

ECE367: Matrix Algebra and Optimization

Reference Notes

Professor Frank R. Kschischang

Scribe: Pranshu Malik

Fall Semester, 2019

Contents

1	Optimization Problems in Standard Form	3
1.1	Linear Regression	4
1.2	Function Approximation	4
2	Mathematical Preliminaries	4
2.1	Sets	4
2.2	Functions	5
2.3	Fields	5
3	Vector Spaces	6
3.1	Subspaces	6
3.2	Affine Sets	6
4	Norms and Inner Products	6
4.1	Norms	6
4.2	Inner Products	6
5	Projection on Subspaces	6
5.1	Orthogonal Projection on One-dimensional Subspaces	6
5.2	Induced Norm	6
5.3	Metrics and Induced Metrics	6
5.4	Projection on Subspaces	6
5.5	Computing Projections with General Basis	6
5.6	Computing Projections with Orthonormal Basis	6
5.7	The Gram-Schmidt Procedure	6
5.8	Projection on the Orthogonal Complement	6
5.9	Projection on Affine Sets, including Lines and Hyperplanes	6
6	Linear Transformations	6
6.1	The Matrix of a Linear Transformation	6
6.2	The Fundamental Theorem of Linear Algebra	6
7	Operators	6
7.1	Invariant Subspaces	6
7.2	Upper Triangular Matrices for Operators on Complex Vector Spaces	6
7.2.1	Theorems for Operators	6
7.2.2	Change of Basis	6
7.2.3	Theorems for Matrices	6
7.3	Diagonal Matrices for Operators	6

8	Operators on Inner-Product Spaces	8
8.1	Upper Triangular Matrices	8
8.2	Linear Functionals and Adjoint	8
8.3	Operators on Inner Product Spaces	8
8.3.1	Self-Adjoint Operators	8
8.3.2	Normal Operators	8
8.4	The Spectral Theorem	8
8.4.1	Complex Inner-Product Spaces	8
8.4.2	Real Inner-Product Spaces	8
8.5	Positive Semidefinite Operators	8
8.6	Isometries	8
8.7	Polar Decomposition	8
9	The Singular Value Decomposition	8
9.1	Singular Values	8
9.2	Singular Value Decomposition for Operators	8
9.3	Singular Value Decomposition for Linear Transformations	8
9.4	Rayleigh Quotients	8
9.5	Projection Matrices	8
9.6	Least Squares and the Pseudoinverse	8
9.6.1	Problem Setup	8
9.6.2	Solution Set	8
9.6.3	The Moore-Penrose Pseudoinverse	8
9.6.4	Back to Least Squares	8
9.6.5	Coordinate View	8
10	Numerical Exercises	8
10.1	Exercise 0	8
10.2	Exercise 1	8
10.3	Exercise 2	8
10.4	Exercise 3	8
10.5	Exercise 4	8
10.6	Exercise 5	8
11	Acknowledgements	8

Note: This document is not meant to be a comprehensive treatment of the material. The primary purpose of this document is to summarise key concepts along with some analysis and proofs, that can aid in revising the course or serve as a quick reference to a particular topic. Any suggestions or corrections are appreciated and can be sent directly to pranshu.malik@mail.utoronto.ca

1 Optimization Problems in Standard Form

A typical optimization problem can be written in the following form:

Definition 1.1: Formulation of an optimization problem

For a vector of decision variables $\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ determine the optimal value,

$$p^* = \min_{\mathbf{x} \in \mathcal{D}} f_0(\mathbf{x})$$

subject to:

$$\begin{aligned} f_1(\mathbf{x}) &\leq 0 \\ f_2(\mathbf{x}) &\leq 0 \\ &\vdots \\ f_m(\mathbf{x}) &\leq 0 \end{aligned}$$

where $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ is the cost function and f_1, \dots, f_m are scalar functions representing constraints over the domain $\mathcal{D} \subseteq \mathbb{R}^n$.

We are often interested in the values $\mathbf{x}^* = [x_1^*, \dots, x_m^*]^\top$ that achieve the optimal value $p^* = f_0(\mathbf{x}^*)$, and can thus also write the optimization problem as:

Definition 1.2: Arg min/max

Subject to the given constraints,

$$\text{find: } \mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{D}} f_0(\mathbf{x}), \quad \text{where} \quad \arg \min_{\mathbf{x} \in \mathcal{D}} f_0(\mathbf{x}) := \{\mathbf{x} \in \mathcal{D} \mid \forall \mathbf{y} \in \mathcal{D}, f_0(\mathbf{y}) \geq f_0(\mathbf{x})\}$$

Note here that $\arg \max_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x})$ is also defined similarly.

Remark: If we want to instead maximize a function $g_0(\mathbf{x})$, we can set $f_0(\mathbf{x}) = -g_0(\mathbf{x})$. Then, we have $g_0 \left(\arg \max_{\mathbf{x} \in \mathcal{D}} g_0(\mathbf{x}) \right) = \max_{\mathbf{x} \in \mathcal{D}} g_0(\mathbf{x}) = - \left(\min_{\mathbf{x} \in \mathcal{D}} -g_0(\mathbf{x}) \right) = - \left(\min_{\mathbf{x} \in \mathcal{D}} f_0(\mathbf{x}) \right) = -f_0 \left(\arg \min_{\mathbf{x} \in \mathcal{D}} f_0(\mathbf{x}) \right)$.

Example: An oil refinery produces two products: jet fuel and gasoline. The profit per barrel is \$0.1 for jet fuel and \$0.2 for gasoline. As constraints, only 10,000 barrels of crude oil is available and the refinery has a contract to produce at least 1000 barrels of jet fuel and 2000 barrels of gasoline. Additionally, the refined products are shipped in tanker-trucks with a capacity of 180,000 barrels/km and the jet fuel is delivered 10 km away and gasoline is delivered 30 km from the refinery. What quantities of each product should be produced to maximize profit?

Let $\mathbf{x} = [x_1, x_2]^\top$ be the amounts of jet fuel and gasoline to be produced respectively. The profit is our objective function, $g(\mathbf{x}) = 0.1x_1 + 0.2x_2$, and the optimization problem becomes finding $\arg \max_{x_1, x_2 \in [0, \infty)} g(\mathbf{x})$ subject to the constraints $x_1 + x_2 \leq 10000$, $x_1 \geq 1000$, $x_2 \geq 2000$, $10x_1 + 30x_2 \leq 180000$.

Add graph here

Example—Knapsack Problem: Given a knapsack that can hold up to W weight in total, we need to select from a collection of N objects where the i^{th} object has weight w_i and value v_i , such that the net value of objects in the sack is maximized.

The problem requires N decision variables $x_1, \dots, x_n \in \{0, 1\}$ where $x_i = 0$ denotes exclusion and $x_i = 1$ denotes inclusion of the i^{th} object in the knapsack, respectively. The optimization problem is to maximize $x_1 v_1 + \dots + x_n v_n$, subject to $x_1 w_1 + \dots + x_n w_n \leq W$. We note here that this problem is an NP-Complete (link to proof).

1.1 Linear Regression

Given a collection of N data points $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$, we would like to find a straight line, $y = mx + b$, which gives the "best fit". We can define the error for the i^{th} data point, (x_i, y_i) , to be $\epsilon_i = y_i - (mx_i + b)$. We can define the "best fit" line as the choice of parameters m, b such that the total squared error is minimized.

$$\min_{m, b \in \mathbb{R}} f_0(m, b), \quad \text{where} \quad f_0(m, b) := \sum_{i=1}^N \epsilon_i^2 = \sum_{i=1}^N (y_i - (mx_i + b))^2$$

We chose to minimize the total squared error instead of the total absolute error, $\sum_i |\epsilon_i|$, because of its nicer mathematical properties such as global differentiability (derivative of absolute error is undefined at 0). This, among many other reasons, makes squared error more amenable to the techniques of mathematical optimization¹.

Add graph here

1.2 Function Approximation

Another example of a mathematical optimization problem is approximating differentiable functions using rectangular functions over discrete intervals. More specifically, given

$$\text{rect}(t) := \begin{cases} 1, & -\frac{1}{2} \leq t \leq \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

we define a set of all possible linear combinations of time-shifted rectangular functions over N intervals,

$$\Omega = \left\{ \sum_{i=0}^{N-1} a_i \text{rect}(t - i) \mid a_i \in \mathbb{R} \right\}.$$

Now, for any given signal $s(t)$, we need to find the element in Ω that "best" approximates $s(t)$ over the period $[-\frac{1}{2}, N - \frac{1}{2}]$. Here again, "best" can be based on least squared error and with a_0, \dots, a_{N-1} as decision variables, the optimization problem becomes finding

$$\arg \min_{a_0, \dots, a_{N-1} \in \mathbb{R}} \int_{-\frac{1}{2}}^{N-\frac{1}{2}} \left(\left(\sum_{i=0}^{N-1} a_i \text{rect}(t - i) \right) - s(t) \right)^2 dt$$

Add graph here

The numerical exercises involve setting up and solving a few variations of optimization problems numerically using Julia packages (covered in section 10).

2 Mathematical Preliminaries

2.1 Sets

We will often deal with sets of elements and a few key definitions are listed below.

¹<https://www.benkuhn.net/squared>

- A set might also be empty and is denoted as ϕ .
- A set with exactly one element, $\{x\}$, is called a *singleton* set.
- If A, B are sets, we write $A \subseteq B$ to assert that every element of A is also an elements of B .
- The Cartesian product of two sets A, B is defined as the set $A \times B := \{(a, b) \mid a \in A, b \in B\}$. This definition may similarly be generalized to a Cartesian product of any number of sets.
- We denote $\underbrace{A \times \dots \times A}_{n \text{ times}} = A^n$.

2.2 Functions

A function $f : \mathcal{D} \rightarrow \mathcal{C}$, from a set \mathcal{D} (*domain*) into a set \mathcal{C} (*co-domain*), is a rule that assigns exactly one element of \mathcal{C} to every element of \mathcal{D} . Some associated terms for such functions are listed below.

- Any argument, x , to the function, f , maps to $f(x)$ and is denoted as $x \mapsto f(x)$.
- For a set $A \subseteq \mathcal{D}$, the *image* of A under mapping by f is the set $f(A) := \{c \in \mathcal{C} \mid c = f(a) \text{ for some } a \in A\} = \{f(a) \mid a \in A\}$. Also note that, $f(\phi) = \phi$.
- $\mathcal{R} = f(\mathcal{D})$, the set of all elements of \mathcal{C} "hit" by f over the domain, is called the *range* of f . Note that $\mathcal{R} \subseteq \mathcal{C}$.
- For a set $B \subseteq \mathcal{C}$, the *inverse image* or *pre-image* of B under mapping by f is the set $f^{-1}(B) := \{d \in \mathcal{D} \mid f(d) \in B\}$. Note that $f^{-1}(B)$ may also be empty even though $B \neq \phi$.
- The function f is called *injective* or *one-to-one* if distinct elements of \mathcal{D} are mapped by f to distinct elements of \mathcal{C} , that is $d_1 \neq d_2 \implies f(d_1) \neq f(d_2)$.
- The function f is called *surjective* or *onto* if every element of \mathcal{C} is mapped to by f over \mathcal{D} .
- A function that it both injective and surjective is called *bijective*. Note that if $f : \mathcal{D} \rightarrow \mathcal{C}$ is a bijection, then the function $f^{-1} : \mathcal{C} \rightarrow \mathcal{D}$ exists, such that $f(d) = c \implies f^{-1}(c) = d \forall d \in \mathcal{D}, c \in \mathcal{C}$.
- A *binary operation* on a set A is a map $f : A^2 \rightarrow A$ taking pairs of elements of A to an element in A . For example, $+(a, b)$ (addition, usually written $a + b$) and $\cdot(a, b)$ (multiplication, usually written $a \cdot b$).
- A binary operation $*$ on set A is *associative* if $(a * b) * c = a * (b * c) \forall a, b, c \in A$.
- A binary operation $*$ on set A is *commutative* if $a * b = b * a \forall a, b \in A$. Note that matrix multiplication is non-commutative.
- A binary operation on set A has an *identity* if $\exists e \in A$ such that $a * e = e * a = a \forall a \in A$. The identity element e is unique, since if we suppose identity elements $e_1 \neq e_2$, then $e_1 = e_1 * e_2 = e_2$ which is a contradiction. As examples, 0 is the additive identity in \mathbb{Z} and I is the identity matrix under multiplication.
- If a binary operation $*$ on A has an identity element e , then an element $b \in A$ is called the inverse of an element $a \in A$ if $a * b = b * a = e$. Note that an element can have at most one inverse. As examples, every nonzero element $x \in \mathbb{R}$ has a multiplicative inverse $\frac{1}{x}$, while in \mathbb{Z} , only 1 and -1 have a multiplicative inverse. Also, every element $x \in \mathbb{R}$ has an additive inverse $-x$.

2.3 Fields

A field is a set \mathcal{F} equipped with two binary operations $+$ (addition) and \cdot (multiplication) such that:

1. $+$ is associative and commutative with identity element 0 and $\forall x, -x$ is its additive inverse
2. \cdot is associative and commutative with identity element 1 and $\forall x \neq 0, \frac{1}{x}$ is its multiplicative inverse
3. the *distributive law* holds, i.e., $\forall a, b, c \in \mathcal{F}, a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

The sets \mathbb{R}, \mathbb{C} form fields under the usual arithmetic operations, but the set \mathbb{Z} does not form a field. Fields are important as they provide scalars in a vector space.

3 Vector Spaces

Start here.

3.1 Subspaces

3.2 Affine Sets

4 Norms and Inner Products

4.1 Norms

4.2 Inner Products

5 Projection on Subspaces

5.1 Orthogonal Projection on One-dimensional Subspaces

5.2 Induced Norm

5.3 Metrics and Induced Metrics

5.4 Projection on Subspaces

5.5 Computing Projections with General Basis

5.6 Computing Projections with Orthonormal Basis

5.7 The Gram-Schmidt Procedure

5.8 Projection on the Orthogonal Complement

5.9 Projection on Affine Sets, including Lines and Hyperplanes

6 Linear Transformations

6.1 The Matrix of a Linear Transformation

6.2 The Fundamental Theorem of Linear Algebra

7 Operators

7.1 Invariant Subspaces

7.2 Upper Triangular Matrices for Operators on Complex Vector Spaces

7.2.1 Theorems for Operators

7.2.2 Change of Basis

7.2.3 Theorems for Matrices

7.3 Diagonal Matrices for Operators

Finish till here.

8 Operators on Inner-Product Spaces

8.1 Upper Triangular Matrices

8.2 Linear Functionals and Adjoint

8.3 Operators on Inner Product Spaces

8.3.1 Self-Adjoint Operators

8.3.2 Normal Operators

8.4 The Spectral Theorem

8.4.1 Complex Inner-Product Spaces

8.4.2 Real Inner-Product Spaces

8.5 Positive Semidefinite Operators

8.6 Isometries

8.7 Polar Decomposition

9 The Singular Value Decomposition

9.1 Singular Values

9.2 Singular Value Decomposition for Operators

9.3 Singular Value Decomposition for Linear Transformations

9.4 Rayleigh Quotients

9.5 Projection Matrices

9.6 Least Squares and the Pseudoinverse

9.6.1 Problem Setup

9.6.2 Solution Set

9.6.3 The Moore-Penrose Pseudoinverse

9.6.4 Back to Least Squares

9.6.5 Coordinate View

10 Numerical Exercises

10.1 Exercise 0

10.2 Exercise 1

10.3 Exercise 2

10.4 Exercise 3

10.5 Exercise 4

10.6 Exercise 5

11 Acknowledgements