\top A)\vec{x}_t + 2\eta A^\top \vec{y}. \tag{6.6}$$

Subtracting $\vec{x}^\star = (A^\top A)^{-1} A^\top \vec{y}$:

$$\vec{x}_{t+1} - \vec{x}^\star = (I - 2\eta A^\top A)(\vec{x}_t - \vec{x}^\star). \tag{6.7}$$

If $\sigma_{\max}\{I - 2\eta A^\top A\} < 1$, then convergence is guaranteed with rate:

$$\|\vec{x}_t - \vec{x}^\star\|_2 \le \sigma_{\max}\{I - 2\eta A^\top A\}^t \|\vec{x}_0 - \vec{x}^\star\|_2 . \tag{6.8}$$

Lemma 6.8 (Gradient Bound for L-Smooth Functions)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an L-smooth function. For all $\vec{x} \in \mathbb{R}^n$:*

$$\|\nabla f(\vec{x})\|_2^2 \le 2L \left(f(\vec{x}) - \min_{\vec{x}'\in\mathbb{R}^n} f(\vec{x}') \right). \tag{6.9}$$

Theorem 6.9 (Convergence of Gradient Descent for Smooth Strongly Convex Functions)**.** *Let $\mu, L > 0$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be an L-smooth, μ-strongly convex function with optimal solution $\vec{x}^\star$. Then with constant step size $\eta = \frac{1}{L}$, the gradient descent iterates satisfy:*

$$\|\vec{x}_{t+1} - \vec{x}^\star\|_2^2 \le \left(1 - \frac{\mu}{L}\right) \|\vec{x}_t - \vec{x}^\star\|_2^2 . \tag{6.10}$$

Corollary 6.10. *With $0 \le 1 - \frac{\mu}{L} < 1$:*

(a) (Descent at every step) $\|\vec{x}_{t+1} - \vec{x}^\star\|_2 \le \|\vec{x}_t - \vec{x}^\star\|_2$.

(b) (Convergence) $\lim_{t\to\infty} \vec{x}_t = \vec{x}^\star$.

The convergence rate is $c = \sqrt{1 - \frac{\mu}{L}}$, and to achieve accuracy ϵ, we need $T \ge \frac{\log(1/\epsilon)+\log(D)}{\log(1/c)}$ iterations where $D = \|\vec{x}_0 - \vec{x}^\star\|_2$.

6.3 Variations: Stochastic Gradient Descent

For problems of the form $\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) = \min_{\vec{x}\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^m f_i(\vec{x})$, computing the full gradient $\nabla f(\vec{x}) = \frac{1}{m}\sum_{i=1}^m \nabla f_i(\vec{x})$ can be expensive.

Stochastic Gradient Descent (SGD) samples a random index i uniformly and updates:

$$\vec{x}_{t+1} = \vec{x}_t - \eta \nabla f_i(\vec{x}_t). \tag{6.11}$$

The expected value of the stochastic gradient equals the full gradient:

$$\mathbb{E}[\nabla f_i(\vec{x})] = \sum_{i=1}^m \frac{1}{m}\nabla f_i(\vec{x}) = \nabla f(\vec{x}). \tag{6.12}$$

SGD requires a variable step size η_t with $\lim_{t\to\infty}\eta_t = 0$ for convergence, and achieves averaged convergence:

$$\lim_{T\to\infty} f\left(\frac{1}{T}\sum_{t=1}^T \vec{x}_t\right) = \min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}). \tag{6.13}$$

6.4 Variations: Gradient Descent for Constrained Optimization

6.4.1 Projected Gradient Descent

Definition 6.11 (Projection onto a Convex Set)**.** Let Ω be a closed convex set. The *projection* of $\vec{y}\in\mathbb{R}^n$ onto Ω is

$$\text{proj}_\Omega(\vec{y}) = \operatorname*{argmin}_{\vec{x}\in\Omega} \|\vec{x}-\vec{y}\|_2^2. \tag{6.14}$$

The **Projected Gradient Descent** update is:

$$\vec{x}_{t+1} = \text{proj}_\Omega(\vec{x}_t - \eta\nabla f(\vec{x}_t)). \tag{6.15}$$

6.4.2 Conditional Gradient Descent (Frank-Wolfe)

Given $\vec{x}_t\in\Omega$, find the search direction:

$$\vec{v}_t = \operatorname*{argmin}_{\vec{v}\in\Omega} [\nabla f(\vec{x}_t)]^\top \vec{v}. \tag{6.16}$$

Update via convex combination with $\delta_t\in[0,1]$:

$$\vec{x}_{t+1} = (1-\delta_t)\vec{x}_t + \delta_t\vec{v}_t. \tag{6.17}$$

This ensures $\vec{x}_{t+1}\in\Omega$ by convexity. A conventional choice is $\delta_t = \frac{1}{t}$.

Chapter 7

Duality

7.1 Lagrangian

Consider the primal problem $\mathcal{P}$:

$$p^\star = \min_{\vec{x}\in\mathbb{R}^n} f_0(\vec{x}) \quad \text{s.t.} \quad f_i(\vec{x}) \leq 0,\ \forall i \in \{1,\dots,m\}, \quad h_j(\vec{x}) = 0,\ \forall j \in \{1,\dots,p\}. \tag{7.1}$$

Using indicator functions $\mathbf{1}[f_i(\vec{x}) \leq 0] = \max_{\lambda_i\in\mathbb{R}_+} \lambda_i f_i(\vec{x})$ and $\mathbf{1}[h_j(\vec{x}) = 0] = \max_{\nu_j\in\mathbb{R}} \nu_j h_j(\vec{x})$, we can write:

$$p^\star = \min_{\vec{x}\in\mathbb{R}^n} \max_{\substack{\vec{\lambda}\in\mathbb{R}^m_+ \\ \vec{\nu}\in\mathbb{R}^p}} L(\vec{x},\vec{\lambda},\vec{\nu}). \tag{7.2}$$

Definition 7.1 (Lagrangian)**.** The *Lagrangian* of problem $\mathcal{P}$ is $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ given by

$$L(\vec{x},\vec{\lambda},\vec{\nu}) = f_0(\vec{x}) + \sum_{i=1}^m \lambda_i f_i(\vec{x}) + \sum_{j=1}^p \nu_j h_j(\vec{x}). \tag{7.3}$$

The $\lambda_i, \nu_j \in \mathbb{R}$ are called *Lagrange multipliers.*

Proposition 7.2. *For every $\vec{x} \in \mathbb{R}^n$, the function $(\vec{\lambda},\vec{\nu}) \mapsto L(\vec{x},\vec{\lambda},\vec{\nu})$ is affine (hence concave) in $\vec{\lambda}$ and $\vec{\nu}$.*

7.2 Weak Duality

Definition 7.3 (Dual Problem)**.** The *dual problem* $\mathcal{D}$ is obtained by swapping min and max:

$$d^\star = \max_{\substack{\vec{\lambda}\in\mathbb{R}^m_+ \\ \vec{\nu}\in\mathbb{R}^p}} g(\vec{\lambda},\vec{\nu}) \quad \text{where} \quad g(\vec{\lambda},\vec{\nu}) = \min_{\vec{x}\in\mathbb{R}^n} L(\vec{x},\vec{\lambda},\vec{\nu}) \tag{7.4}$$

is the *dual function.*

Proposition 7.4. *The dual function g is a concave function of $(\vec{\lambda},\vec{\nu})$, regardless of any properties of $\mathcal{P}$.*

Corollary 7.5. *The dual problem $\mathcal{D}$ is always a convex problem, no matter what the primal problem $\mathcal{P}$ is.*

Definition 7.6 (Types of Duality)**.** Let $\mathcal{P}$ have optimum $p^\star$ and $\mathcal{D}$ have optimum $d^\star$.

(a) If $p^\star \geq d^\star$, we say *weak duality* holds.

(b) If $p^\star = d^\star$, we say *strong duality* holds.

(c) The quantity $p^\star - d^\star$ is called the *duality gap.*

Proposition 7.7 (Minimax Inequality)**.** *Let X and Y be any sets, and $F : X \times Y \to \mathbb{R}$ be any function. Then*

$$\min_{x \in X} \max_{y \in Y} F(x, y) \geq \max_{y \in Y} \min_{x \in X} F(x, y). \tag{7.5}$$

Theorem 7.8 (Weak Duality Always Holds)**.** *For any problem, weak duality holds, i.e., the duality gap is non-negative: $p^\star \geq d^\star$.*

Weak duality provides a *certificate of optimality*: if $f_0(\vec{x}) - g(\vec{\lambda}, \vec{\nu}) \leq \epsilon$, then $f_0(\vec{x}) - p^\star \leq \epsilon$.

7.3 Strong Duality

Theorem 7.9 (Slater's Condition)**.** *Suppose $f_0, f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex functions, and $h_1, \ldots, h_p : \mathbb{R}^n \to \mathbb{R}$ are affine functions. If there exists $\vec{x} \in \mathit{relint}(\Omega)$ which is* strictly feasible*, i.e.:*

- *For all i such that f_i is affine: $f_i(\vec{x}) \leq 0$.*
- *For all i such that f_i is not affine: $f_i(\vec{x}) < 0$.*
- *For all j: $h_j(\vec{x}) = 0$.*

Then strong duality holds: $p^\star = d^\star$.

Example 7.10 (Equality-Constrained Minimum-Norm Problem)**.** For $\min_{\vec{x} \in \mathbb{R}^n} \|\vec{x}\|_2^2$ subject to $A\vec{x} = \vec{y}$:

The Lagrangian is $L(\vec{x}, \vec{\nu}) = \|\vec{x}\|_2^2 + \vec{\nu}^\top (A\vec{x} - \vec{y})$.

Setting $\nabla_{\vec{x}} L = \vec{0}$: $\vec{x}^\star(\vec{\nu}) = -\frac{1}{2} A^\top \vec{\nu}$.

The dual function: $g(\vec{\nu}) = -\frac{1}{4} \vec{\nu}^\top A A^\top \vec{\nu} - \vec{\nu}^\top \vec{y}$.

Solving the unconstrained dual: $\vec{\nu}^\star = -2(AA^\top)^{-1}\vec{y}$.

By strong duality, the optimal primal solution is:

$$\vec{x}^\star = A^\top (AA^\top)^{-1} \vec{y}. \tag{7.6}$$

Example 7.11 (Linear Program)**.** For $\min_{\vec{x} \in \mathbb{R}^n} \vec{c}^\top \vec{x}$ subject to $A\vec{x} = \vec{y}$, $\vec{x} \geq \vec{0}$:

The Lagrangian is $L(\vec{x}, \vec{\lambda}, \vec{\nu}) = (\vec{c} + A^\top \vec{\nu} - \vec{\lambda})^\top \vec{x} - \vec{\nu}^\top \vec{y}$.

The dual function:

$$g(\vec{\lambda}, \vec{\nu}) = \begin{cases} -\vec{\nu}^\top \vec{y} & \text{if } \vec{c} + A^\top \vec{\nu} = \vec{\lambda} \\ -\infty & \text{otherwise} \end{cases} \tag{7.7}$$

7.4 Karush-Kuhn-Tucker (KKT) Conditions

Definition 7.12 (KKT Conditions)**.** Let $(\tilde{\vec{x}}, \tilde{\vec{\lambda}}, \tilde{\vec{\nu}}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$. We say $(\tilde{\vec{x}}, \tilde{\vec{\lambda}}, \tilde{\vec{\nu}})$ fulfills the *KKT conditions* if:

1. **Primal feasibility:** $f_i(\tilde{\vec{x}}) \leq 0$ for all i and $h_j(\tilde{\vec{x}}) = 0$ for all j.
2. **Dual feasibility:** $\tilde{\lambda}_i \geq 0$ for all i.
3. **Complementary slackness:** $\tilde{\lambda}_i f_i(\tilde{\vec{x}}) = 0$ for all i.
4. **Stationarity:** $\vec{0} = \nabla f_0(\tilde{\vec{x}}) + \sum_{i=1}^m \tilde{\lambda}_i \nabla f_i(\tilde{\vec{x}}) + \sum_{j=1}^p \tilde{\nu}_j \nabla h_j(\tilde{\vec{x}})$.

Theorem 7.13 (KKT Conditions are Necessary under Strong Duality)**.** *Suppose strong duality holds and $(\vec{x}^\star, \vec{\lambda}^\star, \vec{\nu}^\star)$ are optimal primal and dual variables. Then $(\vec{x}^\star, \vec{\lambda}^\star, \vec{\nu}^\star)$ fulfill the KKT conditions.*

Theorem 7.14 (KKT Conditions are Sufficient under Convexity)**.** *Suppose $f_0, f_1, \ldots, f_m$ are convex, $h_1, \ldots, h_p$ are affine, and $(\tilde{\vec{x}}, \tilde{\vec{\lambda}}, \tilde{\vec{\nu}})$ fulfill the KKT conditions. Then strong duality holds and $(\tilde{\vec{x}}, \tilde{\vec{\lambda}}, \tilde{\vec{\nu}})$ are optimal.*

Corollary 7.15. *If $\mathcal{P}$ is convex and strong duality holds, then KKT conditions are necessary and sufficient for optimality.*

Problem Solving Strategy 7.16 (Solving Convex Problems Using KKT).

1. Show the problem is convex and differentiable.
2. Show Slater's condition/strong duality holds.
3. Compute the KKT conditions.
4. Solve for optimal primal and dual variables.

Chapter 8

Types of Optimization Problems

8.1 Linear Programs

Definition 8.1 (Linear Program)**.** A *linear program* (LP) has an affine objective and affine constraints. Standard form:

$$p^\star = \min_{\vec{x} \in \mathbb{R}^n} \vec{c}^\top \vec{x} \quad \text{s.t.} \quad A\vec{x} = \vec{y}, \quad \vec{x} \geq \vec{0}. \tag{8.1}$$

Proposition 8.2. *Any linear program is equivalent to a standard form linear program.*

Proposition 8.3. *Any linear program is a convex optimization problem.*

Definition 8.4 (Polyhedron, Polygon)**.** A *polyhedron* is an intersection of a finite number of half-spaces. A *polygon* is a bounded polyhedron.

Definition 8.5 (Extreme Point, Vertex)**.** Let $K \subseteq \mathbb{R}^n$. We say $\vec{x} \in K$ is an *extreme point* if there do not exist $\vec{y}, \vec{z} \in K \setminus \{\vec{x}\}$ and $\theta \in [0, 1]$ such that $\vec{x} = \theta\vec{y} + (1 - \theta)\vec{z}$.

An extreme point of a polyhedron is called a *vertex*.

Proposition 8.6. *A polygon has finitely many vertices and is the convex hull of its vertices.*

Theorem 8.7 (Main Theorem of Linear Programming)**.** *For a standard form LP with bounded feasible set Ω, the optimal value is achieved at a vertex.*

The **Simplex Method**: Start at a vertex, move to neighboring vertices with better objective values until no improvement is possible.

Proposition 8.8 (Dual of Standard Form LP)**.** *The dual of $\min_{\vec{x}} \vec{c}^\top \vec{x}$ s.t. $A\vec{x} = \vec{y}$, $\vec{x} \geq \vec{0}$ is:*

$$d^\star = \max_{\vec{\lambda}, \vec{\nu}} -\vec{y}^\top \vec{\nu} \quad \text{s.t.} \quad \vec{c} - \vec{\lambda} + A^\top \vec{\nu} = \vec{0}, \quad \vec{\lambda} \geq \vec{0}. \tag{8.2}$$

8.2 Quadratic Programs

Definition 8.9 (Quadratic Program)**.** A *quadratic program* (QP) has a quadratic objective and affine constraints. Standard form:

$$p^\star = \min_{\vec{x} \in \mathbb{R}^n} \frac{1}{2}\vec{x}^\top H\vec{x} + \vec{c}^\top \vec{x} \quad \text{s.t.} \quad A\vec{x} \leq \vec{y}, \quad C\vec{x} = \vec{z}, \tag{8.3}$$

where $H \in \mathbb{S}^n$.

Proposition 8.10. *A standard form QP is convex if and only if $H \in \mathbb{S}^n_+$.*

Example 8.11 (Linear-Quadratic Regulator)**.** For a discrete-time system $\vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t$ with initial state $\vec{x}_0 = \vec{\xi}$, reaching goal $\vec{g}$:

$$\min_{\vec{x}_0,\ldots,\vec{x}_T,\vec{u}_0,\ldots,\vec{u}_{T-1}} \|\vec{x}_T - \vec{g}\|_2^2 + \sum_{k=0}^{T-1} \|\vec{u}_k\|_2^2 \quad \text{s.t.} \quad \vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t. \tag{8.4}$$

This is a QP since the objective is quadratic and constraints are affine.

8.3 Quadratically-Constrained Quadratic Programs

Definition 8.12 (QCQP)**.** A *quadratically-constrained quadratic program* (QCQP) has a quadratic objective and quadratic constraints:

$$p^\star = \min_{\vec{x}\in\mathbb{R}^n} \frac{1}{2}\vec{x}^\top H\vec{x} + \vec{c}^\top\vec{x} \quad \text{s.t.} \quad \frac{1}{2}\vec{x}^\top P_i\vec{x} + \vec{b}_i^\top\vec{x} + c_i \le 0, \quad \frac{1}{2}\vec{x}^\top Q_j\vec{x} + \vec{d}_j^\top\vec{x} + f_j = 0. \tag{8.5}$$

Proposition 8.13. *A QCQP is convex if $H, P_1, \ldots, P_m \in \mathbb{S}^n_+$ and $Q_1 = \cdots = Q_p = 0$ (i.e., equality constraints are affine).*

8.4 Second-Order Cone Programs

Definition 8.14 (Second-Order Cone Program)**.** A *second-order cone program* (SOCP) has a linear objective and second-order cone constraints:

$$p^\star = \min_{\vec{x}\in\mathbb{R}^n} \vec{c}^\top\vec{x} \quad \text{s.t.} \quad \|A_i\vec{x} - \vec{y}_i\|_2 \le \vec{b}_i^\top\vec{x} + z_i, \quad \forall i \in \{1,\ldots,m\}. \tag{8.6}$$

Proposition 8.15. *Second-order cone programs are convex optimization problems.*

Example 8.16. The problem $\min_{\vec{x}} \sum_{i=1}^m \|A_i\vec{x} - \vec{y}_i\|_2$ can be written as an SOCP using slack variables:

$$\min_{\vec{x},\vec{s}} \sum_{i=1}^m s_i \quad \text{s.t.} \quad \|A_i\vec{x} - \vec{y}_i\|_2 \le s_i. \tag{8.7}$$

Example 8.17 (Minimax Problem)**.** $\min_{\vec{x}} \max_i \|A_i\vec{x} - \vec{y}_i\|_2$ becomes:

$$\min_{\vec{x},s} s \quad \text{s.t.} \quad \|A_i\vec{x} - \vec{y}_i\|_2 \le s, \quad \forall i. \tag{8.8}$$

Hierarchy: LP $\subset$ QP $\subset$ QCQP $\subset$ SOCP. Any QCQP with $H, P_i \in \mathbb{S}^n_+$ can be reformulated as an SOCP using the identity $(u+v)^2 - (u-v)^2 = 4uv$.

Theorem 8.18. *The dual of an SOCP can be formulated as an SOCP in standard form.*

8.5 (OPTIONAL) Semidefinite Programming

Definition 8.19 (Semidefinite Program — Inequality Form)**.** A *semidefinite program* (SDP) in inequality form is

$$p^\star = \min_{X \in \mathbb{S}^n} \langle C, X \rangle \quad \text{s.t.} \quad \langle A_i, X \rangle \leq b_i, \ \forall i \in \{1, \ldots, m\}, \quad X \succeq 0, \tag{8.9}$$

where $C, A_1, \ldots, A_m \in \mathbb{S}^n$, $\vec{b} \in \mathbb{R}^m$, and $\langle A, B \rangle = \operatorname{tr}(AB)$ is the Frobenius inner product.

Definition 8.20 (Semidefinite Program — Standard Form)**.** A semidefinite program in standard form is

$$p^\star = \min_{X \in \mathbb{S}^n} \langle C, X \rangle \quad \text{s.t.} \quad \langle A_i, X \rangle = b_i, \ \forall i \in \{1, \ldots, m\}, \quad X \succeq 0. \tag{8.10}$$

Proposition 8.21. *Semidefinite programs are convex optimization problems.*

Proof. The objective $\langle C, X \rangle = \operatorname{tr}(CX)$ is linear in X. Each constraint $\langle A_i, X \rangle = b_i$ is linear in X. The constraint $X \succeq 0$ is convex since $\mathbb{S}^n_+$ is a convex cone. □

Theorem 8.22 (Dual of Standard Form SDP)**.** *The dual of the standard form SDP is*

$$d^\star = \max_{\vec{y} \in \mathbb{R}^m} \vec{b}^\top \vec{y} \quad \text{s.t.} \quad C - \sum_{i=1}^{m} y_i A_i \succeq 0. \tag{8.11}$$

Theorem 8.23. *If an SDP has a strictly feasible primal solution (i.e., $X \succ 0$), then strong duality holds.*

8.6 General Taxonomy

The hierarchy of convex optimization problem classes is:

$$\text{LP} \subset \text{Convex QP} \subset \text{Convex QCQP} \subset \text{SOCP} \subset \text{SDP} \subset \text{Convex Problems}. \tag{8.12}$$

Each inclusion is proper:

- LP: Linear objectives and linear constraints.
- QP: Quadratic objectives and linear constraints.
- QCQP: Quadratic objectives and quadratic constraints.
- SOCP: Linear objectives with second-order cone constraints.
- SDP: Linear objectives with semidefinite constraints.

In terms of computational complexity, problems higher in the hierarchy are generally harder to solve.

Chapter 9

Regularization and Sparsity

9.1 Ridge Regression and LASSO

Recall that in least squares, we solve $\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2$. When the solution is not unique or is sensitive to noise, we use *regularization*.

Definition 9.1 (Regularization)**.** A *regularized* least squares problem has the form

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 + \lambda R(\vec{x}) \tag{9.1}$$

where $\lambda > 0$ is the *regularization parameter* and $R : \mathbb{R}^n \to \mathbb{R}$ is the *regularizer*.

Ridge Regression uses the ℓ_2-norm squared regularizer $R(\vec{x}) = \|\vec{x}\|_2^2$:

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 + \lambda \|\vec{x}\|_2^2 . \tag{9.2}$$

This is a convex QP with closed-form solution $\vec{x}^\star = (A^\top A + \lambda I)^{-1} A^\top \vec{y}$.

Definition 9.2 (LASSO)**.** The *LASSO* (Least Absolute Shrinkage and Selection Operator) uses the ℓ_1-norm regularizer:

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 + \lambda \|\vec{x}\|_1 . \tag{9.3}$$

Proposition 9.3. *LASSO is a convex optimization problem.*

Proof. The objective $\|A\vec{x} - \vec{y}\|_2^2 + \lambda \|\vec{x}\|_1$ is the sum of a convex quadratic function and a convex norm function, hence convex. □

9.2 Understanding ℓ_2-Norm vs ℓ_1-Norm

The key difference between ridge regression and LASSO lies in the geometry of their unit balls:

- The ℓ_2-ball $\{\vec{x} : \|\vec{x}\|_2 \le 1\}$ is smooth (a sphere).
- The ℓ_1-ball $\{\vec{x} : \|\vec{x}\|_1 \le 1\}$ has "corners" at the coordinate axes.

LASSO promotes *sparsity*: solutions tend to have many zero components. This is because the ℓ_1-ball's corners lie on the coordinate axes, and the optimal solution often occurs at these corners.

Example 9.4 (Least ℓ_1-Norm as LP)**.** The problem $\min_{\vec{x}\in\mathbb{R}^n} \|\vec{x}\|_1$ subject to $A\vec{x} = \vec{y}$ can be reformulated as a linear program. Introduce variables $\vec{t} \in \mathbb{R}^n$ such that $-t_i \le x_i \le t_i$:

$$\min_{\vec{x},\vec{t}} \vec{1}^\top \vec{t} \quad \text{s.t.} \quad A\vec{x} = \vec{y}, \quad -\vec{t} \le \vec{x} \le \vec{t}. \tag{9.4}$$

Example 9.5 (Mean vs Median)**.** Consider fitting a constant c to data $y_1, \dots, y_n$:

- ℓ_2 loss: $\min_c \sum_{i=1}^n (c - y_i)^2$ gives $c^\star = \frac{1}{n}\sum_{i=1}^n y_i$ (mean).
- ℓ_1 loss: $\min_c \sum_{i=1}^n |c - y_i|$ gives $c^\star = \text{median}(y_1, \dots, y_n)$.

The median is more robust to outliers than the mean.

9.3 Analysis of LASSO

Consider the scalar LASSO problem:

$$\min_{x\in\mathbb{R}} (y - x)^2 + \lambda|x|. \tag{9.5}$$

Let $f(x) = (y-x)^2 + \lambda|x|$. Since $|x|$ is not differentiable at $x = 0$, we analyze three cases:

Case 1: $x > 0$. Then $f(x) = (y-x)^2 + \lambda x$, so $f'(x) = -2(y-x) + \lambda = 0$ gives $x = y - \frac{\lambda}{2}$.

Case 2: $x < 0$. Then $f(x) = (y-x)^2 - \lambda x$, so $f'(x) = -2(y-x) - \lambda = 0$ gives $x = y + \frac{\lambda}{2}$.

Case 3: $x = 0$. Check if $0 \in \partial f(0)$ using subgradients.

The solution is the **soft thresholding operator**:

$$x^\star = S_{\lambda/2}(y) = \begin{cases} y - \frac{\lambda}{2} & \text{if } y > \frac{\lambda}{2} \\ 0 & \text{if } |y| \le \frac{\lambda}{2} \\ y + \frac{\lambda}{2} & \text{if } y < -\frac{\lambda}{2} \end{cases} = \text{sign}(y) \cdot \max\left(|y| - \frac{\lambda}{2}, 0\right). \tag{9.6}$$

This shows that small values of y (within $\pm\frac{\lambda}{2}$) are "shrunk" to zero, promoting sparsity.

9.4 Geometry of LASSO

Theorem 9.6 (Equivalence of Regularized and Constrained Problems)**.** *Let $\lambda > 0$. Then LASSO is equivalent to the constrained problem:*

$$\min_{\vec{x}\in\mathbb{R}^n} \|A\vec{x} - \vec{y}\|_2^2 \quad \text{s.t.} \quad \|\vec{x}\|_1 \le t \tag{9.7}$$

for some $t = t(\lambda) > 0$. Specifically, if $\vec{x}^\star$ solves LASSO with parameter λ, then $\vec{x}^\star$ solves the constrained problem with $t = \|\vec{x}^\star\|_1$.

Proof. By strong duality (Slater's condition holds), there exists $\mu \geq 0$ such that $\vec{x}^\star$ minimizes the Lagrangian

$$L(\vec{x}, \mu) = \|A\vec{x} - \vec{y}\|_2^2 + \mu(\|\vec{x}\|_1 - t). \tag{9.8}$$

Setting $\mu = \lambda$ gives the equivalence. □

Geometrically, the constrained LASSO seeks the point where the sublevel sets of $\|A\vec{x} - \vec{y}\|_2^2$ (ellipsoids centered at $(A^\top A)^{-1}A^\top \vec{y}$) first touch the ℓ_1-ball $\{\vec{x} : \|\vec{x}\|_1 \leq t\}$. Due to the corners of the ℓ_1-ball, this contact point often occurs at a corner, yielding a sparse solution.

Figure: *LASSO geometry: Ellipsoidal sublevel sets touching the ℓ_1-ball at a corner*

Chapter 10

Advanced Descent Methods

10.1 Coordinate Descent

Recall the general idea of descent-based methods: start with an initial guess $\vec{x}^{(0)} \in \mathbb{R}^n$, then generate a sequence of refined guesses using the update rule

$$\vec{x}^{(t+1)} = \vec{x}^{(t)} + \eta \vec{v}^{(t)} \tag{10.1}$$

for some search direction $\vec{v}^{(t)}$ and step size η.

Coordinate descent finds a minimizer of multivariate functions by iteratively minimizing along one coordinate at a time. Consider $\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$.

For notation, let $\vec{x}_{i:j} = (x_i, x_{i+1}, \ldots, x_j) \in \mathbb{R}^{j-i+1}$ denote entries between indices i and j.

Coordinate Descent Algorithm: Given $\vec{x}^{(0)}$, for $t \geq 0$ update by sequentially minimizing with respect to each coordinate:

$$x_i^{(t+1)} \in \operatorname*{argmin}_{x_i \in \mathbb{R}} f(\vec{x}_{1:i-1}^{(t+1)}, x_i, \vec{x}_{i+1:n}^{(t)}). \tag{10.2}$$

This breaks down the difficult multivariate optimization problem into a sequence of simpler univariate problems.

Theorem 10.1 (Convergence of Coordinate Descent for Differentiable Convex Functions)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable convex function which is* separately strictly convex *in each argument. That is, for each i and each fixed $\vec{x}_{1:i-1}$ and $\vec{x}_{i+1:n}$, the function $x_i \mapsto f(\vec{x}_{1:i-1}, x_i, \vec{x}_{i+1:n})$ is strictly convex. If the coordinate descent algorithm is well-posed and $\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$ has a solution, then the sequence of iterates converges to an optimal solution.*

Coordinate descent may not converge for general non-differentiable convex functions. However, it converges for functions of the form

$$f(\vec{x}) = g(\vec{x}) + \sum_{i=1}^{n} h_i(x_i) \tag{10.3}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and each $h_i : \mathbb{R} \to \mathbb{R}$ is convex (but not necessarily differentiable). This includes ℓ_1 regularization problems like LASSO.

Example 10.2 (Coordinate Descent for LASSO)**.** For $A \in \mathbb{R}^{m\times n}$ with columns $\vec{a}_1, \ldots, \vec{a}_n$ and $\vec{y} \in \mathbb{R}^m$, consider

$$f(\vec{x}) = \frac{1}{2} \|A\vec{x} - \vec{y}\|_2^2 + \lambda \|\vec{x}\|_1 . \tag{10.4}$$

The coordinate descent update has closed form. Let $A_{i:j}$ denote the submatrix with columns i through j. Define $\vec{r}_i = \vec{y} - A_{1:i-1}\vec{x}_{1:i-1}^{(t+1)} - A_{i+1:n}\vec{x}_{i+1:n}^{(t)}$. Then:

$$x_i^{(t+1)} = \begin{cases} \frac{1}{\|\vec{a}_i\|_2^2}\left(\vec{a}_i^\top \vec{r}_i - \lambda\right) & \text{if } \vec{a}_i^\top \vec{r}_i > \lambda \\ 0 & \text{if } |\vec{a}_i^\top \vec{r}_i| \le \lambda \\ \frac{1}{\|\vec{a}_i\|_2^2}\left(\vec{a}_i^\top \vec{r}_i + \lambda\right) & \text{if } \vec{a}_i^\top \vec{r}_i < -\lambda \end{cases} \tag{10.5}$$

10.2 Newton's Method

Consider $\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x})$ where f is twice differentiable. Algorithms that utilize second derivatives (e.g., the Hessian) are called *second-order methods*.

Newton's method is based on the following idea: start with $\vec{x}^{(0)}$, then in each iteration t, approximate $f(\vec{x})$ with its second-order Taylor approximation around $\vec{x}^{(t)}$. The minimizer of this quadratic approximation becomes $\vec{x}^{(t+1)}$.

The second-order Taylor approximation around $\vec{x}^{(t)}$ is:

$$\hat{f}_2(\vec{x}; \vec{x}^{(t)}) = f(\vec{x}^{(t)}) + [\nabla f(\vec{x}^{(t)})]^\top (\vec{x} - \vec{x}^{(t)}) + \frac{1}{2}(\vec{x} - \vec{x}^{(t)})^\top [\nabla^2 f(\vec{x}^{(t)})](\vec{x} - \vec{x}^{(t)}). \tag{10.6}$$

If $\nabla^2 f(\vec{x}^{(t)}) \succ 0$, setting the gradient to zero yields:

$$\vec{x}^{(t+1)} = \vec{x}^{(t)} - [\nabla^2 f(\vec{x}^{(t)})]^{-1}[\nabla f(\vec{x}^{(t)})]. \tag{10.7}$$

The vector $[\nabla^2 f(\vec{x}^{(t)})]^{-1}[\nabla f(\vec{x}^{(t)})]$ is called the *Newton direction*.

The basic Newton's method is not guaranteed to converge in general. **Damped Newton's method** introduces a step size $\eta > 0$:

$$\vec{x}^{(t+1)} = \vec{x}^{(t)} - \eta[\nabla^2 f(\vec{x}^{(t)})]^{-1}[\nabla f(\vec{x}^{(t)})]. \tag{10.8}$$

Newton's method requires computing and inverting the Hessian in every iteration, which is more expensive than computing the gradient. However, Newton's method often converges in fewer iterations than gradient descent.

10.3 Newton's Method with Linear Equality Constraints

Consider the equality-constrained problem:

$$\min_{\vec{x}\in\mathbb{R}^n} f(\vec{x}) \quad \text{s.t.} \quad A\vec{x} = \vec{y} \tag{10.9}$$

where f is twice-differentiable strictly convex with positive definite Hessian.

We minimize the second-order Taylor approximation over the constraint set. This gives a constrained QP that we solve via KKT conditions. Defining $\vec{v}^{(t)} = \vec{x}^\star - \vec{x}^{(t)}$, the system becomes:

$$\begin{bmatrix} \nabla^2 f(\vec{x}^{(t)}) & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \vec{v}^{(t)} \\ \vec{\nu} \end{bmatrix} = \begin{bmatrix} -\nabla f(\vec{x}^{(t)}) \\ \vec{0} \end{bmatrix}. \tag{10.10}$$

After solving for $\vec{v}^{(t)}$, the update is $\vec{x}^{(t+1)} = \vec{x}^{(t)} + \vec{v}^{(t)}$.

10.4 (OPTIONAL) Interior Point Method

Interior point methods solve convex problems with inequality constraints:

$$\min_{\vec{x} \in \mathbb{R}^n} f_0(\vec{x}) \quad \text{s.t.} \quad f_i(\vec{x}) \leq 0, \ \forall i = 1, \ldots, m, \quad A\vec{x} = \vec{y} \tag{10.11}$$

where $f_0, f_1, \ldots, f_m$ are convex and twice-differentiable.

10.4.1 Barrier Functions

To eliminate inequality constraints, we augment them to the objective using a *barrier function* ϕ that approximates the indicator function $I(z) = 0$ if $z \leq 0$, $+\infty$ otherwise.

The **logarithmic barrier function** is:

$$\phi_\alpha(z) = -\frac{1}{\alpha} \log(-z) \tag{10.12}$$

where $\alpha > 0$ controls the approximation accuracy (larger α gives better approximation).

The approximate problem becomes:

$$\min_{\vec{x} \in \mathbb{R}^n} f_0(\vec{x}) + \sum_{i=1}^{m} \phi_\alpha(f_i(\vec{x})) \quad \text{s.t.} \quad A\vec{x} = \vec{y}. \tag{10.13}$$

10.4.2 Barrier Method

The **barrier method** overcomes numerical difficulties by solving a sequence of approximate problems with increasing α:

1. Start with small $\alpha^{(0)}$ and strictly feasible $\vec{x}^{(0)}$.
2. For $t = 1, 2, \ldots$:
   - Solve the approximate problem with $\alpha^{(t-1)}$ using Newton's method, starting at $\vec{x}^{(t-1)}$.
   - Update $\alpha^{(t)} = \mu\alpha^{(t-1)}$ for some $\mu > 1$.

This "easy-to-hard" approach uses solutions from easier problems as initial guesses for harder ones.

Chapter 11

Applications

11.1 (OPTIONAL) Deterministic Control and Linear-Quadratic Regulator

Control applies to any dynamical system where the state depends on time via $\vec{x}_{t+1} = \vec{f}(\vec{x}_t, \vec{u}_t)$, taking state $\vec{x}_t$ and control input $\vec{u}_t$ to produce next state $\vec{x}_{t+1}$.

A *discrete linear time-invariant* system has the form:

$$\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k, \quad \forall k \in \{0, 1, \ldots, K-1\}. \tag{11.1}$$

Definition 11.1 (Linear Quadratic Regulator (LQR))**.** Let $K \geq 0$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $Q, Q_f \in \mathbb{S}^n_+$, $R \in \mathbb{S}^m_{++}$, and $\vec{\xi} \in \mathbb{R}^n$. The LQR problem is:

$$\min_{(\vec{x}_k)_{k=0}^K, (\vec{u}_k)_{k=0}^{K-1}} \frac{1}{2}\sum_{k=0}^{K-1} (\vec{x}_k^\top Q\vec{x}_k + \vec{u}_k^\top R\vec{u}_k) + \frac{1}{2}\vec{x}_K^\top Q_f \vec{x}_K \tag{11.2}$$

subject to $\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k$ for all k and $\vec{x}_0 = \vec{\xi}$.

This is a QP with $(K+1)n + Km$ variables. It can be solved efficiently using the Riccati equation.

Theorem 11.2 (Optimal Control in LQR is Linear)**.** *An optimal control for the LQR problem is linear in the state:*

$$\vec{u}_k^\star = -R^{-1}B^\top (I + P_{k+1}BR^{-1}B^\top)^{-1} P_{k+1} A\vec{x}_k^\star \tag{11.3}$$

where P_k are given by the Riccati recurrence*:*

$$P_K = Q_f \tag{11.4}$$

$$P_k = A^\top (I + P_{k+1}BR^{-1}B^\top)^{-1} P_{k+1} A + Q, \quad \forall k \in \{0, \ldots, K-1\}. \tag{11.5}$$

The P_k can be computed backwards from $k = K$ to $k = 0$ offline. Then the optimal trajectory is computed forward using matrix multiplication.

11.2 Support Vector Machines

In binary classification, we have data $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{R}^d$ with labels $y_1, \ldots, y_n \in \{-1, +1\}$. We want to find a classifier $f : \mathbb{R}^d \to \{-1, +1\}$ such that $f(\vec{x}_i) = y_i$.

Support vector machines find an affine function $g_{\vec{w},b}(\vec{x}) = \vec{w}^\top \vec{x} - b$ that separates the data, with $f = \operatorname{sgn} \circ g_{\vec{w},b}$. Geometrically, this corresponds to the hyperplane $\mathcal{H}_{\vec{w},b} = \{\vec{x} \in \mathbb{R}^d : \vec{w}^\top \vec{x} = b\}$.

11.2.1 Hard-Margin SVM

Assuming data are strictly linearly separable, we want the $(\vec{w}, b)$ pair with the largest *margin* (distance from hyperplane to closest point). After simplification, this becomes:

$$\min_{\vec{w} \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2} \|\vec{w}\|_2^2 \quad \text{s.t.} \quad y_i(\vec{w}^\top \vec{x}_i - b) \geq 1, \ \forall i \in \{1, \ldots, n\}. \tag{11.6}$$

This is a quadratic program in $(\vec{w}, b)$.

11.2.2 Soft-Margin SVM

For non-separable data, we relax the hard margin constraint using the *hinge loss* $\ell_{\text{hinge}}(z) = \max\{z, 0\}$. Introducing slack variables $\vec{\xi}$:

$$\min_{\vec{w}, b, \vec{\xi}} \frac{1}{2} \|\vec{w}\|_2^2 + C \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad \xi_i \geq 0, \ \xi_i \geq 1 - y_i(\vec{w}^\top \vec{x}_i - b), \ \forall i. \tag{11.7}$$

The parameter C controls the trade-off: large C allows only small margin violations, small C allows larger violations.

11.2.3 KKT Conditions and Support Vectors

For the hard-margin SVM, the Lagrangian is:

$$L(\vec{w}, b, \vec{\lambda}) = \frac{1}{2} \|\vec{w}\|_2^2 + \sum_{i=1}^{n} \lambda_i (1 - y_i(\vec{w}^\top \vec{x}_i - b)). \tag{11.8}$$

From the KKT stationarity condition: $\vec{w}^\star = \sum_{i=1}^n \lambda_i^\star y_i \vec{x}_i$.

We say $(\vec{x}_i, y_i)$ is a *support vector* if $\lambda_i^\star > 0$. By complementary slackness:

- If $\lambda_i^\star = 0$: $(\vec{x}_i, y_i)$ does not contribute to the optimal solution.
- If $\lambda_i^\star > 0$: $y_i((\vec{w}^\star)^\top \vec{x}_i - b^\star) = 1$, so $\vec{x}_i$ is on the margin and contributes to the solution.

For soft-margin SVM with optimal $(\vec{w}^\star, b^\star, \vec{\xi}^\star, \vec{\lambda}^\star, \vec{\mu}^\star)$:

- If $\lambda_i^\star = 0$: $\xi_i^\star = 0$, point does not violate margin and does not contribute.
- If $\lambda_i^\star = C$: point is on or violates the margin and contributes.

- If $\lambda_i^\star \in (0, C)$: $\xi_i^\star = 0$ and point is exactly on the margin.

In general, support vectors contribute to the optimal solution and are on or violate the margin.

Bibliography

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] G. Calafiore and L. El Ghaoui. *Optimization Models*. Cambridge University Press, 2014.

[3] C. C. Pugh. *Real Mathematical Analysis*. Springer, 2002.

[4] P. Varaiya et al. Lecture notes on optimization. Unpublished manuscript, University of California, Department of Electrical Engineering and Computer Science, 1998.

[5] D. P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[6] G. Garrigos and R. M. Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

[7] Y. Nesterov et al. *Lectures on Convex Optimization*. Springer, 2018.