Indian Statistical Institute
Bachelor of Statistical Data Science

Optimization and Numerical Methods

Lecture Notes & Course Material

Course Instructor: Dr. Kaushik Jana
Teaching Assistants: Mr. Adithya Vaidyanathan, Mr. Rathin Das and Mr. Gahin Maity

Spring 2026

Preface

This document is a comprehensive compilation of lecture notes prepared by Mr. Utkarsh Bharadwaj for the **Optimization and Numerical Methods** course. The course was taught over the Spring 2026 semester by **Dr. Kaushik Jana** at the **Indian Statistical Institute (ISI), Delhi Centre** and teaching assistant Mr. Adithya Vaidyanathan.

While these notes strive to capture the lectures accurately and provide a structured overview of the curriculum, they are meant to act as a supplementary material. To gain a deeper, more rigorous understanding of the mathematical foundations and to make the best use of this text, the audience is strongly advised to attend all the classes and tutorial sessions and read through the primary recommended textbooks:

1. **Convex Optimization**
   by Stephen Boyd and Lieven Vandenberghe
2. **Operations Research: An Introduction**
   by Hamdy A. Taha
3. **Elementary Numerical Analysis: An Algorithmic Approach**
   by S.D. Conte and Carl de Boor

We hope these notes serve as a valuable companion in your study of optimization techniques, linear programming, and numerical root-finding algorithms. Best of luck with your studies!

— Prepared by the BSDS Class of 2029

Contents

PART I

OPTIMIZATION

1 Mathematical Optimization

1.1 Standard Form and Definitions

Standard Form

Minimize $f_0(x)$ subject to $f_i(x) \leq b_i, \quad i = 1, \cdots, m.$

Components:

- **Optimization Variable:** $x = (x_1, x_2, \cdots, x_n)$
- **Objective Function:** $f_0 : \mathbb{R}^n \to \mathbb{R}$
- **Constraint Function:** $f_i : \mathbb{R}^n \to \mathbb{R} \qquad i = 1, \cdots, m$

Definition: The *Optimal Solution* x^* has the smallest value of f_0 among all vectors that satisfy the constraint.

1.2 Examples of Application

Example: Portfolio Optimization

- **Variable (x):** Amount invested in different assets.
- **Objective:** Minimizing Risk or Minimizing Return Variance.
- **Constraints:** Budget, Minimum Return Amount, Maximum Loss, maximum or minimum investment per asset.

Example: Device Sizing in Electronic Circuits

- **Variables:** Device widths and lengths.
- **Constraint:** Manufacturing limits, time constraint, maximum available area.
- **Objective:** Power Consumption.

Example: Data Fitting (Simple Linear Regression)

- **Variables:** Model Parameters (β_0, β_1).
- **Constraint:** Parameter limits.
- **Objective:** Predicting error or measuring misfit.

We minimize the sum of squared errors:

$$\sum \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Note

Many problems can be considered as an optimization problem! Our job is to identify the *type* of optimization problem.

1.3 Solving Optimization Problems

A general optimization problem is surprisingly hard to solve. The methods used generally involve long computation time and the solution may not always be achieved. However, we can solve a set of optimization problems using efficient and reliable methods.

```
# 1. Least Squares Problem
```

$$\text{minimize } ||Ax - b||_2^2$$

- **Analytical Solution:** $x^* = (A^T A)^{-1} A^T b$.
- **Properties:** Easy to recognize; standard techniques increase flexibility.
- **Computation Time:** $\approx n^2 k$ (where $A \in \mathbb{R}^{k \times n}$).

```
# 2. Linear Programming (LP)
```

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } a_i^T x \leq b_i, \qquad i = 1, \cdots, m \end{aligned}$$

where $c = (c_1, \ldots, c_k)^T$ and $x = (x_1, \ldots, x_k)^T$.

- **Solution:** No analytical formula, but reliable and efficient algorithms exist.
- **Difficulty:** Not as easy to recognize as least-squares.
- **Computation Time:** $\approx n^2 m$.

```
# 3. Convex Optimization
```

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq b_i, \qquad i = 1, \cdots, m \end{aligned}$$

Condition: The objective and constraint functions must be convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

if $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$.

Note on Convex Optimization

- Includes least-square and linear programming problems as special cases.
- Like LP, there is no analytical solution, but efficient numerical methods exist.

1.4 Example: Illumination Problem

Example:

We have m lamps and have to illuminate n small, flat patches. The intensity I_k at patch k depends linearly on lamp powers p_j:

$$I_k = \sum_{j=1}^{m} a_{kj} p_j \qquad a_{kj} = r_{kj}^{-2} \max\{\cos\theta_{kj}, 0\}.$$

We need to achieve the desired illumination of I_{des} with bounded lamp powers. We can formalize our problem into the following mathematical equation:

$$\begin{aligned} \text{minimize} \quad & \max_{k=1,\cdots,n} |\log I_k - \log I_{des}| \\ \text{subject to} \quad & 0 \leq p_j \leq p_{max}, \quad j = 1, \cdots, m \end{aligned}$$

Solutions discussed:

1. Using uniform power: $p_j = p$
2. Using least squares method:

   $$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{des})^2$$

   We can round-off p_j if $p_j > p_{max}$ or $p_j < 0$.
3. Using weighted least-squares:

   $$\text{minimize} \quad \sum_{k=1}^{n} (I_k - I_{des})^2 + \sum_{j=1}^{m} w_j (p_j - p_{max}/2)^2$$

   Here, we can iteratively adjust weights w_j until $0 \leq p_j \leq p_{max}$.
4. Using Linear Programming:

   $$\begin{aligned} \text{minimize} \quad & \max_{k=1,\cdots,n} |I_k - I_{des}| \\ \text{subject to} \quad & 0 \leq p_j \leq p_{max}, \quad j = 1, \cdots, m \end{aligned}$$

5. Using Convex Optimization:

$$\begin{aligned} \text{minimize} \quad & f_0(p) = \max_{k=1,\cdots,n} h(I_k/I_{des}) \\ \text{subject to} \quad & 0 \leq p_j \leq p_{max}, \quad j = 1, \cdots, m \end{aligned}$$

where $h(u) = \max\{u, 1/u\}$.

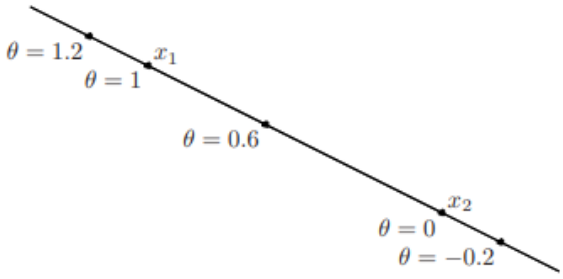
2 Affine Sets

2.1 Lines and Affine Combinations

Definition: Line

A line through two distinct points x_1 and x_2 can be given by:

$$\begin{aligned} x &= \theta x_1 + (1-\theta)x_2 \\ &= x_2 + \theta(x_1 - x_2) \end{aligned}$$



2.2 Observations on Lines

We can clearly observe the position of x based on θ:

- If $\theta = 0$, then $x = x_2$.
- If $\theta = 1$, then $x = x_1$.
- If $\theta > 1$, x lies beyond x_1 (outside the segment).
- If $\theta < 0$, x lies behind x_2 (outside the segment).

Note

· θ is the scale by which we are moving from x_2 to x_1, and $(x_1 - x_2)$ represents the direction of movement starting from x_2.
· The range of θ is $\mathbb{R}$.

2.3 Definition and Properties of Affine Sets

Definition: Affine Set

An affine set is a set which contains the **entire line** through any two distinct points in the set.
Formally, a set C is affine if for any $x_1, x_2 \in C$ and $\theta \in \mathbb{R}$:

$$\theta x_1 + (1-\theta)x_2 \in C$$

It means C contains the linear combination of any two points in C provided the coefficients in the linear combination sum to 1.

This idea can be generalized for more than two points as well. Consider a point of the form:

$$\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$$

where $\sum \theta_i = 1$. This is called an **affine combination** of the points $x_1, x_2, \cdots, x_k$.

Example: Affine Set as Subspace + Offset

If C is an affine set and a point $x_0 \in C$, then the set

$$V = C - x_0 = \{x - x_0 \mid x \in C\}$$

is a subspace. We know that a subspace is closed under addition and scalar multiplication.
Conclusion: Thus the affine set C can always be expressed as:

$$C = V + x_0 = \{v + x_0 \mid v \in V\}$$

which implies that the affine set C is a subspace plus an offset.

Example: Solution Set of Linear Equations

Solution Set of linear equations $C = \{x | Ax = b\}$ is an affine set.

Proof. Consider $x_1, x_2 \in C$, i.e., $Ax_1 = b$ and $Ax_2 = b$. Then, for any θ, consider the affine combination:

$$\begin{aligned} A(\theta x_1 + (1-\theta)x_2) &= \theta A x_1 + (1-\theta) A x_2 \\ &= \theta b + (1-\theta) b \\ &= b \end{aligned}$$

□

Homework: The subspace associated with the affine set C is the "null space of A".

3 Convex Sets

3.1 Definition of Convex Sets

Definition: Line segment

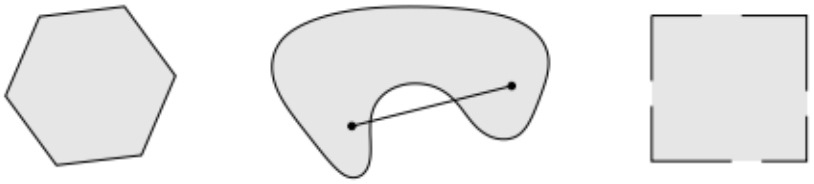
The line segment between x_1 and x_2 is given by

$$x = \theta x_1 + (1-\theta)x_2, \qquad 0 \leq \theta \leq 1$$

Definition: Convex Set

The convex set contains line segment between any two points in the set

$$x_1, x_2 \in C, 0 \leq \theta \leq 1 \implies \theta x_1 + (1-\theta)x_2 \in C$$



3.2 Examples and Properties

Example: Geometric Shapes

- A hexagon is a convex set.
- A kidney shaped structure is not a convex set.

Example: Singleton Set

$A = \{x\}, x \in \mathbb{R}^n$. Is A convex or Affine?
Solution: Since the set A is a singleton set, that means $x = y$. Hence, $\lambda x + (1-\lambda)x = x \in A \; \forall \lambda$. Therefore, A is an affine set and thus a convex set as well.

Example: Null Set

Show that the null set, ϕ is a convex set.
Solution: We need to show that $\forall x, y \in \phi,\ \lambda \in [0,1], \lambda x + (1-\lambda)y \in \phi$. Since there is no element in ϕ, we will not get a pair of elements x and y. Therefore, the statement of convexity of ϕ is vacuously true.

Example: Space $\mathbb{R}^n$

Show that the whole space $\mathbb{R}^n$ is an affine set.
Solution: We can consider $\mathbb{R}^n$ as a vector space. Since a vector space is closed under addition and scaler multiplication, the space $\mathbb{R}^n$ is closed under linear combination. Thus, it is an affine set.

3.3 Convex Combinations and Hulls

Convex Combination

A Convex combination of $x_1, x_2, \cdots, x_k$ is any point of the form

$$x = \theta_1 x_1 + \cdots \theta_k x_k$$

with $\sum \theta_i = 1$ and $0 \leq \theta_i \leq 1\ \forall i$.
This can be thought of as a mixture or weighted average of the points.

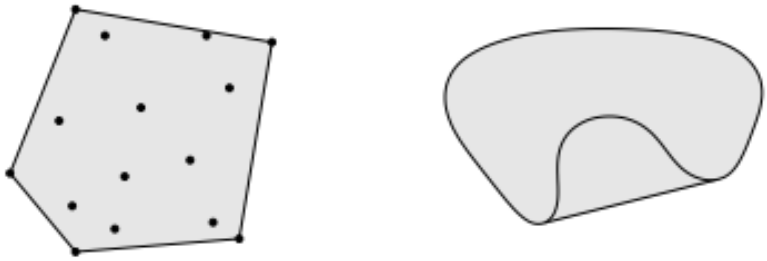
Convex Hull

A convex hull (S_1) of a set C, defined as

$$Conv\, C = \left\{\theta_1 x_1 + \cdots + \theta_k x_k | x_i \in C, \theta_i \geq 0, \sum \theta_i = 1\right\}$$

is the set of all convex combinations of points in S. It is the **smallest convex set** that contains C.

Example: Visualization: Convex Hull



4 Convex Cone

4.1 Cones and Convex Cones

Definition: Cone

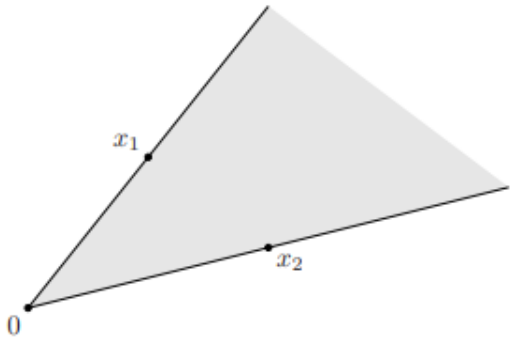
A set C is called a cone, if for every $x \in C$ and $\theta \geq 0$, we have $\theta x \in C$.

Convex Cone

A set C is a convex cone if it is convex and a cone. This means, for any two points, x_1, x_2:

$$x = \theta_1 x_1 + \theta_2 x_2 \qquad \theta_1, \theta_2 \geq 0$$

Then $x \in C$.



Example: Relation between Convex Cone and Convex Set

Show that a convex cone is a convex set.

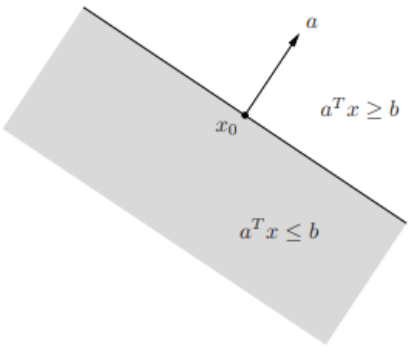
Proof. To be a cone: $\theta x \in C \; \forall \theta \geq 0$ and $x, y \in C \implies x + y \in C$.
To be convex: $\theta x + (1-\theta) y \in C \; \forall x, y \in C, \theta \in [0,1]$.
Now, take any two points, $x, y \in C, \theta \in [0,1]$. If C is a cone, then $\theta x \in C$ and $(1-\theta) y \in C$.
This implies $\theta x + (1-\theta) y \in C$. Hence, C is a convex set. □

5 Hyperplanes and Halfspaces

5.1 Hyperplanes

Definition: Hyperplane

It is a set of the form $\{x : a^T x = b\}, (a \neq 0,\ b \in \mathbf{R})$ is a hyperplane.



Example: Properties of Hyperplanes

1. Is a hyperplane an affine set? Yes. Take $x_1, x_2 \in D$, where D is a hyperplane. By definition, $a^T x_1 = b, a^T x_2 = b$. The affine combination yields $a^T x = (\theta_1 + \theta_2)b = b$.
2. Is a hyperplane a convex set? Yes. Proving affine automatically implies convex: $Affine \implies Convex$.
3. Dimension: The dimension of hyperplane in $\mathbb{R}^n$ is $(n-1)$.

5.2 Halfspaces

Definition: Half spaces

A hyperplane divides $\mathbf{R}^n$ into two halfspaces. A halfspace is a set of the form $\{x : a^T x \leq b\}, (a \neq 0,\ b \in \mathbf{R})$.

Example: Halfspaces

Affinity: A halfspace is **not** an affine space.

Proof. Since $\theta_1, \theta_2 \in \mathbb{R}$, it is not always true that $\theta_1 a^T x_1 \leq \theta_1 b \forall \theta_1 \in \mathbb{R}$. □

Convexity: A halfspace **is** a convex set.

6 Euclidean Balls and Ellipsoids

6.1 Euclidean Balls

Euclidean Ball

A ball (Euclidean) with center x_c and radius r:

$$B(x_c, r) = \{x : ||x - x_c||_2 \leq r\}$$

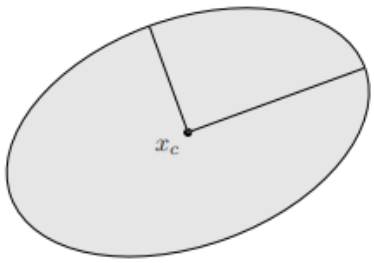
6.2 Ellipsoids

Ellipsoid

A set of the form

$$\{x : (x - x_c)^T P^{-1}(x - x_c) \leq 1\}$$

with $P \in S^n_{++}$ (i.e., P is a symmetric positive definite matrix).



Note

A positive definite matrix is defined as P if

$$x^T P x > 0 \forall x \neq 0$$

7 Norms and Polyhedra

7.1 Norms and Norm Balls

Norm

A function that satisfies:
- $||x|| \geq 0$ and $||x|| = 0 \iff x = 0$
- $||tx|| = |t| \cdot ||x||$ for $t \in \mathbb{R}$
- $||x + y|| \leq ||x|| + ||y||$

Example: Types of norm

Absolute norm, Euclidean Norm, Infinite Norm.

Norm Ball and Norm Cone

Norm Ball with center x_c and radius r:

$$\{x : ||x - x_c|| \leq r\}$$

Norm Cone:

$$\{(x, t) : ||x|| \leq t\}$$

Example: Homework

Show that Norm Balls and cones are convex sets.

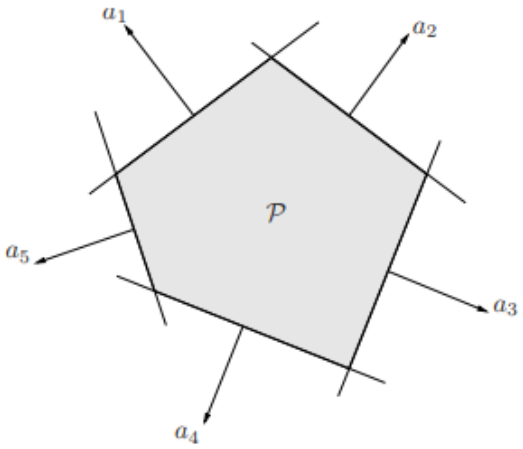
7.2 Polyhedra

Polyhedra

It is a solution set of finitely many linear inequalities and equalities:

$$Ax \preceq b, \quad Cx = d$$

where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^{n \times 1}$, $b \in \mathbb{R}^{m \times 1}$ and $C \in \mathbb{R}^{p \times n}$.



Note

Polyhedra is the intersection of finite number of half-spaces and hyperplanes.

Example: Homework

Show that a polyhedra is a convex set.

8 Convexity Preservation and Hyperplane Theorems

8.1 Convexity Preserving Operations

Operations that preserve convexity

1. **Definition of convexity:** $x_1, x_2 \in C$ and $0 \leq \theta \leq 1$, then $\theta x_1 + (1-\theta)x_2 \in C$.
2. The operations that preserve convexity are:
   - intersections of convex sets
   - affine functions

Affine function

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be an affine function if

$$f(x) = Ax + b$$

with $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$.

Note

- The image of a convex set under f is convex. So, consider the following:

$$S \subset \mathbb{R}^n \text{ is convex } \implies f(S) = \{f(x) \mid x \in S\} \text{ is convex}$$

- The inverse image $f^{-1}(C)$ of a convex set is convex:

$$C \subset \mathbb{R}^m \text{ is convex } \implies f^{-1}(C) = \{x \in \mathbb{R}^n \mid f(x) \in C\} \text{ is convex}$$

Note: The converse for both of these statements is not true.

Example: Affine function examples

- Scaling
- Translation
- Projection $(x_1, x_2) \in C \implies \{x_1 \mid (x_1, x_2) \in C\}$

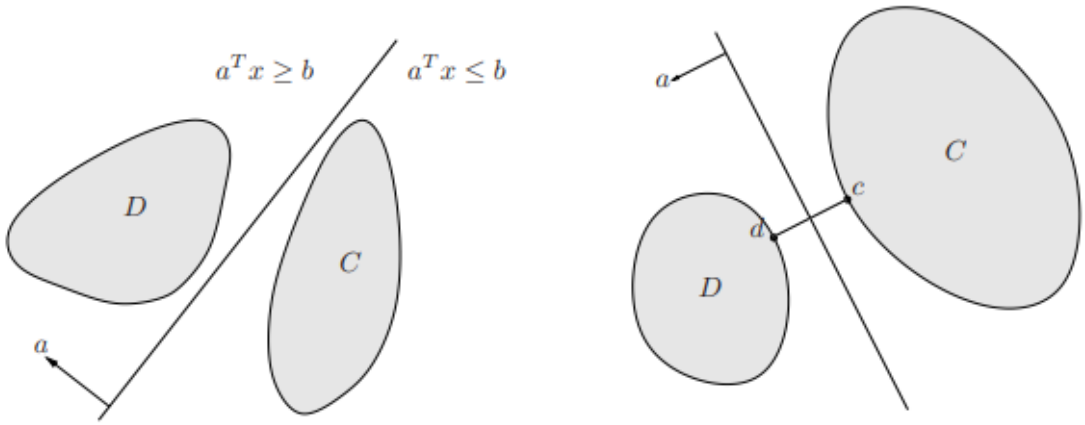
8.2 Separating and Supporting Hyperplanes

Separating Hyperplane Theorem

If C and D are non-empty disjoint convex sets, then there exists $a \neq 0$ such that:

$$a^T x \leq b \quad \forall x \in C, \quad a^T x \geq b \quad \forall x \in D$$

The hyperplane $\{x \mid a^T x = b\}$ separates C and D.
For any disjoint C, D, there has to be a separating hyperplane.



Example: Homework

Is disjointness necessary in the above theorem?

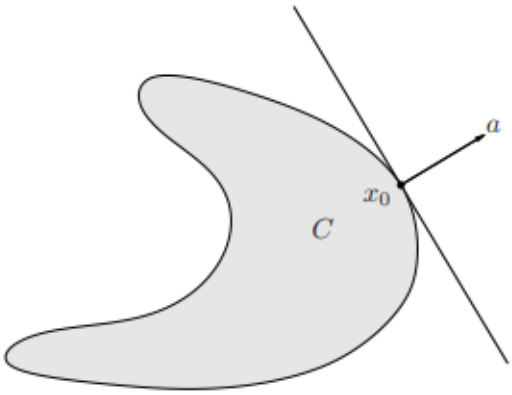
Supporting Hyperplane Theorem

Supporting hyperplane to set C at boundary point x_0:

$$\{x : a^T x = a^T x_0\}$$

where $a \neq 0$ and $a^T x \leq a^T x_0, \forall x \in C$.

Theorem: If C is convex, then there exists a supporting hyperplane at every boundary point of C.


a
x_0
C

9 Convex and Concave Functions

9.1 Definitions and Properties

Convex Function

$f : \mathbb{R}^n \to \mathbb{R}$ is a convex function if $dom(f)$ is a convex set and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \quad \forall x, y \in dom(f), 0 \leq \theta \leq 1.$$

Concave Function

$f : \mathbb{R}^n \to \mathbb{R}$ is a concave function if $dom(f)$ is a convex set and

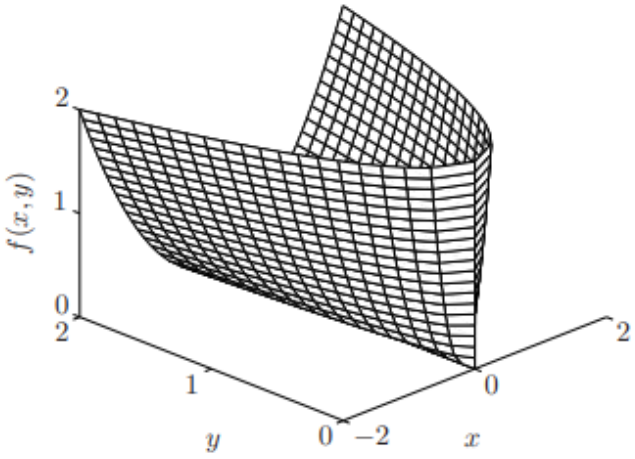
$$f(\theta x + (1-\theta)y) \geq \theta f(x) + (1-\theta)f(y) \quad \forall x, y \in dom(f), 0 \leq \theta \leq 1.$$

Properties

- f is concave if $-f$ is convex.
- f is strictly convex if $dom(f)$ is convex and

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y)$$

$\forall x, y \in dom(f), x \neq y, 0 < \theta < 1.$



Example: Examples in $\mathbb{R}$

1. **Affine Function:** $f(x) = ax + b$ on $\mathbb{R}$, for any $a, b \in \mathbb{R}$. Consider the following such that $x, y \in \mathbb{R}$:

$$\begin{aligned} f(\theta x + (1-\theta)y) &= a(\theta x + (1-\theta)y) + b \\ &= \theta ax + (1-\theta)ay + b \\ &= \theta ax + \theta b + (1-\theta)ay + (1-\theta)b \\ &= \theta(ax+b) + (1-\theta)(ay+b) \\ &= \theta f(x) + (1-\theta)f(y) \end{aligned}$$

   Therefore an affine function is a convex function.

2. **Exponential Function:** e^{ax} for any $a \in \mathbb{R}$. Consider the following such that $x, y \in \mathbb{R}$:

$$\begin{aligned} \exp(a(\theta x + (1-\theta)y)) &= \exp(a\theta x) \cdot \exp(a(1-\theta)y) \\ &= (e^{ax})^{\theta} \cdot (e^{ay})^{(1-\theta)} \\ &\leq \theta e^{ax} + (1-\theta)e^{ay} \\ &= \theta f(x) + (1-\theta)f(y) \end{aligned}$$

   This implies e^{ax} is a convex function.

> **Note**
>
> - The function is convex for any $a \in \mathbb{R}$.
> - If $a = 0$, it is a constant function.
> - If $a \neq 0$, it is strictly convex.

9.2 Sufficient Conditions for Convexity

Sufficient Condition for Checking Convexity

If f is twice differentiable, then $f''(x) \geq 0 \implies f$ is convex.

Example: Power Function

Check convexity of the function $f(x) = x^\alpha$ on $\mathbb{R}_+$ for $\alpha \geq 1, \alpha \leq 0$:

- For $\alpha \geq 1$, $f(x)$ is convex. We have:

$$f'(x) = \alpha x^{\alpha-1} \qquad f''(x) = \alpha(\alpha-1)x^{\alpha-2} \geq 0$$

- For $\alpha \leq 0$, $f(x)$ is convex.
- For $0 < \alpha < 1$, $f(x)$ is concave as $f''(x) < 0$.

Example: Homework: Check Convexity or Concavity

- $f(x) = |x|^p$ on $\mathbb{R}$, $p \geq 1$

  Solution: We can apply the definition of convex functions on $f(x)$ to check whether it satisfies it or not. Consider $x, y, \theta \in \mathbb{R}$. Thus we have:

$$\begin{aligned} f(\theta x + (1-\theta)y) &= |\theta x + (1-\theta)y|^p \\ &\leq \theta|x|^p + (1-\theta)|y|^p \qquad \text{(We need to check this statement)} \end{aligned}$$

- $f(x) = x \log x$ on $\mathbb{R}_+$

  Solution: For $x \in \mathbb{R}^+$, we have:

$$f'(x) = \log x + 1 > 0 \qquad f''(x) = \frac{1}{x} > 0$$

  Since $f''(x) > 0 \; \forall x \in \mathbb{R}^+$, we can conclude that $f(x) = x \log x$ is a *Convex Function*.

- $f(x) = \log x$ on $\mathbb{R}_+$

  Solution: For $x \in \mathbb{R}^+$, we have:

$$f'(x) = \frac{1}{x} > 0 \qquad f''(x) = -\frac{1}{x^2} < 0$$

  Since $f''(x) < 0 \; \forall x \in \mathbb{R}^+$, we can conclude that $f(x) = \log x$ is a *Concave Function*.

10 Conditions for Convexity

10.1 First-Order Conditions

Note

First Order Condition f is differentiable if $dom(f)$ is open and the gradient

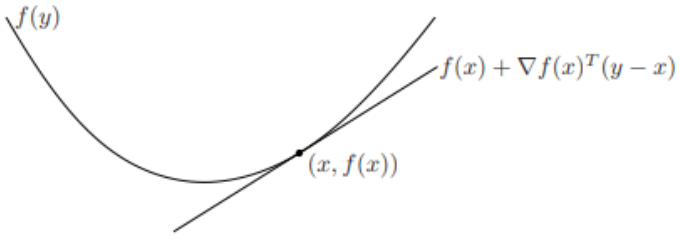
$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}\right)$$

exists at each $x \in dom(f)$.

1^{st} Order Condition for Convexity:

A differentiable function f with a convex domain is a convex function if and only if:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \forall x, y \in dom(f)$$



Note

Notes
- The first-order approximation of f is a global underestimate of f.
- This translates local information to global information.
- If $\nabla f(x) = 0 \implies f(y) \geq f(x)$. This means x is the global minimizer of f.

10.2 Second-Order Conditions

Note

f is twice differentiable if $dom(f)$ is open and the Hessian $\nabla^2 f(x) \in \mathbb{S}^n$ (set of symmetric matrices)

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \qquad i, j = 1, \ldots, n$$

exists at each $x \in dom(f)$.

Second Order Conditions

For a twice differentiable f with a convex domain, f is convex if and only if:

$$\nabla^2 f(x) \succeq 0 \qquad \forall x \in dom(f)$$

(i.e., the Hessian is positive semi-definite).

Note

Properties and Matrix Definiteness

- For a function in $\mathbb{R}$, $f''(x) \geq 0 \implies$ the derivative of f is non-decreasing.
- $\succeq 0$ means the matrix is *Positive Semi-definite* $\implies x^T A x \geq 0 \quad \forall x \neq 0$.
- If $\nabla^2 f(x) \succ 0 \quad \forall x \in dom(f)$, then f is strictly convex.
- If the determinants of all the leading principal minors are positive, then the corresponding matrix is a positive definite matrix.
- If the determinants of all principal minors are ≥ 0, then it is a positive semi-definite matrix.

Example: Examples

1. Check the convexity of $f(x) = \frac{1}{x^2}$ for all $x \in R \backslash \{0\}$
2. Least Squares Problem: Consider the least-squares objective function:

   $$f(x) = ||Ax - b||_2^2 = (Ax - b)^T (Ax - b)$$

   So, we have:

   $$\begin{aligned} \nabla f(x) &= \nabla (x^T A^T A x - b^T A x - x^T A^T b + b^T b) \\ &= \nabla (x^T A^T A x - 2 x^T A^T b + b^T b) \\ &= 2 A^T A x - 2 A^T b \end{aligned}$$

   Thus, taking the second derivative:

   $$\nabla^2 f(x) = 2 A^T A$$

   Since $2A^T A$ is a positive semi-definite matrix for any A, this implies that the objective function is convex.

11 Matrix Properties and Convexity

11.1 Matrix Definiteness and Sylvester's Criterion

Principal Minors

The determinant of the submatrix obtained by removing the same set of rows and columns.

Example: Finding Principal Minors

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

We can verify that $\begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$ is a principal minor, and $\begin{pmatrix} 1 & 3 \\ 7 & 9 \end{pmatrix}$ is not.

Leading Principal Minor

The determinant of the upper left $k \times k$ submatrix.

Sylvester's Criterion

Let A be an $n \times n$ symmetric matrix.

- A is positive definite (PD) if and only if all the leading principal minors are strictly positive.
- A is positive semi-definite (PSD) if and only if all principal minors are non-negative.

Why only symmetric matrices?

Let $A \in \mathbb{R}^{n \times n}$, then the symmetric part of A is

$$A_{sym} = \frac{1}{2}(A + A^T)$$

One can show that $x^T A x = x^T A_{sym} x \quad \forall x \in \mathbb{R}^n$

Solution:

$$\begin{aligned} x^T A_{sym} x &= x^T \left[\frac{1}{2}(A + A^T)\right] x \\ &= \frac{1}{2} x^T A x + \frac{1}{2} x^T A^T x \\ &= \frac{1}{2} x^T A x + \frac{1}{2} (x^T A x)^T \\ &= \frac{1}{2} x^T A x + \frac{1}{2} x^T A x = x^T A x \end{aligned}$$

Hence, we only use symmetric matrices.

Example: Checking for PD/PSD

1. $A = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$

   Leading Principal minors: 4, $det(A) = 8$. Since both are positive, hence A is PD as well as PSD.

2. $B = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$

   Leading Principal minors: 1, $det(B) = 0$. Since $det(B) = 0$, hence *B is not PD*.

   Principal Minors: 1, 4, $det(A) = 0$. Since all Principal Minors are non-negative, thus *B is PSD*.

Example: Exercise

1. Check whether the following matrices are PD/PSD:

   (a) $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

   (b) $B = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 5 & 6 \\ 3 & 6 & 9 \end{bmatrix}$

2. Let P be any $m \times n$ matrix with real entries and let $A = P^T P$. Show that A is positive semi-definite.

11.2 Domain Considerations in Convexity

Recap: 1^{st} and 2^{nd} order condition of convex function

1. The domain has to be a convex set.
2. The function should satisfy the conditions.

Example: Check for convexity

Check if $f(x) = \frac{1}{x^2}$ is a convex function on the domain $\mathbb{R} \setminus \{0\}$. Consider the following:

$$f'(x) = -\frac{2}{x^3} \qquad f''(x) = \frac{6}{x^4}$$

Thus the function is convex for $x \in \mathbb{R} \setminus \{0\}$. An important point to note is that the domain of $f = \{x \in \mathbb{R} \mid x \neq 0\}$ is *NOT a convex set*. Therefore $f(x) = \frac{1}{x^2}$ is not a convex function over the domain of f.

Reading Assignment

Read Chapter 3 from *Convex Optimization by Stephen Boyd* till Section 3.1.5.

12 Epigraphs, Sublevel Sets, and Inequalities

12.1 Epigraphs and Hypographs

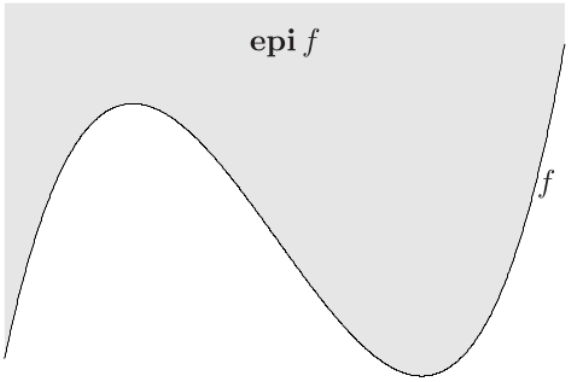
Epigraph

A graph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\{(x, f(x)) \mid x \in dom(f)\}$$

The epigraph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$epi\ f = \{(x, t) \mid x \in dom(f), f(x) \leq t\}$$



Note

- f is a convex function if and only if $epi\ f$ is a convex set.
- A function is concave if and only if its *hypograph* defined as

$$hypo\ f = \{(x, t) \mid t \leq f(x)\}$$

is a convex set.

12.2 Sublevel Sets

Sublevel Set

The α-sublevel set of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$C_\alpha = \{x \in dom\ f \mid f(x) \leq \alpha\}$$

> **Note**
>
> The sublevel set of a convex function is a convex set (but the converse is not true).

12.3 Jensen's Inequality

Jensen's Inequality

Basic Inequality: If f is a convex function, then for $0 \leq \theta \leq 1$ and $x, y \in \mathbb{R}$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

Extension: If f is convex, then

$$f(E[X]) \leq E[f(X)]$$

for any random variable X.

Proof. We know that $E(X) = \int xg(x)dx$ where $g(x)$ is the pdf of X. Consider the following:

$x_1, \ldots, x_k \in dom(f)$ and $\sum_{i=1}^{k} \theta_i = 1$, then from the basic inequality, we have

$$f(\theta_1 x_1 + \ldots + \theta_k x_k) \leq \theta_1 f(x_1) + \ldots + \theta_k f(x_k)$$

This inequality also holds if $g(x) \geq 0$, $\int g(x)dx = 1$. Consider $S \subset dom(f)$, we have

$$f\left(\int_S xg(x)dx\right) \leq \int_S (f(x)g(x))\,dx$$

provided the integral exists. Thus we have:

$$f(E[X]) \leq E[f(X)]$$

□

Implication of Jensen's Inequality

Suppose $x \in dom(f) \subset \mathbb{R}$ and Z is any zero mean random vector in $\mathbb{R}^n$ and f is a convex function. Then

$$\begin{aligned} E[f(x+Z)] \geq f(E[x+Z]) &= f(x+E[Z]) \\ &= f(x) \end{aligned}$$

Thus we have

$$E[f(x+Z)] \geq f(x) \quad \forall x \in dom(f)$$

This implies, if f is convex, the larger the variance, the larger the value of $E[f(x+Z)]$.

13 Operations that Preserve Convexity

13.1 Practical Methods for Establishing Convexity

Practical Methods for Establishing Convexity

There are several practical methods for establishing the convexity of a function:

1. Verify the definition: $f(\theta x+(1-\theta)y) \leq \theta f(x)+(1-\theta)f(y)$ where $x, y \in dom(f)$, $dom(f)$ is a convex set, and $\theta \in [0,1]$.
2. For twice differentiable functions, show that the Hessian is positive semi-definite:
$$\nabla^2 f(x) \succeq 0$$
3. Show that f is obtained from simple convex functions by operations that preserve convexity. Such operations include:
   - Non-negative weighted sum
   - Composition with an affine function
   - Pointwise maximum and supremum
   - Composition
   - Minimization

Homework: Non-negative Weighted Sum

Non-negative multiple: αf is convex if f is convex, and $\alpha \geq 0$. **Sum:** $f_1 + f_2$ is convex if f_1 and f_2 are convex.

Proof. Define $g = f_1 + f_2$. Consider $x, y \in dom(f_1) \cap dom(f_2)$ and $0 \leq \theta \leq 1$. (Proof to be completed). □

Note: This aggregation can be extended to an infinite sum and even to integrals.

13.2 Affine Composition and Pointwise Maximum

Composition with an Affine Function

$f(Ax + b)$ is convex if f is convex.

Proof. Let $f : \mathbb{R}^m \to \mathbb{R}$ be a convex function. Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Define $g : \mathbb{R}^n \to \mathbb{R}$ as $g(x) = f(Ax + b)$. We need to show that if f is convex, then g is also convex. Recalling the definition of convexity, we have: $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$ where $x, y \in dom(f)$, $dom(f)$ is a convex set, and $\theta \in [0,1]$. We then follow these steps:

1. $g(\theta x + (1-\theta)y) = f\left(A(\theta x + (1-\theta)y) + b\right)$

2. Use the linearity of the affine transformation. Consider the following:

$$A[\theta x + (1-\theta)y] + b = \theta Ax + (1-\theta)Ay + b$$

3. Add and subtract θb. We have the following:

$$\theta Ax + (1-\theta)Ay + \theta b + (1-\theta)b$$
$$= \theta(Ax+b) + (1-\theta)(Ay+b)$$

4. Apply the convexity of f:

$$f(\theta(Ax+b) + (1-\theta)(Ay+b))$$
$$\leq \theta f(Ax+b) + (1-\theta)f(Ay+b)$$

5. Use the definition of g:

$$g(\theta x + (1-\theta)y) \leq \theta g(x) + (1-\theta)g(y)$$

We can now conclude that g is a convex function and $dom(g) = \{x \in \mathbb{R}^n \mid Ax+b \in dom(f)\}$. □

Question: Is $dom(g)$ a convex set?
Yes. We are taking the inverse affine map of the domain of f (which is a convex set). The inverse image of a convex set under an affine mapping is convex. **Note:** Geometrically, an affine transformation will just move or linearly distort the convex function graph. That obviously won't change its convexity.

Example: Application of Affine Composition

Let $f(y) = y^2$, where $y \in \mathbb{R}$. Take $A = \begin{pmatrix} 2 & 3 \end{pmatrix}$ and $b = 1$. Consider the following:

$$g(x) = f(Ax+b)$$
$$= (Ax+b)^2$$
$$= \left(\begin{pmatrix} 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 1 \right)^2$$
$$= (2x_1 + 3x_2 + 1)^2$$

We can conclude that $g(x)$ is a convex function because it is the composition of the convex function $f(y) = y^2$ with an affine transformation. This property is extremely useful in convex optimization for the following reasons:

1. It allows us to transform complex problems into equivalent tractable problems through mapping.
2. In Machine Learning: Optimization of functions of the type $f(w^T x_i + b)$ are commonly observed. This is a convex function in w and b (if f is convex). Identifying convex functions helps to develop efficient algorithms.

Pointwise Maximum

If $f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex functions, then their pointwise maximum:

$$f(x) = \max\{f_1(x), \ldots, f_m(x)\}$$

is a convex function.

Example: Piecewise-Linear Function

A piecewise-linear function can be expressed as:

$$f(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i)$$

where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. Since each $a_i^T x + b_i$ is an affine (and therefore convex) function, their pointwise maximum is also convex.

14 Advanced Convexity Operations and Log-Concavity

14.1 Maximization and Minimization Operations

Example: 1: Distance to the Farthest Point

Distance to the farthest point in a set C:

$$f(x) = \sup_{y \in C} ||x - y||$$

Is $f(x)$ a convex function?

Note on Maximization and Minimization

1. **Maximum Eigenvalue of a Symmetric Matrix:** For a symmetric matrix $X \in \mathbb{S}^r$, the maximum eigenvalue:

$$\lambda_{\max}(X) = \sup_{||y||_2 = 1} y^T X y$$

   is a convex function.
2. **Minimization:** If $f(x, y)$ is convex in (x, y) and C is a convex set, then the infimum over C:

$$g(x) = \inf_{y \in C} f(x, y)$$

   is also a convex function.

Example: 2: Distance to a Set

Distance to a set: $dist(x, S) = \inf_{y \in S} ||x - y||$ is a convex function if S is a convex set.

14.2 Log-Concavity and Log-Convexity

Log-Concave and Log-Convex Functions

A positive function f is a **log-concave function** if $\log f$ is a concave function:

$$f(\theta x + (1 - \theta)y) \geq f(x)^\theta f(y)^{1-\theta} \quad \text{for } 0 \leq \theta \leq 1$$

Similarly, f is a **log-convex function** if $\log f$ is a convex function.

Example: 3: Examples of Log-Concavity

1. **Powers:** $f(x) = x^\alpha$ on $\mathbb{R}^+$ is log-convex for $\alpha \leq 0$ and log-concave for $\alpha \geq 0$.
2. Many common probability density functions are log-concave. For example, the multivariate normal distribution:

$$f(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}$$

14.3 Composition of Scalar Functions

Composition of Scalar Functions

Composition of $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$. Define $f(x) = h(g(x))$.
f is convex if:
- g is convex, and h is convex and monotonically non-decreasing.
- g is concave, and h is convex and monotonically non-increasing.

Proof. For $n = 1$, assuming g and h are twice differentiable:

$$f'(x) = h'(g(x)) \cdot g'(x)$$
$$f''(x) = h''(g(x))(g'(x))^2 + h'(g(x))g''(x)$$

By analyzing the signs of h'', h', and g'' based on the conditions above, we can guarantee that $f''(x) \geq 0$, ensuring convexity. □

Example: 4: Applying Composition Rules

1. Check the convexity of $\exp(g(x))$ if g is convex.
2. Check the convexity of $\frac{1}{g(x)}$ if g is concave and strictly positive.

Note on Hessian Eigenvalues

- All Eigenvalues of the Hessian Matrix ≥ 0 (Positive Semi-Definite) $\implies$ convexity.
- All Eigenvalues of the Hessian Matrix ≤ 0 (Negative Semi-Definite) $\implies$ concavity.

15 Optimization Problems in Standard Form

15.1 Problem Formulation and Terminology

Optimization Problem in Standard Form

An optimization problem in standard form is written as:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0 \quad \forall i = 1, \ldots, m \\ &\qquad\qquad h_i(x) = 0 \quad \forall i = 1, \ldots, p \end{aligned}$$

- $x \in \mathbb{R}^n$ is the optimization variable.
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective or cost function.
- $f_i : \mathbb{R}^n \to \mathbb{R}$ is the inequality constraint function.
- $h_i : \mathbb{R}^n \to \mathbb{R}$ is the equality constraint function.

Optimal Value (p^*)

Our goal is to find the optimal value. Define:

$$p^* = \inf\{f_0(x) \mid f_i(x) \leq 0 \; \forall i = 1, \ldots, m \text{ and } h_i(x) = 0, \; \forall i = 1, \ldots, p\}$$

We have the following special cases:

$$p^* = \begin{cases} \infty & \text{if the problem is infeasible (i.e., no } x \text{ satisfies the constraints)} \\ -\infty & \text{if the problem is unbounded below} \end{cases}$$

Optimal and Locally Optimal Points

- x is **feasible** if $x \in dom(f_0)$ and it satisfies all the constraints.
- A feasible x is **optimal** if $f_0(x) = p^*$.
- x is **locally optimal** if there exists $R > 0$ such that x is optimal for the following restricted problem:

$$\begin{aligned} &\text{minimize (over } z) \; f_0(z) \\ &\qquad \text{subject to } f_i(z) \leq 0, \quad i = 1, \ldots, m \\ &\qquad\qquad h_i(z) = 0 \quad \forall i = 1, \ldots, p \\ &\qquad\qquad ||z - x||_2 \leq R \end{aligned}$$

Example: Examples with $n = 1, m = p = 0$ (Unconstrained)

1. $f_0(x) = \frac{1}{x}, \quad dom(f_0) = \mathbb{R}_{++} \implies$ Clearly, $p^* = \inf f_0(x) = 0$ over $dom(f_0)$. There is no optimal point.
2. $f_0(x) = -\log x, \quad dom(f_0) = \mathbb{R}_{++} \implies p^* = -\infty$. There is no optimal point.
3. $f_0(x) = x \log x, \quad dom(f_0) = \mathbb{R}_{++} \implies$ Upon taking the derivative, we find $p^* = -\frac{1}{e}$ at the optimal point $x = \frac{1}{e}$.
4. $f_0(x) = x^3 - 3x \implies$ We can check that $p^* = -\infty$, but there is a local optimum at $x = 1$.

15.2 Implicit Constraints

Implicit Constraints

The standard form of an optimization problem has an implicit constraint:

$$x \in \mathcal{D} = \left\{ \bigcap_{i=0}^{m} dom(f_i) \right\} \bigcap \left\{ \bigcap_{i=1}^{p} dom(h_i) \right\}$$

- We call $\mathcal{D}$ the *domain* of the problem.
- The constraints $f_i(x) \leq 0$ and $h_i(x) = 0$ are the *explicit constraints*.
- A problem is *unconstrained* if it has no explicit constraints (i.e., $m = p = 0$).

Example: Unconstrained Problem Example

Let us consider the following problem:

$$\text{minimize } f_0(x) = -\sum_{i=1}^{k} \log\left(b_i - a_i^T x\right)$$

This is technically an unconstrained problem in standard form, but it has the implicit constraint $a_i^T x < b_i$ defined by the domain of the logarithm.

15.3 Convex Optimization Problems

Convex Optimization Problem

The standard form of a convex optimization problem is:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0 \quad \forall i = 1, \ldots, m \\ &\qquad\qquad a_i^T x = b_i \quad \forall i = 1, \ldots, p \end{aligned}$$

where $f_0, f_1, \ldots, f_m$ are convex functions and the equality constraints are affine. Alternatively, it can also be written in matrix form as:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0 \quad \forall i = 1, \ldots, m \\ &\qquad\qquad Ax = b \end{aligned}$$

Note

The feasible set of a convex optimization problem is always a convex set.

15.4 Global Optimality in Convex Problems

Locally and Globally Optimal

Theorem: Any locally optimal point of a convex optimization problem is also globally optimal.

Proof. Suppose x is locally optimal, but there exists a feasible y such that $f_0(y) < f_0(x)$. Since x is locally optimal, there is an $R > 0$ such that for any feasible z where $||z - x||_2 \leq R$, we have $f_0(z) \geq f_0(x)$. Now, consider the point:

$$z = \theta y + (1 - \theta)x \quad \text{with} \quad \theta = \frac{R}{2||y - x||_2}$$

Since we assume $||y - x||_2 > R$, it follows that $0 < \theta < \frac{1}{2}$. Here, z is a convex combination of two feasible points (x and y). Because the feasible set is convex, z is also feasible. Note that $||z - x||_2 = \frac{R}{2}$, which means z lies within the local optimality radius R. Furthermore, by the convexity of f_0:

$$\begin{aligned} f_0(z) &\leq \theta f_0(y) + (1 - \theta) f_0(x) \\ &= \theta(f_0(y) - f_0(x)) + f_0(x) \end{aligned}$$

Since we assumed $f_0(y) < f_0(x)$, the term $(f_0(y) - f_0(x))$ is negative. Therefore:

$$f_0(z) < f_0(x)$$

This contradicts the premise that x is locally optimal (since z is within radius R and has a strictly lower objective value). Hence, no such y can exist, and x must be globally optimal. □

15.5 Optimality Criterion

Optimality Criterion for Differentiable f_0

For a convex optimization problem, a feasible point x is optimal if and only if:

$$\nabla f_0(x)^T (y - x) \geq 0 \qquad \forall \text{ feasible } y$$

Geometrically, if $\nabla f_0(x) \neq 0$, this means that the vector $-\nabla f_0(x)$ defines a supporting hyperplane to the feasible set at the point x.

Example: Homework

Prove that $x^* = (1, 1/2, -1)^T$ is optimal for the optimization problem:

$$\text{minimize } \frac{1}{2} x^T P x + q^T x + r$$

$$\text{subject to } -1 \leq x_i \leq 1, \quad i = 1, 2, 3$$

where $r = 1$

$$P = \begin{pmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{pmatrix} \qquad q = \begin{pmatrix} -22 \\ -14.5 \\ 13 \end{pmatrix}$$

16 Applications of Linear Programming

16.1 Overview of Applications

Applications of Linear Programming

1. Investment Sector
2. Production Planning and inventory control
3. Workforce Planning
4. Transport Planning

16.2 Case Study: Investment Sector

Example: Investment Sector

The main goal is to maximize returns or minimize risk. Suppose a bank is in the process of devising a loan policy that involves a maximum of \$12 million. Now, there could be different types of loans such as:

Type of Loan	Interest Rate	Bad-Debt Ratio
Personal	0.14	0.10
Home	0.12	0.03
Car	0.13	0.07
Farm	0.125	0.05
Commercial	0.10	0.02

Competition with other financial organizations requires the imposition of constraints:

- Allocation of at least 40% to farm and commercial loans.
- Home loans must be equal to at least 50% of the personal, car, and home loans combined.
- The bank limits the bad-debt on all loans to at most 4%.

Mathematical Formulation:
First, we define the required decision variables (in millions of dollars) based on the table order:

- x_1 = Personal Loan
- x_2 = Home Loan
- x_3 = Car Loan
- x_4 = Farm Loan
- x_5 = Commercial Loan

Objective Function:
The bank wants to maximize its net return.

$$\text{Net Return} = \text{Total Interest} - \text{Lost Principal due to Bad-Debt}$$

$$\begin{aligned}\text{Total Interest} = {} & 0.14(1-0.10)x_1 + 0.12(1-0.03)x_2 + 0.13(1-0.07)x_3 \\ & + 0.125(1-0.05)x_4 + 0.10(1-0.02)x_5\end{aligned}$$

$$\text{Bad Debt} = 0.10x_1 + 0.03x_2 + 0.07x_3 + 0.05x_4 + 0.02x_5$$

Hence, we have the following Objective Function:

$$\text{Maximize } z = \text{Total Interest} - \text{Bad Debt}$$

We also have the following constraints:

- $x_1 + x_2 + x_3 + x_4 + x_5 \leq 12$
- $x_4 + x_5 \geq 0.40(x_1 + x_2 + x_3 + x_4 + x_5)$
- $x_2 \geq 0.50(x_1 + x_2 + x_3)$
- $0.10x_1 + 0.03x_2 + 0.07x_3 + 0.05x_4 + 0.02x_5 \leq 0.04(x_1 + x_2 + x_3 + x_4 + x_5)$
- $x_1, \ldots, x_5 \geq 0$

Now, we can solve for the optimum solution using R (`lpSolve` function).

17 Duality and The Lagrangian

17.1 Primal and Dual Formulations

The Dual Problem

Every linear programming problem has a second problem associated with it. One problem is called the **Primal Problem** and the other problem is called the **Dual Problem**. **Advantages:** They share closely related properties. The optimal solution of one problem yields complete information about the optimal solution to the other. There are two common forms of the primal problem:
1. Canonical Form
2. Standard Form

Canonical Form

$$\begin{aligned} \text{Maximize} \quad & x_0 = \sum_{j=1}^{n} c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_j \quad i = 1, \ldots, m \\ & x_j \geq 0 \quad j = 1, \ldots, n \end{aligned}$$

Note: Any LP problem can be put into this form.

Dual Problem Formulation

Let's write down the associated dual problem for the canonical primal problem above:

$$\begin{aligned} \text{Minimize} \quad & y_0 = \sum_{i=1}^{m} b_i y_i \\ \text{subject to} \quad & \sum_{i=1}^{m} a_{ij} y_j \geq c_j \quad j = 1, \ldots, n \\ & y_i \geq 0 \quad i = 1, \ldots, m \end{aligned}$$

where $y_1, \ldots, y_m$ are the dual variables.

1. Each constraint in one problem corresponds to a variable in the other problem.
2. Elements of the right-hand side of the constraints in one problem are equal to the respective coefficients of the objective function in the other problem.
3. A maximization problem becomes a minimization problem.
4. A maximization problem has a less-than-or-equal-to $(\leq)$ constraint which changes to a greater-than-or-equal-to $(\geq)$ constraint for the minimization problem.
5. The variables in both problems are non-negative.

Example: Converting Primal to Dual

Consider the following primal problem:

$$\begin{aligned} \text{Maximize} \quad & x_0 = 5x_1 + 6x_2 \\ \text{subject to} \quad & x_1 + 9x_2 \leq 60 \\ & 2x_1 + 3x_2 \leq 45 \\ & 5x_1 - 2x_2 \leq 20 \\ & x_2 \leq 30 \end{aligned}$$

Now we consider the equivalent dual problem:

$$\begin{aligned} \text{Minimize} \quad & y_0 = 60y_1 + 45y_2 + 20y_3 + 30y_4 \\ \text{subject to} \quad & y_1 + 2y_2 + 5y_3 \geq 5 \\ & 9y_1 + 3y_2 - 2y_3 + y_4 \geq 6 \\ & y_1, y_2, y_3, y_4 \geq 0 \end{aligned}$$

The underlying idea is that this dual problem may be easier to solve than the original primal problem.

Computational Advantages:

1. The dual problem has fewer constraints in this case (2 constraints vs. 4 constraints).
2. Computational difficulty in linear programming problems mainly depends on the number of constraints rather than the number of variables.
3. Since the optimal solution of one problem can be obtained from the optimal solution of the other, it is computationally more efficient to solve the dual problem in this case.

Reference: Similarly, the dual problem can be obtained from a primal problem in standard form. Please refer to Chapter 4 of H.A. Taha's book.

17.2 The Lagrangian and Dual Function

The Lagrangian

Reference: Convex Optimization by Stephen Boyd, Chapter 5, page 215.
Consider an optimization problem in the standard form:

$$\begin{aligned} \text{Minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{aligned}$$

with variable $x \in \mathbb{R}^n$. We assume its domain

$$\mathcal{D} = \left\{ \bigcap_{i=0}^{m} dom f_i \right\} \bigcap \left\{ \bigcap_{i=1}^{p} dom h_i \right\}$$

is non-empty and denote the optimal value of $f(x)$ by p^*. **Note:** We do not assume that the above problem is convex. The basic idea is that: The Lagrangian duality takes the constraints in the above problem into account by augmenting the objective function with a weighted sum of the constraint functions.

Lagrangian: It is a function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ associated with the above-mentioned problem, defined as:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

where:
- λ_i is the Lagrange multiplier associated with $f_i(x) \leq 0$.
- ν_i is the Lagrange multiplier associated with $h_i(x) = 0$.
- The vectors λ and ν are called the dual variables or Lagrange multiplier vectors.

The Lagrange Dual Function

The dual function $g(\lambda, \nu)$ is the infimum of the Lagrangian over x:

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu)$$

Expanding this, we have:

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)$$

When the Lagrangian is unbounded below in x, the dual function takes on the value $g(\lambda, \nu) = -\infty$.

Properties of the Dual Function

1. The dual function $g(\lambda, \nu)$ is concave, as it is the pointwise infimum of a family of affine functions of (λ, ν).
2. For any $\lambda \geq 0$ and any ν, it provides a lower bound on the optimal value:

$$g(\lambda, \nu) \leq p^*$$

Lower Bound on Optimal Value

The dual function yields lower bounds on the optimal value p^* of the original problem. Thus, for any $\lambda \geq 0$ and any ν, we have:

$$g(\lambda, \nu) \leq p^*$$

Proof. Suppose $\tilde{x}$ is a feasible point in the optimization problem. This means that $f_i(\tilde{x}) \leq 0$ and $h_i(\tilde{x}) = 0$. Thus, since $\lambda \geq 0$, we have:

$$\underbrace{\sum_{i=1}^{m} \lambda_i f_i(\tilde{x})}_{\leq 0} + \underbrace{\sum_{i=1}^{p} \nu_i h_i(\tilde{x})}_{=0} \leq 0$$

Therefore, evaluating the Lagrangian at $\tilde{x}$:

$$L(\tilde{x}, \lambda, \nu) = f_0(\tilde{x}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} \nu_i h_i(\tilde{x}) \leq f_0(\tilde{x})$$

Taking the infimum over all $x \in \mathcal{D}$, we naturally get a value less than or equal to the value at $\tilde{x}$:

$$\begin{aligned} g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) &\leq L(\tilde{x}, \lambda, \nu) \\ &\leq f_0(\tilde{x}) \end{aligned}$$

Since $g(\lambda, \nu) \leq f_0(\tilde{x})$ holds for every feasible point $\tilde{x}$, it must also hold for the optimal point x^* (where $f_0(x^*) = p^*$). This means that:

$$g(\lambda, \nu) \leq p^*$$

□

18 Optimality Conditions and KKT

18.1 Duality Gap and Strong Duality

Optimality Conditions

Recall the optimization problem with variable $x \in \mathbb{R}^n$:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \quad i = 1, \ldots, m \\ & h_i(x) = 0 \quad i = 1, \ldots, p \end{aligned}$$

We have the following domain of the problem:

$$\mathcal{D} = \left\{ \bigcap_{i=0}^{m} dom f_i \right\} \bigcap \left\{ \bigcap_{i=1}^{p} dom h_i \right\}$$

and denote the optimal value of the problem by p^*. **Note:** We do not have any assumption on the problem that it is convex. Now, consider a dual feasible solution (λ, ν) which gives us a lower bound on the optimal value of the primal problem. We thus have the following:

$$p^* \geq g(\lambda, \nu)$$

Here, (λ, ν) provide "a proof or a certificate" that $p^* \geq g(\lambda, \nu)$.

Strong Duality

This means that there exists an arbitrarily good certificate.

Duality Gap

The difference between primal and dual objectives is:

$$f_0(x) - g(\lambda, \nu)$$

which are associated with primal feasible point x and dual feasible point (λ, ν).

- If the duality gap is 0, i.e., $f_0(x) = g(\lambda, \nu)$, then x is primal optimal and (λ, ν) is dual optimal.

18.2 Karush-Kuhn-Tucker (KKT) Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Kindly refer to section 5.5.3 of Stephen Boyd's Book.
Consider the above discussed optimization problem. Add an assumption that the functions f_i and h_j are differentiable $\forall i = 1, \ldots, m$ and $j = 1, \ldots, p$ respectively. But no assumption about convexity is assumed.
Let x^* and (λ^*, ν^*) be any primal and dual optimal points respectively. Since x^* minimizes $L(x, \lambda^*, \nu^*)$ over x, it thus follows that its gradient must vanish at x^*. Thus, we have the following KKT Conditions:

$$
\begin{aligned}
f_i(x^*) &\leq 0 \quad i = 1, \ldots, m \\
h_i(x^*) &= 0 \quad i = 1, \ldots, p \\
\lambda_i^* &\geq 0 \quad i = 1, \ldots, m \\
\lambda_i^* f_i(x^*) &= 0 \quad i = 1, \ldots, m \quad \text{(Complementary Slackness)}
\end{aligned}
$$

$$
\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(x^*) = 0 \quad \text{(Stationarity)}
$$

For any optimization problem with differentiable objective and constraint functions for which strong duality obtains, any pair of primal and dual optimal points must satisfy the KKT conditions.

Weak and Strong Duality

Let d^* be the optimal value of the Lagrangian Dual Function. By definition, it is the best lower bound on p^*:

$$d^* \leq p^*$$

18.3 Optimization Examples

Example: Best Linear Unbiased Estimator (BLUE)

BLUE is a solution to a convex optimization problem *(Refer to Stephen Boyd's Book page 176, section 4.7)*. Suppose $y = Ax + V$ where:

- $V \in \mathbb{R}^m$ is measurement noise.
- $y \in \mathbb{R}^m$ is the vector of measurements.

- $x \in \mathbb{R}^n$ is a vector to be estimated.

Assumptions: A is a full rank matrix ($Rank = n$) and $E(V) = 0$. Also $E(VV^T) = I$, meaning zero mean and uncorrelated noise. We know that the linear estimator of x has the form $\hat{x} = Fy$. The estimator is called an unbiased estimator if for all x, we have $E(\hat{x}) = x$. Checking under what condition $E(\hat{x}) = x$:

$$\begin{aligned} E(\hat{x}) = E(Fy) &= E\left[F(Ax + V)\right] \\ &= FAE(x) + FE(V) \\ &= FAx \\ &= x \quad (\text{when } FA = I) \end{aligned}$$

To check whether it is the best unbiased estimator, we need to compute the error covariance of the estimator:

$$\begin{aligned} E[(\hat{x} - x)(\hat{x} - x)^T] &= E\left[(Fy - x)(Fy - x)^T\right] \\ &= E\left[FV(FV)^T\right] \\ &= E[FVV^TF^T] \\ &= FE(VV^T)F^T \\ &= FF^T \quad \left(\because E(VV^T) = I\right) \end{aligned}$$

Now, we have the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & tr(FF^T) \\ \text{subject to} \quad & FA = I \end{aligned}$$

with optimization variable $F \in \mathbb{R}^{m \times n}$. We have the following solution:

$$\boxed{F = (A^TA)^{-1}A^T}$$

Example: Optimization with Logistic Model

A random variable $Y \in \{0, 1\}$ satisfies:

$$Prob(Y = 1) = \frac{\exp(a^Tx + b)}{1 + \exp(a^Tx + b)}$$

where $x \in \mathbb{R}^n$ is the vector of variables that affects the probability of Y, and a and b are known parameters. **Case Example:** $Y = 1$ denotes the survival of a person and x is the vector of variables that affects the survival probability, e.g., BMI, age, income. The variable x, which we are to optimize over, is subjected to a set of linear constraints: $Fx \leq g$. **Questions:** Formulate the following problems as a convex optimization problem:

1. Maximize the survival probability where the goal is to choose x to maximize $P(Y = 1)$.

2. Maximize expected profit: Let $c^T x + d$ be the profit derived from selling the products. The goal is to maximize the expected profit, say $P(Y = 1)(c^T x + d)$.

19 Stochastic Optimization and Gradient Descent

19.1 Stochastic Linear Programming

Example: Revisiting the Company Insurance Problem

Reference: Convex Optimization by Stephen Boyd, page 210.

$$P(Y=1) = p = \frac{\exp(a^T x + b)}{1 + \exp(a^T x + b)}$$

Problem 1: Maximize the above probability subject to $Fx \leq g$. **Solution:** Consider the function $f(u) = \frac{e^u}{1+e^u}$. Note that $f(u)$ is a monotonically increasing function (which can be verified by taking the derivative). Thus, maximizing $f(u)$ is equivalent to maximizing u. Therefore, we have:

$$\text{maximize } \left(\frac{\exp(a^T x + b)}{1 + \exp(a^T x + b)}\right) \equiv \text{maximize } (a^T x + b)$$

This reduces to a simple linear programming problem:

$$\begin{aligned} &\text{maximize } a^T x + b \\ &\text{subject to } Fx \leq g \end{aligned}$$

Problem 2: Maximize expected profit where $c^T x + d$ is the profit derived from selling the product. Thus we have the expected profit:

$$E[c^T x + d] = \frac{\exp(a^T x + b)}{1 + \exp(a^T x + b)} \times (c^T x + d)$$

> **Hometask**
>
> Formulate the problem of maximizing $E[c^T x + d]$ subject to $Fx \leq g$ as a convex optimization problem, if possible.

Example: Linear Programming with Random Components

Consider an LP with variable $x \in \mathbb{R}^n$:

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Gx \leq h \\ &\qquad\qquad Ax = b \end{aligned}$$

Suppose the cost vector $c \in \mathbb{R}^n$ is random (and assume that other problem parameters

are deterministic). Assume the mean value of c is $\bar{c} = E[c]$ and the Covariance matrix is $\Sigma = E\left[(c-\bar{c})(c-\bar{c})^T\right]$. For a given $x \in \mathbb{R}^n$, the cost function will be $c^T x$, which is a scalar random variable. Therefore, we have:

$$\text{Mean cost: } E[c^T x] = \bar{c}^T x$$

and the variance of the cost:

$$Var(c^T x) = E\left[\left(c^T x - E[c^T x]\right)\left(c^T x - E[c^T x]\right)^T\right] = x^T \Sigma x$$

There is a trade-off between a small expected cost and a small cost variance. One way to address this is to minimize a linear combination of the expected cost and the variance of the cost:

$$E[c^T x] + \gamma Var(c^T x)$$

This is called a **risk-sensitive cost**, where $\gamma > 0$ is the risk aversion parameter since it sets the relative weights of cost variance and expected cost. To minimize the risk-sensitive cost, we solve the following Quadratic Programming (QP) problem:

$$\begin{array}{ll} \text{minimize} & \bar{c}^T x + \gamma x^T \Sigma x \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

Reference: Example 4.8, page 158 of Stephen Boyd's Book.

19.2 Gradient Descent Method

Gradient Descent Method

Gradient Descent is an optimization algorithm used to find the local minimum of a differentiable function. Let us start with an example of minimization problem.
Least Square: Suppose we want to predict an output y from an input x. Let us define a hypothetical linear model:

$$h_\beta(x) = \beta_0 + \beta_1 x$$

where:
- $h_\beta(x)$ is our model prediction.
- β_0, β_1 are the parameters to estimate.

Finding the "best fit" means finding the best estimates for β_0 and β_1. We define the following cost function (Mean Squared Error):

$$J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\beta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

where m is the size of the training sample and $\left(h_\beta\left(x^{(i)}\right) - y^{(i)}\right)$ is the error of a single point.
Note: We take squared error so that positive and negative errors don't cancel out, and because sqaure function a mathematically convenient (differentiable and convex) function.

Gradient Descent Update Rule

Gradient descent method is an iterative method which includes the following steps to calculate minimum of a differentiable function
Flowchart: Initialize variables $\rightarrow$ Calculate Gradient $\rightarrow$ Update Variables $\rightarrow$ Repeat until convergence.

- Start with some initial guess $\beta_0^{(0)}$ and $\beta_1^{(0)}$
- Calculate gradient: For our specific least square cost function, we can compute the partial derivatives:

$$\frac{\partial J}{\partial \beta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\beta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$\frac{\partial J}{\partial \beta_1} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\beta\left(x^{(i)}\right) - y^{(i)}\right) x^{(i)}$$

- The general update rule for any parameter β_j is:

$$\beta^{(j+1)} := \beta^{(j)} - \alpha \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$$

where α is the learning rate (step size, given before hand) where $j = 1, \ldots, n$, and n is total number of iterations.

- Repeat the following updates simultaneously until convergence:

$$\beta_0 := \beta_0 - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\beta \left(x^{(i)} \right) - y^{(i)} \right) \right]$$

$$\beta_1 := \beta_1 - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\beta \left(x^{(i)} \right) - y^{(i)} \right) x^{(i)} \right]$$

- Stop when the convergence happen, that is when, $|\beta^{(j+1)} - \beta^{(j)}| \leq \epsilon$, for a given small value of ϵ.

Reference: Chapter 9 of Stephen Boyd's Book for unconstrained minimization.

Example: Implementation of Gradient Descent in R (Simple Linear Regression)

The following script demonstrates the Gradient Descent algorithm to estimate the parameters (β_0 and β_1) for a Simple Linear Regression model. **Methodology Overview:**

1. **Data Generation:** We simulate a dataset where the true relationship is $y = 3 + 2x + \epsilon$ (where ϵ is random noise).
2. **Cost Function:** We define the Mean Squared Error (MSE) to evaluate how far off our predictions are from the actual values.
3. **Gradient Descent Loop:** We initialize our parameters at 0. Over a set number of iterations (epochs), we compute the gradients of the MSE with respect to β_0 and β_1, and subtract a fraction of this gradient (determined by the learning rate α) to update the parameters.
4. **Evaluation:** We track the cost at each epoch to ensure the algorithm is converging properly, and finally compare our iterative results with the exact analytical solution provided by Ordinary Least Squares (OLS).

```r
# Step 1: Generate Data from a model
experience <- c(1,2,3,4,5,6,7,8,9,10)
salary <- 3 + 2*experience + rnorm(10, 1, 5) # True: beta0=3, beta1=2
df <- data.frame(experience, salary)

# Step 2: Define the Cost Function (Mean Squared Error)
cost_function <- function(y, y_pred) {
  mean((y - y_pred)^2)
}

```

```r
# Step 3: Gradient Descent Algorithm Setup
x <- df$experience
y <- df$salary
n <- length(y)

# Initialize parameters
beta0 <- 0
beta1 <- 0
alpha <- 0.01    # Learning rate
epochs <- 100    # Number of iterations
cost_history <- numeric(epochs)

# Step 4: Iterative Update Loop
for (i in 1:epochs) {
  y_pred <- beta0 + beta1 * x

  # Calculate
  gradients
  d_beta0 <- (-2/n) * sum(y - y_pred)
  d_beta1 <- (-2/n) * sum((y - y_pred) * x)

  # Update parameters
  beta0 <- beta0 - alpha * d_beta0
  beta1 <- beta1 - alpha * d_beta1

  # Record cost
  cost_history[i] <- cost_function(y, y_pred)
}

# Step 5: Output Results and Compare with Least Squares
cat("GD Intercept:", beta0, "\n")
cat("GD Slope:", beta1, "\n")

model <- lm(salary ~ experience, data=df)
cat("OLS Estimates:\n")
coef(model)
```

Note

Algorithm Convergence & Diagnostics By plotting the `cost_history` vector against the number of iterations, we can visually diagnose the learning process.

- **Is the algorithm converging?** If the curve drops sharply and then flattens out (asymptotes), the algorithm is converging successfully.
- **What does the curve tell us?** It tells us if our learning rate α is appropriate. If the cost oscillates or explodes upwards, α is too high. If it decreases too slowly, α is too low, or we need more epochs.

Home Tasks: Experimenting with GD

Task 1: Variance Comparison
Generate the data 50 times (Monte Carlo simulation) and calculate the estimates using both the Gradient Descent (GD) method and the Least Squares (LS) method. Compare the variances of the two types of estimates.
Task 2: Impact of Initial Guesses
Change the initial guess (e.g., set $\beta_0 = 1, \beta_1 = 1$) and compute the estimates again. Observe the changes in the final estimates and their convergence speed. How does the initial starting point affect the trajectory of the algorithm on a convex surface?

Example:

1. **Logistic Regression:**

   Here the response variable Y_i takes two possible values (say, 0 and 1) and the covariates x_i can take any values. Unlike linear regression, there is **no closed-form solution** for the maximum likelihood estimates of the parameters of logistic regression.

   Logistic regression function of Y on x:

$$p_i = P(Y_i = 1) = \frac{1}{1 + e^{-x_i^\top \beta}}$$

   Suppose we have binary outcomes (response variable)

$$y_i \in \{0, 1\}, \quad i = 1, \ldots, n$$

   and predictor variable x_i.

   The logistic regression model assumes

$$p_i = P(y_i = 1 \mid x_i) = \frac{1}{1 + e^{-x_i^\top \beta}}$$

   Since y_i follows a Bernoulli distribution, the likelihood function for the i-th observation is

$$P(y_i \mid x_i, \beta) = p_i^{y_i}(1 - p_i)^{1-y_i} \tag{1}$$

   The joint density function (likelihood function) of n observations, assuming observations are independent, leads to

$$L(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}$$

   Substituting the logistic form of p_i,

$$L(\beta) = \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-x_i^\top \beta}}\right)^{y_i} \left(1 - \frac{1}{1 + e^{-x_i^\top \beta}}\right)^{1-y_i}.$$

The log-likelihood is

$$\ell(\beta) = \log L(\beta) = \sum_{i=1}^{n} \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right].$$

Using

$$p_i = \frac{e^{x_i^\top \beta}}{1 + e^{x_i^\top \beta}},$$

we obtain

$$\ell(\beta) = \sum_{i=1}^{n} \left[y_i x_i^\top \beta - \log\left(1 + e^{x_i^\top \beta}\right) \right]. \tag{2}$$

Since obtaining a closed-form expression for the MLE of the unknown parameter β is difficult in this case, we can use the gradient descent method to compute the MLE numerically. The algorithm for the computation is given below:

- **Step 1: Define the model**
  As defined in equation (1).
- **Step 2: Derive the objective function**
  Derive the negative log-likelihood function $J(\beta) = -\ell(\beta)$.
- **Step 3: Initialize the Parameter**
  Choose an initial value for the parameter vector:
  $$\beta^{(0)} = \mathbf{0}.$$
- **Step 4: Compute the Gradient**
  The gradient of the negative log-likelihood function is
  $$\nabla J(\beta^{(t)}) = -\sum_{i=1}^{n} \left(y_i - \frac{\exp(x_i^\top \beta^{(t)})}{1 + \exp(x_i^\top \beta^{(t)})} \right) x_i$$
- **Step 5: Update the Parameter**
  Update the parameter vector using the gradient descent rule:
  $$\beta^{(t+1)} = \beta^{(t)} - \alpha \nabla J(\beta^{(t)}),$$
  where $\alpha > 0$ is the learning rate.
- **Step 6: Check Convergence**
  Repeat Steps 4–5 until convergence, that is,
  $$\left\| \beta^{(t+1)} - \beta^{(t)} \right\| < \varepsilon,$$
  or equivalently,
  $$|J(\beta^{(t+1)}) - J(\beta^{(t)})| < \varepsilon,$$
  for some small $\varepsilon > 0$.
- **Step 7: Obtain the MLE**
  After convergence, the estimator is
  $$\hat{\beta}_{MLE} = \beta^{(t+1)}.$$

PART II

NUMERICAL METHODS

20 Numerical Methods: Root Finding

20.1 Motivation and Overview

Example: Motivation for Numerical Methods

Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous function.
Aim 1: To find roots of $f(x)$, i.e., solutions to $f(x) = 0$. Example: Compute roots of $f(x) = e^x + x - 5$.
Aim 2: Compute the area under the curve of $f(x)$, i.e., $\int_a^b f(x)dx$.
Example: Let $X \sim N(0, 1)$, compute $P(a \leq X \leq b) = \int_a^b f(x)dx$.
Aim 3: Suppose we have a finite set of data points $\{(x_i, y_i)_{1 \leq i \leq n}\}$.
Fit a polynomial curve passing through all data points (Interpolation).

We will be learning the following topics:

1. Root Finding Methods
2. Polynomial Interpolation
3. Numerical Integration

20.2 Root Finding Methods

Reference: "Numerical Methods: An Algorithmic Approach" by Conte and de Boor.

Intermediate Value Theorem (IVT)

One of the core results we will use is the Intermediate Value Theorem.
Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous function and let $a, b \in \mathbb{R}$, $a < b$. If $f(a)f(b) < 0$, then $f(x)$ has at least one root in the interval (a, b).

20.3 Bisection Method

Core Principle

Given a continuous function $f : \mathbb{R} \to \mathbb{R}$ and an initial interval $[a_0, b_0]$ with $f(a_0)f(b_0) < 0$, we halve the interval and search the half in which the root occurs, iteratively.

Algorithm

For $n = 0, 1, 2, \ldots$ until satisfied, do the following steps:
1. Set $m = \frac{a_n + b_n}{2}$.
2. If $f(m) = 0$, exit (exact root found).
3. If $f(a_n)f(m) < 0$, set $a_{n+1} = a_n$ and $b_{n+1} = m$. Otherwise, set $a_{n+1} = m$ and $b_{n+1} = b_n$.

Convergence of Bisection Method (Theorem)

Let $f(x)$ be a continuous function on $[a_0, b_0]$ such that $f(a_0)f(b_0) < 0$. Let $\{[a_n, b_n]\}_{n\geq 0}$ be the sequence of intervals generated by the bisection method and let $x_n = \frac{a_n + b_n}{2}$ be the n^{th} estimate. Then:

(a) There exists a point $x^* \in \bigcap_{n=0}^{\infty}[a_n, b_n]$ such that $f(x^*) = 0$.

(b) The absolute error is bounded by $|x_n - x^*| \leq \left(\frac{b_0 - a_0}{2^{n+1}}\right) \forall n \geq 0$, hence $\lim_{n\to\infty} x_n = x^*$.

Proof. **For (a):** By Cantor's Intersection Theorem (or Nested Interval Theorem), we get that the intersection $\bigcap_{n=0}^{\infty}[a_n, b_n]$ is non-empty. Since the length of the interval halves each step:

$$\lim_{n\to\infty}(b_n - a_n) = \lim_{n\to\infty}\frac{(b_0 - a_0)}{2^n} = 0$$

We thus get that $\bigcap_{n=0}^{\infty}[a_n, b_n]$ contains exactly one point, i.e., x^*. Now, we have $a_n \leq x^* \leq b_n \quad \forall n \geq 0$. Hence $\lim_{n\to\infty} a_n = x^*$ and $\lim_{n\to\infty} b_n = x^*$ as the boundaries a_n and b_n both squeeze towards x^*. Since f is continuous:

$$f(x^*) = \lim_{n\to\infty} f(a_n) = \lim_{n\to\infty} f(b_n)$$

Given $f(a_0)f(b_0) < 0$, assume $f(a_0) < 0$ and $f(b_0) > 0$. Now $x_1 = \frac{a_0 + b_0}{2}$ and:

$$a_1 = \begin{cases} a_0 & \text{if } f(x_1) > 0 \\ x_1 & \text{if } f(x_1) < 0 \end{cases}$$

In both cases, $f(a_1) < 0$. Inductively, we have that $f(a_n) < 0 \quad \forall n \geq 0$ and $f(b_n) > 0 \quad \forall n \geq 0$. This implies:

$$\lim_{n\to\infty} f(a_n) \leq 0 \qquad \text{and} \qquad \lim_{n\to\infty} f(b_n) \geq 0 \implies f(x^*) = 0$$

□

Proof. **For (b):** We have $x^* \in [a_n, b_n]$ and x_n is the midpoint of $[a_n, b_n]$. Hence the distance from the midpoint to any point in the interval is at most half the interval's length:

$$|x_n - x^*| \leq \frac{1}{2}(b_n - a_n) = \frac{1}{2}\frac{(b_0 - a_0)}{2^n} = \frac{(b_0 - a_0)}{2^{n+1}}$$

This implies:

$$|x_n - x^*| \leq \frac{(b_0 - a_0)}{2^{n+1}} \quad \forall n \geq 0$$

Hence, $x_n \to x^*$ as $n \to \infty$. □

Example: Homework Exercise

Show that $|x_{n+1} - x_n| = \frac{b_0 - a_0}{2^{n+2}} \quad \forall n \geq 0$.

Accuracy in Bisection Method

We say that an estimate x_n is "accurate to d decimal places" if $|x_n - x^*| \leq 0.5 \times 10^{-d}$. Now, for the bisection method, we know that $|x_n - x^*| \leq \frac{M}{2^{n+1}}$, where $M = b_0 - a_0$. Thus, if $\frac{M}{2^{n+1}} \leq 0.5 \times 10^{-d}$, then x_n is accurate to d decimal places. This implies:

$$n > \frac{d + \log_{10} M}{\log_{10} 2}$$

Solving this inequality gives the minimum number of iterations n required to achieve d decimal places of accuracy.

Example: Finding a Root using Bisection

Find the root of $f(x) = x^3 - x - 1$ in the interval $[1, 2]$ accurate to one decimal place, using the bisection method. **Solution:** Firstly, $f(1) = -1 < 0$ and $f(2) = 5 > 0$. So, by the Intermediate Value Theorem, there exists a root in $[1, 2]$. For one decimal place ($d = 1$), from our formula, we get:

$$n > \frac{(1 + \log_{10}(2 - 1))}{\log_{10} 2} = \frac{1}{0.301} \approx 3.3$$

So we need $n = 4$ iterations.

Iterations (n)	a_n	b_n	x_n	$f(x_n)$
0	1	2	1.5	0.875
1	1	1.5	1.25	-0.2969
2	1.25	1.5	1.375	0.2246
3	1.25	1.375	1.3125	-0.0515
4	1.3125	1.375	1.34375	0.0826

Remark: Note that $f(x_3)$ is closer to 0 than $f(x_4)$. The drawback of the bisection method is that it bases its updates only on the sign of the function, not the magnitude, which means the absolute error is not strictly monotonically decreasing at every single step, even though the interval boundary strictly decreases.

Example: Application: Finding the Median of a Cauchy Distribution

Suppose we are analyzing the location parameter μ of a Cauchy distribution. We want to find the median of the distribution. We have the following Probability Density Function (PDF):

$$f(x;\mu,\sigma) = \frac{1}{\pi\sigma}\left[\frac{1}{1+\left(\frac{x-\mu}{\sigma}\right)^2}\right]$$

We also have the following Cumulative Distribution Function (CDF):

$$F(x;\mu,\sigma) = \int_{-\infty}^{x} f(t;\mu,\sigma)dt = \frac{1}{\pi}\tan^{-1}\left(\frac{x-\mu}{\sigma}\right) + \frac{1}{2}$$

To find the median, we set $F(x;\mu,\sigma) = 0.5$:

$$\frac{1}{\pi}\tan^{-1}\left(\frac{x-\mu}{\sigma}\right) + \frac{1}{2} = 0.5$$
$$\implies \tan^{-1}\left(\frac{x-\mu}{\sigma}\right) = 0$$

Thus, for the median, we need to solve the root-finding equation $\tan^{-1}\left(\frac{x-\mu}{\sigma}\right) = 0$. Similarly, for any arbitrary quantile at level p, we have:

$$F(x) = p$$
$$\implies \frac{1}{\pi}\tan^{-1}\left(\frac{x-\mu}{\sigma}\right) + \frac{1}{2} - p = 0$$

This requires numerical root-finding methods to solve for x.

20.4 Regula-Falsi (False Position) Method

Core Principle

Given a continuous function $f : \mathbb{R} \to \mathbb{R}$ and an interval $[a, b]$ such that $f(a)f(b) < 0$, we consider the secant line joining $(a, f(a))$ and $(b, f(b))$ and take the x-intercept c as our estimate.
We then choose either the interval $[a, c]$ or $[c, b]$ using the Intermediate Value Theorem (IVT) and proceed iteratively.

Formulation

The equation of the line joining $(a, f(a))$ and $(b, f(b))$ is given by:

$$y - f(b) = \frac{f(a) - f(b)}{a - b}(x - b)$$

To find the x-intercept, we substitute $y = 0$. Thus, we have:

$$\text{x-intercept } c = \frac{bf(a) - af(b)}{f(a) - f(b)}$$

Algorithm

For $n = 0, 1, 2, \ldots,$ until satisfied, do the following steps:

1. Set $c = \dfrac{b_n f(a_n) - a_n f(b_n)}{f(a_n) - f(b_n)}$.
2. If $f(c) = 0$, exit (exact root found).
3. If $f(a_n)f(c) < 0$, set $a_{n+1} = a_n$ and $b_{n+1} = c$.
4. Otherwise, set $a_{n+1} = c$ and $b_{n+1} = b_n$.

Notes on Regula-Falsi

- Like the bisection method, one can prove the guaranteed convergence of the Regula-Falsi method (since it brackets the root).
- However, the explicit error bounds of the Regula-Falsi method are extremely messy to formulate compared to Bisection.

General Stopping Criteria

1. **Explicit number of iterations:** Stop after a fixed $n = 10, 20, 50$, etc.
2. **Accuracy to d decimal places:** We stop at iteration n if x_n and x_{n-1} are accurate to d decimal places, i.e.,

$$|x_n - x_{n-1}| \leq 0.5 \times 10^{-d}$$

Example: Homework

Compute the root of $f(x) = x^3 - x - 1$ in the interval $[1, 2]$ using the Regula-Falsi method, up to 3 iterations.

20.5 Secant Method

Core Principle

Given a continuous function $f : \mathbb{R} \to \mathbb{R}$ and two initial guesses x_0 and x_1, we compute the next estimate iteratively: x_{n+1} is the x-intercept of the line joining $(x_n, f(x_n))$ and $(x_{n-1}, f(x_{n-1}))$.

Example: Homework

Compute the explicit formula for x_{n+1} and write out the formal Algorithm for the Secant Method.

Convergence & Stopping Criterion

Stopping Criterion:
1. Fixed number of iterations.
2. Accurate to d decimal places for consecutive iterations, i.e., $|x_n - x_{n-1}| \leq 0.5 \times 10^{-d}$.

Important Note: The secant method completely skips the IVT bracket-checking process of the Regula-Falsi method. Hence, **convergence is NOT guaranteed**.

20.6 Newton-Raphson Method

Core Principle

Given a differentiable function $f : \mathbb{R} \to \mathbb{R}$ and an initial point x_0, we define the next estimate iteratively as: x_{n+1} is the x-intercept of the **tangent line** at the point $(x_n, f(x_n))$.

Geometric Formulation

The equation of the tangent line at $(x_n, f(x_n))$ is:

$$y - f(x_n) = f'(x_n)(x - x_n)$$

To find the x-intercept, substitute $y = 0$, to get:

$$-f(x_n) = f'(x_n)(x_{n+1} - x_n) \implies x_{n+1} - x_n = -\frac{f(x_n)}{f'(x_n)}$$

This implies the update rule:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Taylor Series Formulation

Given a differentiable function $f : \mathbb{R} \to \mathbb{R}$, from the first-order Taylor series expansion we have:

$$f(x+h) = f(x) + hf'(x) + O(h^2) \implies f(x+h) \approx f(x) + hf'(x)$$

Substitute $x = x_n$ and $h = x_{n+1} - x_n$ to get:

$$f(x_{n+1}) \approx f(x_n) + (x_{n+1} - x_n)f'(x_n)$$

Assuming we want the next point to be the root, we set $f(x_{n+1}) = 0$, giving:

$$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n) \implies x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Remarks on Newton-Raphson

- The convergence of the Newton-Raphson method is **NOT guaranteed**. It can fail if the initial guess is too far from the true root, or if the derivative $f'(x_n)$ vanishes (becomes zero) at any iteration.
- **Stopping Criterion:** Similar to the Secant method, we stop after a maximum number of iterations or when $|x_n - x_{n-1}| \leq 0.5 \times 10^{-d}$.

Reference: Kindly refer to Section 3.5 of Conte and de Boor's book to read more about the convergence of the Secant and Newton-Raphson Methods.

21 Interpolation

Polynomial Interpolation

Aim: Given data points $\{(x_0, y_0), \ldots, (x_n, y_n)\}$, we want to fit a polynomial $P(x)$ such that $P(x_i) = y_i \ \forall i$.

Theorem: Existence and Uniqueness of Interpolating Polynomial

Let $(x_0, y_0), \ldots, (x_n, y_n)$ be $(n+1)$ points such that $x_i \neq x_j \ \forall i \neq j$. Then there exists a unique polynomial $P(x)$ of degree up to n such that $P(x_i) = y_i \ \forall i$.

Proof. Suppose $P(x)$ is a polynomial of degree up to n such that $P(x_i) = y_i$ for every $0 \leq i \leq n$. Let

$$P(x) = a_0 + a_1 x + \ldots + a_n x^n \quad (a_i \in \mathbb{R}, \forall i)$$

Since $P(x_i) = y_i$, we get a system of linear equations:

$$a_0 + a_1 x_i + \ldots + a_n x_i^n = y_i \quad \forall 0 \leq i \leq n$$

This can be written in matrix form as:

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}}_{\text{Vandermonde matrix } (V)} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

The determinant of the Vandermonde matrix is known to be:

$$\det(V) = \prod_{0 \leq i < j \leq n} (x_j - x_i)$$

Since all x_i are distinct, $\det(V) \neq 0$. Therefore, the matrix is invertible, and a unique solution for the coefficients a_i exists. □

21.1 Lagrange Interpolation

Core Theory

Idea: Construct polynomials $l_0, \ldots, l_n$ of degree up to n such that:

$$l_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad \forall 0 \leq i, j \leq n$$

For example: $l_0(x_0) = 1$, and $l_0(x_j) = 0 \; \forall j \geq 1$, and so on.
Then, we define the interpolating polynomial as a linear combination:

$$P(x) = \sum_{i=0}^{n} y_i l_i(x)$$

Evaluating this at any point x_k gives:

$$P(x_k) = \sum_{i=0}^{n} y_i l_i(x_k) = y_k l_k(x_k) = y_k \quad \text{for every } 0 \leq k \leq n$$

So, $P(x)$ is the required unique interpolating polynomial.

Construction of Basis Polynomials $l_i(x)$

We require:

$$l_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Since $l_i(x_j) = 0$ for all $j \neq i$, the term $(x - x_j)$ must divide $l_i(x)$ for every $j \neq i$.
So, we construct it as:

$$l_i(x) = c \prod_{\substack{j=0 \\ j \neq i}}^{n} (x - x_j)$$

where c is a constant.
Substituting the condition $l_i(x_i) = 1$, we solve for c:

$$1 = c \prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j)$$

$$\implies c = \frac{1}{\prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j)}$$

Therefore, the Lagrange basis polynomials are defined as:

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \quad \forall\, 0 \leq i \leq n$$

Example: Lagrange Interpolation Example

Compute the interpolating polynomial passing through $(0, 1), (1, 2)$ and $(2, 5)$ using Lagrange Interpolation.
Solution: We have $(x_0, y_0) = (0, 1)$, $(x_1, y_1) = (1, 2)$, and $(x_2, y_2) = (2, 5)$.
First, we calculate the basis polynomials:

$$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x-1)(x-2)}{(-1)(-2)} = \frac{1}{2}(x^2 - 3x + 2)$$

$$l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{x(x-2)}{(1)(-1)} = 2x - x^2$$

$$l_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{x(x-1)}{(2)(1)} = \frac{1}{2}(x^2 - x)$$

Now, we construct the final interpolating polynomial:

$$\begin{aligned} P(x) &= y_0 l_0(x) + y_1 l_1(x) + y_2 l_2(x) \\ &= (1)\left[\frac{1}{2}(x^2 - 3x + 2)\right] + (2)\left[2x - x^2\right] + (5)\left[\frac{1}{2}(x^2 - x)\right] \\ &= \frac{1}{2}x^2 - \frac{3}{2}x + 1 + 4x - 2x^2 + \frac{5}{2}x^2 - \frac{5}{2}x \\ &= x^2 + 1 \end{aligned}$$

Drawback of Lagrange Interpolation

Given a new data point (x_{n+1}, y_{n+1}), Lagrange Interpolation is computationally inefficient. We cannot easily update the existing polynomial; we have to recompute all the basis polynomials $l_i(x)$ from scratch to find the new $P(x)$.

21.2 Newton's Divided Difference Method

Core Theory

Given data points $(x_0, y_0), \ldots, (x_n, y_n)$, we define polynomials $P_0(x), \ldots, P_n(x)$ iteratively as follows: $P_i(x)$ interpolates the first $i+1$ points, satisfying $P_i(x_k) = y_k \ \forall 0 \leq k \leq i$.
Constructing $P_{i+1}(x)$ from $P_i(x)$: Since $P_i(x)$ is a polynomial of degree $\leq i$ and correctly interpolates the first i points, we write:

$$P_{i+1}(x) = P_i(x) + c_i(x)$$

where $c_i(x)$ satisfies the following conditions:
1. $c_i(x_k) = 0 \quad \forall 0 \leq k \leq i$ (to preserve the previous interpolations).
2. $P_i(x_{i+1}) + c_i(x_{i+1}) = y_{i+1}$ (to interpolate the new point).
3. $\deg(c_i(x)) \leq i+1$.

From condition (1) and (3), $c_i(x)$ must be of the form:

$$c_i(x) = \alpha_i \prod_{k=0}^{i} (x - x_k)$$

Using condition (2), we solve for the constant α_i:

$$\alpha_i \prod_{k=0}^{i} (x_{i+1} - x_k) = y_{i+1} - P_i(x_{i+1})$$

$$\implies \alpha_i = \frac{y_{i+1} - P_i(x_{i+1})}{\prod_{k=0}^{i}(x_{i+1} - x_k)}$$

The coefficients α_i are known as **Divided Differences**. They are built recursively using a divided difference table.
Recursive Definitions:

- **0th Divided Difference:** $f[x_i] = y_i$
- **1st Divided Difference (FDD):** $f[x_i, x_{i+1}] = \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i}$
- **2nd Divided Difference (SDD):** $f[x_i, x_{i+1}, x_{i+2}] = \dfrac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$

The final Newton Interpolating Polynomial takes the form:

$$P_n(x) = y_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \ldots + f[x_0, \ldots, x_n] \prod_{k=0}^{n-1} (x - x_k)$$

Example: Newton's Divided Difference Example

Compute the quadratic polynomial passing through $(0,1),(1,2)$ and $(2,5)$ using Newton's Divided Difference method.

Solution: We construct the divided difference table:

x	$y = f[x_i]$	**FDD** $f[x_i, x_{i+1}]$	**SDD** $f[x_i, x_{i+1}, x_{i+2}]$
0	1		
		$\frac{2-1}{1-0} = 1$	
1	2		$\frac{3-1}{2-0} = 1$
		$\frac{5-2}{2-1} = 3$	
2	5		

Taking the top entries of each column as our coefficients, we get:

$$\begin{aligned} P_2(x) &= 1 + 1(x-0) + 1(x-0)(x-1) \\ &= 1 + x + x(x-1) \\ &= x^2 + 1 \end{aligned}$$

Now, consider we receive a new data point $(3,8)$. We simply append it to the bottom of our table and calculate the new differences:

x	y	**FDD**	**SDD**	**TDD** $f[x_0, x_1, x_2, x_3]$
0	1			
		1		
1	2		1	
		3		$\frac{0-1}{3-0} = -\frac{1}{3}$
2	5		$\frac{3-3}{3-1} = 0$	
		$\frac{8-5}{3-2} = 3$		
3	8			

Thus, the new interpolating polynomial updates efficiently by just adding the new term:

$$\begin{aligned} P_3(x) &= P_2(x) - \frac{1}{3}(x-0)(x-1)(x-2) \\ &= (x^2+1) - \frac{1}{3}(x^3 - 3x^2 + 2x) \\ &= -\frac{1}{3}x^3 + 2x^2 - \frac{2}{3}x + 1 \end{aligned}$$

22 Numerical Integration

Aim

Given a continuous function $f : [a, b] \to \mathbb{R}$, we want to approximate the definite integral:

$$\int_a^b f(x)dx$$

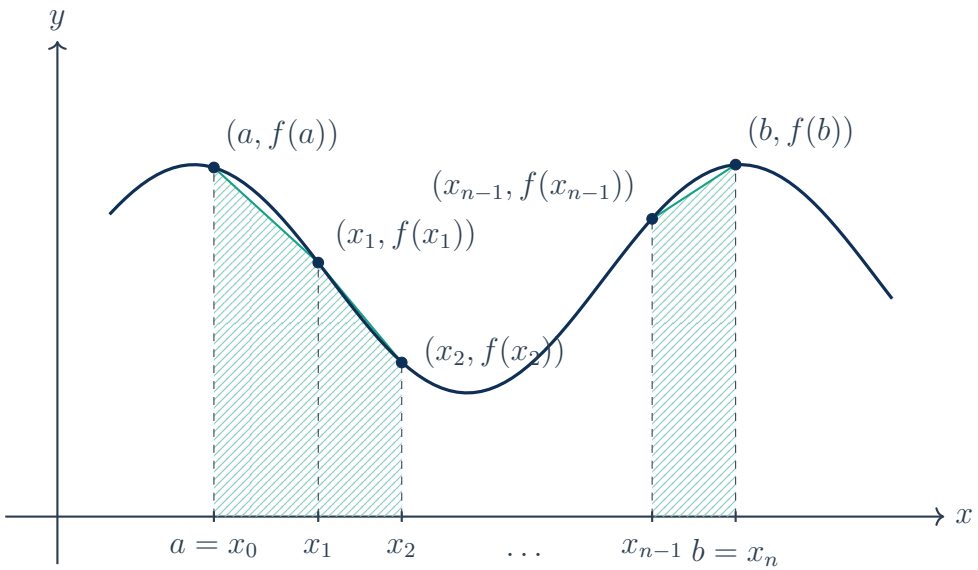
22.1 Trapezoidal Rule

Core Theory

Given a continuous function $f : [a, b] \to \mathbb{R}$, we divide $[a, b]$ into n sub-intervals $[x_i, x_{i+1}]$ for $i = 0, \ldots, n-1$, where:

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b$$

In each sub-interval $[x_i, x_{i+1}]$, we draw a line segment joining the points $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$, and we approximate $\int_{x_i}^{x_{i+1}} f(x)dx$ by calculating the area of the resulting trapezoid.



Derivation

Firstly, the equation of the line joining $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$ is:

$$\frac{y - f(x_i)}{f(x_{i+1}) - f(x_i)} = \frac{x - x_i}{x_{i+1} - x_i}$$

This implies:

$$y = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i)$$

Integrating this linear approximation over the sub-interval gives:

$$\begin{aligned}
\int_{x_i}^{x_{i+1}} f(x)dx &\approx f(x_i)\int_{x_i}^{x_{i+1}} dx + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}\int_{x_i}^{x_{i+1}} (x - x_i)dx \\
&= hf(x_i) + \frac{f(x_{i+1}) - f(x_i)}{h}\left[\frac{x^2}{2} - x_i x\right]_{x_i}^{x_{i+1}} \\
&= hf(x_i) + \frac{f(x_{i+1}) - f(x_i)}{h}\left[\left(\frac{x_{i+1}^2}{2} - x_i x_{i+1}\right) - \left(\frac{x_i^2}{2} - x_i^2\right)\right] \\
&= hf(x_i) + \frac{f(x_{i+1}) - f(x_i)}{h}\left(\frac{h^2}{2}\right) \\
&= \frac{h}{2}\left[f(x_i) + f(x_{i+1})\right]
\end{aligned}$$

Summing the areas of all n trapezoids, we have:

$$\begin{aligned}
\int_a^b f(x)dx &\approx \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(t)dt \approx \sum_{i=0}^{n-1}\frac{h}{2}\left[f(x_i) + f(x_{i+1})\right] \\
&= \frac{h}{2}\Big[(f(x_0) + f(x_1)) + (f(x_1) + f(x_2)) + \ldots + (f(x_{n-1}) + f(x_n))\Big]
\end{aligned}$$

Thus, we arrive at the final approximation formula for the Trapezoidal Rule:

$$\boxed{\int_a^b f(x)dx \approx \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]}$$

22.2 Simpson's 1/3 Rule

Core Theory

Given a continuous function $f : [a, b] \to \mathbb{R}$, we divide $[a, b]$ into n sub-intervals of equal length $h = \frac{b-a}{n}$, where n is an **even** integer.
We consider the quadratic interpolating polynomial passing through the three points $(x_{i-1}, f(x_{i-1}))$, $(x_i, f(x_i))$, and $(x_{i+1}, f(x_{i+1}))$ for $i = 1, 3, \ldots, n-1$, and we approximate $\int_{x_{i-1}}^{x_{i+1}} f(t)dt$ as the area under this parabola.

Derivation

Let $P_i(t) = a_i t^2 + b_i t + c_i$ for $i = 1, 3, \ldots, n-1$ be the interpolating polynomial in the sub-interval $[x_{i-1}, x_{i+1}]$.
Integrating this quadratic polynomial yields:

$$\begin{aligned}
\int_{x_{i-1}}^{x_{i+1}} f(t)dt &\approx \left[a_i \frac{t^3}{3} + b_i \frac{t^2}{2} + c_i t\right]_{x_{i-1}}^{x_{i+1}} \\
&= \frac{a_i}{3}\left(x_{i+1}^3 - x_{i-1}^3\right) + \frac{b_i}{2}\left(x_{i+1}^2 - x_{i-1}^2\right) + c_i(x_{i+1} - x_{i-1}) \\
&= \frac{a_i}{3}\left[(x_i + h)^3 - (x_i - h)^3\right] + \frac{b_i}{2}\left[(x_i + h)^2 - (x_i - h)^2\right] + c_i(2h) \\
&= \frac{a_i}{3}\left[2h(3x_i^2 + h^2)\right] + \frac{b_i}{2}[4hx_i] + 2c_i h \\
&= \frac{h}{3}\left[2a_i(3x_i^2 + h^2) + 6b_i x_i + 6c_i\right]
\end{aligned}$$

Since $P_i(x)$ interpolates the points exactly, we know:

$$\begin{aligned}
f(x_{i-1}) &= a_i x_{i-1}^2 + b_i x_{i-1} + c_i \\
f(x_i) &= a_i x_i^2 + b_i x_i + c_i \\
f(x_{i+1}) &= a_i x_{i+1}^2 + b_i x_{i+1} + c_i
\end{aligned}$$

Evaluating the linear combination $f(x_{i-1}) + 4f(x_i) + f(x_{i+1})$, we get:

$$\begin{aligned}
&(a_i x_{i-1}^2 + b_i x_{i-1} + c_i) + 4(a_i x_i^2 + b_i x_i + c_i) + (a_i x_{i+1}^2 + b_i x_{i+1} + c_i) \\
&= a_i(x_{i-1}^2 + 4x_i^2 + x_{i+1}^2) + b_i(x_{i-1} + 4x_i + x_{i+1}) + 6c_i \\
&= a_i\left[(x_i - h)^2 + 4x_i^2 + (x_i + h)^2\right] + b_i\left[(x_i - h) + 4x_i + (x_i + h)\right] + 6c_i \\
&= a_i(6x_i^2 + 2h^2) + b_i(6x_i) + 6c_i \\
&= 2a_i(3x_i^2 + h^2) + 6b_i x_i + 6c_i
\end{aligned}$$

Substituting this identity back into our area integral, we get:

$$\int_{x_{i-1}}^{x_{i+1}} f(t)dt \approx \frac{h}{3}\Big[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})\Big] \qquad \forall i = 1, 3, \ldots, n-1$$

Summing this over all pairs of sub-intervals, we have:

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{j=1}^{n/2} \int_{x_{2j-2}}^{x_{2j}} f(x)dx \\ &\approx \frac{h}{3}\Big[\big(f(x_0) + 4f(x_1) + f(x_2)\big) + \ldots + \big(f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\big)\Big]\end{aligned}$$

Thus, we arrive at the final approximation formula for Simpson's 1/3 Rule:

$$\boxed{\int_a^b f(x)dx \approx \frac{h}{3}\left[f(x_0) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + f(x_n)\right]}$$

22.3 Some examples of use numerical integration in Statistics

Example: 1. Probability computation for Normal Distribution

Let $Z \sim \text{Normal}(0, 1)$. Compute $P(0 \le Z \le 1)$ using the Trapezoidal rule and Simpson's 1/3 rule for $n = 4$ sub-intervals .

Solution: We want to approximate $\int_0^1 f(x)dx$, where the PDF $f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$.

Since we need $n = 4$ sub-intervals, we use the nodes $0, 0.25, 0.5, 0.75, 1$, with a step size of $h = \frac{1}{4} = 0.25$.

x_i	0	0.25	0.5	0.75	1
$f(x_i)$	0.39894	0.38667	0.35207	0.30114	0.24197

Using the Trapezoidal Rule we can write

$$\begin{aligned}P(0 \le Z \le 1) &\approx \frac{0.25}{2}\Big[f(0) + 2\big(f(0.25) + f(0.5) + f(0.75)\big) + f(1)\Big] \\ &\approx 0.125\Big[0.39894 + 2(0.38667 + 0.35207 + 0.30114) + 0.24197\Big] \\ &\approx 0.34008\end{aligned}$$

Simpson's 1/3 Rule:

$$\begin{aligned} P(0 \le Z \le 1) &\approx \frac{0.25}{3}\Big[f(0) + 4\big(f(0.25) + f(0.75)\big) + 2f(0.5) + f(1)\Big] \\ &\approx \frac{1}{12}\Big[0.39894 + 4(0.38667 + 0.30114) + 2(0.35207) + 0.24197\Big] \\ &\approx 0.34136 \end{aligned}$$

Example: 2. Computation of Expected Value of a Transformation

Let X be a random variable with PDF:

$$f(x) = \begin{cases} 2x & \text{if } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Task 1: Compute $E[X^2]$ analytically.

$$\begin{aligned} E[X^2] &= \int_0^1 x^2 f(x)dx = \int_0^1 x^2(2x)dx = \int_0^1 2x^3 dx \\ &= \left[2 \cdot \frac{x^4}{4}\right]_0^1 = \frac{2}{4} - 0 = 0.5 \end{aligned}$$

Task 2: Numerical Approximation (Trapezoidal Rule, $h = 0.1$).
We need to evaluate $\int_0^1 g(x)dx$ where $g(x) = 2x^3$. The 11 nodes are $x_i \in \{0, 0.1, 0.2, \ldots, 1.0\}$.

$$\begin{aligned} \int_0^1 2x^3 dx &\approx \frac{0.1}{2}\left[g(0) + 2\sum_{i=1}^{9} g(x_i) + g(1)\right] \\ &\approx 0.05\left[0 + 2\Big(2(0.1)^3 + 2(0.2)^3 + \ldots + 2(0.9)^3\Big) + 2\right] \\ &\approx 0.05\left[0 + 4\Big(0.001 + 0.008 + \ldots + 0.729\Big) + 2\right] \\ &\approx 0.05\,[0 + 4(2.025) + 2] \\ &\approx 0.05\,[8.1 + 2] = 0.05[10.1] = 0.505 \end{aligned}$$

Task 3: Comparison. The analytical result is **0.5**, while the numerical result is **0.505**. The numerical approximation overestimates the true value.
Reason: The integrand $g(x) = 2x^3$ is strictly convex on the interval $(0, 1]$ because its second derivative $g''(x) = 12x > 0$. The Trapezoidal rule interpolates the curve using straight line segments. Secant lines drawn between points on a convex curve will always lie *above* the curve, thus capturing extra area and resulting in a positive truncation error.

Example: 3. Survival analysis

Suppose the hazard rate of a compound is:

$$h(t) = 0.1t \quad \forall t \geq 0$$

The survival function $S(t)$ is defined as:

$$P(X > t) = S(t) = \exp\left(-\int_0^t h(u)du\right)$$

Task 1: Compute $S(5)$ analytically.

$$\int_0^5 0.1u\,du = \left[0.1 \cdot \frac{u^2}{2}\right]_0^5 = 0.05(25) - 0 = 1.25$$

$$S(5) = \exp(-1.25) \approx 0.2865$$

Task 2: Numerical Approximation (Trapezoidal Rule, $h = 0.5$). We approximate $\int_0^5 0.1u\,du$ with $h = 0.5$. The 11 nodes are $u_i \in \{0, 0.5, 1.0, \ldots, 5.0\}$. Let $g(u) = 0.1u$.

$$\int_0^5 g(u)du \approx \frac{0.5}{2}\left[g(0) + 2\sum_{i=1}^{9} g(u_i) + g(5)\right]$$

$$\approx 0.25\Big[0 + 2\big(0.05 + 0.10 + 0.15 + \ldots + 0.45\big) + 0.5\Big]$$

$$\approx 0.25\Big[0 + 2(2.25) + 0.5\Big]$$

$$\approx 0.25\Big[4.5 + 0.5\Big] = 0.25[5.0] = 1.25$$

$$S(5) \approx \exp(-1.25) \approx 0.2865$$

Task 3: Comparison. Both the analytical and numerical integrals yield exactly $\mathbf{1.25}$, resulting in the exact same survival probability.
Reason: The function we integrated, $g(u) = 0.1u$, is a linear function. The error bound for the Trapezoidal rule depends strictly on the second derivative of the function being integrated. Because the second derivative of a line is zero, the straight tops of our numerical trapezoids fit the linear curve perfectly, resulting in zero approximation error.