MAT3007 Notes

Course notes for CUHK-Shenzhen, Spring 2026

Hongyi Chen

Jan 7, 2026

Contents

Chapter 1: Linear Programming

MAT3007 Notes

Included lectures

- Lecture 1 & 2: Introduction and Modeling
- Lecture 3: SVM & Linear Programming
- Lecture 4: Transformation Techniques
- Lecture 5: Geometry of Linear Programming
- Lecture 6: The Simplex Method
- Lecture 7: Simplex Method Continued
- Lecture 8: The Simplex Tableau
- Lecture 9: Simplex Method Continued
- Lecture 10: Complexity of the Simplex Method & Duality Theory
- Lecture 11: Weak and Strong Duality
- Lecture 12: Interpreting the Dual & Sensitivity Analysis

Lecture 1 & 2: Introduction and Modeling

1. Classification of Optimization Problems

- **Constrained** vs **Unconstrained**
- **Linear** vs **Non-linear**
- **Continuous** vs **Integer**

2. Formulating Optimization Problems

Optimization Problem Definition:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, p \end{aligned}$$

- **Decision Variables** $\rightsquigarrow$ Decision variables $\boldsymbol{x}$
- **Objective Function** $\rightsquigarrow$ Objective function $f(\boldsymbol{x})$
- **Constraints** $\rightsquigarrow$ Constraint functions: equalities / inequalities

Lecture 3: SVM & Linear Programming

1. Support Vector Machine (SVM) - Classification

Goal: Find a linear function $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ separating the data points such that:

$$\begin{aligned} f(\boldsymbol{x}_i) &\geq +1 \quad \text{if } y_i = +1 \\ f(\boldsymbol{x}_i) &\leq -1 \quad \text{if } y_i = -1 \end{aligned}$$

> **Note: Why +1 and -1?** The linear classifier is not unique. We use the constants +1 and -1 to fix the scale of $\boldsymbol{w}$ and b. The margin is given by $\frac{2}{\|\boldsymbol{w}\|}$.

A. Hard-margin SVM Find the hyperplane that maximizes the margin:

$$\begin{aligned} \underset{\boldsymbol{w},b}{\text{maximize}} \quad & \frac{2}{\|\boldsymbol{w}\|} \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

→ **Equivalent to minimizing:**

$$\begin{aligned} \underset{\boldsymbol{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

B. Soft-margin SVM Used when the training set cannot be perfectly separated by a hyperplane (not linearly separable).

- **Hinge Loss:**
$$\max\{0, 1 - y_i f(\boldsymbol{w}^T\boldsymbol{x}_i + b)\}$$
  - If a point is on the correct side and far enough from the margin, loss is 0.
  - If a point is on the wrong side or within the margin, loss is positive ($\propto$ degree of misclassification).

2. Linear Optimization Formulation for SVM

Define slack variable $t_i = \max\{0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\}$. We can rewrite the SVM Problem as:

$$\begin{aligned} \underset{\boldsymbol{w},b,\boldsymbol{t}}{\text{minimize}} \quad & \sum_{i=1}^{m} t_i \\ \text{subject to} \quad & t_i = \max\{0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\}, \quad i = 1, \ldots, m \end{aligned}$$

Relaxation Steps: 1. Relax "=" to "≥". 2. Split the max function constraint: $t_i \geq \max\{0, \ldots\}$ is equivalent to $t_i \geq \ldots$ AND $t_i \geq 0$.

Final Optimization Problem:

$$\begin{aligned} \underset{\boldsymbol{w},b,\boldsymbol{t}}{\text{minimize}} \quad & \sum_{i=1}^{m} t_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + t_i \geq 1, \quad i = 1, \ldots, m \\ & t_i \geq 0, \quad i = 1, \ldots, m \end{aligned}$$

3. Linear Programming (LP) Definitions

An LP problem is an optimization problem where the objective and all constraints are linear.

General Form of LP:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A_1}\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{A_2}\boldsymbol{x} \leq \boldsymbol{d} \\ & \boldsymbol{A_3}\boldsymbol{x} = \boldsymbol{f} \\ & x_i \geq 0, \quad i \in N_1 \\ & x_i \leq 0, \quad i \in N_2 \\ & x_i \text{ free}, \quad i \in N_3 \end{aligned}$$

Standard Form of LP:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & x_i \geq 0 \end{aligned}$$

- Where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, assume $m < n$ to ensure infinite solutions (degrees of freedom).

4. Converting to Standard Form

1. **Objective Function:** Convert maximization to minimization.

$$\underset{\boldsymbol{x}}{\text{maximize}} \quad f(\boldsymbol{x}) \quad \Longrightarrow \quad \underset{\boldsymbol{x}}{\text{minimize}} \quad -f(\boldsymbol{x})$$

2. **Inequalities:** Convert to equalities by introducing slack/surplus variables.
   - For $\boldsymbol{A_1}\boldsymbol{x} \geq \boldsymbol{b}$, introduce **surplus** variable $\boldsymbol{s_1} \geq 0$:

$$\boldsymbol{A_1}\boldsymbol{x} - \boldsymbol{s_1} = \boldsymbol{b}$$

   - For $\boldsymbol{A_2}\boldsymbol{x} \leq \boldsymbol{d}$, introduce **slack** variable $\boldsymbol{s_2} \geq 0$:

$$\boldsymbol{A_2}\boldsymbol{x} + \boldsymbol{s_2} = \boldsymbol{d}$$

3. **Free Variables:** Convert to non-negative variables.
   - For x_i free, represent it as the difference of two non-negative variables:

$$x_i = x_i^+ - x_i^- \quad \text{where } x_i^+, x_i^- \geq 0$$

Lecture 4: Transformation Techniques

1. Maximin and Minimax Problems
2. How to deal with absolute values

Lecture 5: Geometry of Linear Programming

1. Solve LP using Graph (Example)

$$\begin{aligned} \text{maximize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & x_1 \leq 100 \\ & 2x_2 \leq 200 \\ & x_1 + x_2 \leq 150 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Procedure: 1. Draw the **feasible region**. 2. Draw the level sets of the objective function $x_1 + 2x_2 = c$. 3. Find the largest c such that the line still touches the feasible region.

Observations: * The feasible region of an LP is a **polyhedron**. * The optimal solution (if it exists) is at a **vertex (corner point)** of the feasible region. * Some constraints are active, some are not.

2. Key Definitions

- **Polyhedron:** A set that can be written in the form $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$.
- **Convex Set:** A set $S \subseteq \mathbb{R}^n$ is convex if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ and any $\lambda \in [0, 1]$, the point $\lambda\boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in S$.

3. Extreme Points

Assume a standard form LP where $\boldsymbol{A}$ has full row rank m.

3.1. Basic Solution $\boldsymbol{x}$ is a **basic solution** iff: 1. $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 2. There exist indices $B(1), \ldots, B(m)$ such that the columns $[\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(m)}]$ are linearly independent. 3. $x_i = 0$ for all $i \notin \{B(1), \ldots, B(m)\}$ (Non-basic variables).

> **Algorithm to find a basic solution:** 1. Choose any m linearly independent columns of $\boldsymbol{A}$. 2. Set corresponding variables as **basic variables**; set the rest as **non-basic variables** (0). 3. Solve the linear system for the basic variables.

3.2. Basic Feasible Solution (BFS) A basic solution $\boldsymbol{x}$ is a **Basic Feasible Solution** iff $\boldsymbol{x} \geq 0$.

3.3. Fundamental Theorem of LP Consider a standard form LP where $\boldsymbol{A}$ has full row rank m: * If the feasible set is non-empty $\rightarrow$ there exists a BFS. * If there exists an optimal solution $\rightarrow$ there exists an optimal BFS.

Lecture 6: The Simplex Method

Motivation: How to search efficiently among Basic Feasible Solutions (BFS)? **Idea:** Proceed from one BFS to a neighboring BFS to continuously improve the objective function until optimality is reached.

Key Questions: 1. What is a neighboring BFS? 2. How to find & move to the neighboring BFS? 3. How to stop?

1. Neighboring BFS

Definition: Two basic feasible solutions are **neighboring** iff they differ in exactly one basic variable.

- **Setup:** Assume current BFS has basis B.

$$\mathbf{A}_B = [\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(m)}]$$

  Let $\mathbf{A}_N$ be the non-basic columns. Then $\mathbf{A} = [\mathbf{A}_B, \mathbf{A}_N]$ and $\mathbf{x} = [\mathbf{x}_B; \mathbf{x}_N]$. Currently: $\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b}$ and $\mathbf{x}_N = 0$.

2. Moving to a Neighboring BFS

We select a non-basic variable x_j to enter the basis (increase from 0). Move from $\boldsymbol{x}$ to $\boldsymbol{x} + \theta\boldsymbol{d}$, where $\theta > 0$.

- **Direction Vector $\boldsymbol{d}$:** We need $\mathbf{A}(\boldsymbol{x} + \theta\boldsymbol{d}) = \boldsymbol{b} \implies \mathbf{A}\boldsymbol{d} = 0$.

$$\mathbf{A}_B\boldsymbol{d}_B + \mathbf{A}_N\boldsymbol{d}_N = 0$$

  Since only x_j changes among non-basics ($d_j = 1$, others 0):

$$\mathbf{A}_B\boldsymbol{d}_B + \boldsymbol{A}_j = 0 \implies \boldsymbol{d}_B = -\mathbf{A}_B^{-1}\boldsymbol{A}_j$$

$$\implies \boldsymbol{d} = [\boldsymbol{d}_B; \boldsymbol{d}_N] = [-\mathbf{A}_B^{-1}\boldsymbol{A}_j; 0; \ldots; 1; \ldots; 0]$$

- **Change in Objective Function:**

$$\text{New Cost} = \boldsymbol{c}^T(\boldsymbol{x} + \theta\boldsymbol{d}) = \boldsymbol{c}^T\boldsymbol{x} + \theta\boldsymbol{c}^T\boldsymbol{d}$$

  The rate of change is $\boldsymbol{c}^T\boldsymbol{d}$.

  Definition: Reduced Cost The reduced cost of variable x_j is:

$$\bar{c}_j = \boldsymbol{c}^T\boldsymbol{d} = \boldsymbol{c}_j - \boldsymbol{c}_B^T\mathbf{A}_B^{-1}\boldsymbol{A}_j$$

  A negative reduced cost means introducing x_j improves (reduces) the objective.

3. Stopping Criterion

If all reduced costs $\bar{c}_j \geq 0$ for all non-basic variables x_j, then **STOP**. The current BFS is optimal.

Lecture 7: Simplex Method Continued

Recall `Stopping Criterion`: At a certain $\boldsymbol{x}$, if all reduced costs $\bar{c}_j \geq 0$, then $\boldsymbol{x}$ is optimal. What if some $\bar{c}_j < 0$? $\Rightarrow$ By bringing x_j into the basis, we can reduce the objective.

1. Choosing a Step Size

Assume $\boldsymbol{d}$ is the j-th basic direction with $\bar{c}_j < 0$. Moving in this direction can reduce the objective. How far can we go?

$$\theta^* = \max\{\theta \geq 0 : \boldsymbol{x} + \theta \boldsymbol{d} \geq 0\}$$

To maintain feasibility ($\boldsymbol{x} + \theta \boldsymbol{d} \geq 0$) for every basic variable x_i:

$$\theta^* = \min_{i:d_i<0} -\frac{x_i}{d_i}$$

2. Summary of Simplex Method Steps

Initialization: Find an initial BFS $\boldsymbol{x}$ with basis B. 1. Compute reduced costs $\bar{c}_j$ for all non-basic variables.

$$\bar{c}_j = \boldsymbol{c}_j - \boldsymbol{c}_B^T \mathbf{A}_B^{-1} \boldsymbol{A}_j$$

- If all $\bar{c}_j \geq 0$, STOP. $\boldsymbol{x}$ is optimal. - Else, choose a non-basic variable x_j with $\bar{c}_j < 0$ to enter the basis. 2. Compute the j-th basic direction $\boldsymbol{d}$:

$$\boldsymbol{d} = [-\mathbf{A}_B^{-1} \boldsymbol{A}_j; 0; \ldots; 1; \ldots; 0]$$

- If $d_i \geq 0$ for all basic variables, the problem is unbounded. The optimal value is $-\infty$. STOP. - Else, compute $\theta^* = \min_{i:d_i<0} -\frac{x_i}{d_i}$ to determine the leaving variable. 3. Set $\boldsymbol{y} = \boldsymbol{x} + \theta^* \boldsymbol{d}$ as the new BFS. The objective value decreases by $\theta^* \bar{c}_j$. 4. Repeat these procedures.

3. Degeneracy

Definition: A BFS is **degenerate** if some of the basic variables are 0.

Recall that

$$\theta^* = \min_{i:d_i<0} -\frac{x_i}{d_i}$$

If for some $i \in B$, if $x_i = 0$ and $d_i < 0$, then $\theta^* = 0$. Hence $\boldsymbol{y} = \boldsymbol{x}$,and the objective value does not change.

3.1. Reduced Cost in Degeneracy If the reduced costs vector $\bar{\boldsymbol{c}} \geq 0$, the current BFS is optimal. But when $\boldsymbol{x}$ is optimal, it is possible that some $\bar{c}_j < 0$ (when x is degenerate).

3.2. Pivoting Rules to Avoid Cycling - Smallest Index Rule: Choose the smallest index j with $\bar{c}_j < 0$. - **Most Negative Rule:** Choose the smallest $\boldsymbol{c}_j$. - **Most Decrease Rule:** Choose j with the smallest $\theta^* \bar{c}_j$.

3.3. Bland's Rule If we use the smallest index rule for choosing both the entering index and the exiting index, then no cycle will occur in the simplex algorithm.

Lecture 8: The Simplex Tableau

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

1. Finding an Initial BFS

Two-Phase Method: - **Phase 1:** Solve an auxiliary (辅助的) problem.

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{y}}{\text{minimize}} \quad & \mathbf{1}^\top\boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq 0 \end{aligned}$$

Trivial BFS for this problem is $\boldsymbol{x} = \mathbf{0}$, $\boldsymbol{y} = \boldsymbol{b}$.

Theorem: Feasibility The original problem is feasible iff the optimal value of the auxiliary problem is 0.

We can solve the auxiliary problem by the simplex method: - If the optimal value is not 0, then we can claim that the original problem is infeasible. - If the optimal value is 0 with solution $(x^*, y^* = 0)$, then x^* can be viewed as a BFS for the original problem. $\Rightarrow$ We can start from this point to initialize the simplex method.

If there is at least one basic index in y^* (degenerate case), x^* will not contain the full basis B $\Rightarrow$ pick any other column from A (such that it forms a full-rank matrix with $A_{B(1)}, ..., A_{B(m-1)}$) in the non-basic part in x^* to make it a BFS for the original problem.

1.1. Procedure of the Two-Phase Method - **Phase 1:** 1. Construct the auxiliary problem such that $\boldsymbol{b} \geq 0$. 2. Solve the auxiliary problem using the simplex method. - **Phase 2:** Solve the original problem starting from the BFS $\boldsymbol{x}^*$. $\Rightarrow$ If $(\boldsymbol{x}^*, \boldsymbol{y}^* = \mathbf{0})$ is degenerate for the auxiliary problem (some artificial variables remain in the basis), supplement indices from the original variables to form a full basis for the original problem.

1.2. Big-M Method Consider the following problem:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} + M\sum_{i\in\mathcal{A}} y_i \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq 0 \end{aligned}$$

Initial BFS: $\boldsymbol{x} = \mathbf{0}$, $\boldsymbol{y} = \boldsymbol{b}$.

In the simplex procedure, treat M as a sufficiently large positive constant. If the original problem is feasible, the optimal solution will satisfy $\boldsymbol{y}^* = \mathbf{0}$ and will not involve any artificial variables in the basis.

2. Simplex Tableau

2.1. Structure of the Tableau

$\boldsymbol{c}^\top - \boldsymbol{c}_B^\top\boldsymbol{A}_B^{-1}\boldsymbol{A}$	$-\boldsymbol{c}_B^\top\boldsymbol{A}_B^{-1}\boldsymbol{b}$
$\boldsymbol{A}_B^{-1}\boldsymbol{A}$	$\boldsymbol{A}_B^{-1}\boldsymbol{b}$

Lower Part:

$$\boldsymbol{Ax} = \boldsymbol{b} \Rightarrow \boldsymbol{A_B^{-1}Ax} = \boldsymbol{A_B^{-1}b}$$

- Writing $\boldsymbol{A} = [\boldsymbol{A}_B, \boldsymbol{A}_N]$, we have:

$$\boldsymbol{A}_B^{-1}\boldsymbol{A} = [\mathbf{I}, \boldsymbol{A}_B^{-1}\boldsymbol{A}_N]$$

So, this block must contain an identity matrix. - Writing $\boldsymbol{x} = [\boldsymbol{x}_B; \boldsymbol{x}_N] = [\boldsymbol{A}_B^{-1}\boldsymbol{b}; 0]$. So, this block contains the current BFS.

Upper Part: The term $\bar{\boldsymbol{c}}^\top = \boldsymbol{c}^\top - \boldsymbol{c}_B^\top \boldsymbol{A}_B^{-1}\boldsymbol{A}$ is the reduced cost at current basis.

The term $-\boldsymbol{c}_B^\top \boldsymbol{A}_B^{-1}\boldsymbol{b} = -\boldsymbol{c}_B^\top \boldsymbol{x}_B$ is the negative of the current objective value.

2.2. Canonical Form of the Tableau The simplex tableau should look like (after reordering the columns in $[\boldsymbol{B}; \boldsymbol{N}]$):

$\mathbf{0}_m^\top$	$\boldsymbol{c}_N^\top - \boldsymbol{c}_B^\top \boldsymbol{A}_B^{-1}\boldsymbol{A}_N$	$-\boldsymbol{c}_B^\top \boldsymbol{x}_B$
I_m	$\boldsymbol{A}_B^{-1}\boldsymbol{A}_N$	$\boldsymbol{x}_B$

If in a simplex tableau, the numbers in the top-left block are all nonnegative, then we have reached optimality.

Lecture 9: Simplex Method Continued

1. Pivoting

Step 1: Choose the Incoming Index Check the reduced costs in the top-left block.

Step 2: θ^* and the Leaving Index Assume we chose column j as the incoming index. Incoming column is called the **pivot column**. The step size θ^* is:

$$\theta^* = \min_{i:d_i<0} -\frac{x_i}{d_i}$$

where x_i is the ith entry of the BS and $\boldsymbol{d}_B = -\mathbf{A}_B^{-1}\boldsymbol{A}_j$. **Minimal Ratio Test (MRT)** In the simplex tableau,

$$\theta^* = \min_i \{\frac{x_i}{\bar{A}_{ij}} : \bar{A}_{ij} > 0\}$$

where $x_i \in \boldsymbol{x}_B$ is the lower right block and $\bar{A}$ is the lower left block.

Updating the Tableau: Assume we have determined the incoming and leaving index (pivot element $\bar{A}_{ij}$). We perform the following steps: a. Divide each element in the pivot row by the pivot element. b. Add proper multiples (could be negative) of the pivot row (after the first step) to each other rows, s.t. all other elements in the pivot column become zeros (including the top row).

2. Two-Phase Method in Simplex Tableau

When there is no obvious initial basic feasible solution, we can use the two-phase method.

Phase I: * There is a trivial initial BFS to the auxiliary problem. Then construct the initial tableau based on the trivial initial BFS and solve the auxiliary problem from there.

Phase II: * When enter Phase II, construct the initial tableau from the final tableau of Phase I. * When there is basic index in $y^* = 0$. Choose one index in the non-basic part of x^* to form a new basis B_b. The main difficulty is how to read $\boldsymbol{A}_{B_b}^{-1}\boldsymbol{A}$ from the final tableau of Phase I.

Lecture 10: Complexity of the Simplex Method & Duality Theory

1. Complexity of the Simplex Method

2. Duality Theory

2.1. Formulation of the Dual Problem

Consider a standard form LP:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Write it as:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} + max_{\boldsymbol{y}\in\mathbb{R}^m}\boldsymbol{y}^T(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \geq 0 \end{aligned}$$

Then we have:

$$\min_{\boldsymbol{x}\geq 0} \quad \max_{\boldsymbol{y}\in\mathbb{R}^m} \left(\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{y}^T(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x})\right)$$

Exchange max and min:

$$\max_{\boldsymbol{y}\in\mathbb{R}^m} \left(\boldsymbol{b}^T\boldsymbol{y} + \min_{\boldsymbol{x}\geq 0} \boldsymbol{x}^T(\boldsymbol{c}-\boldsymbol{A}^T\boldsymbol{y})\right)$$

Equivalently,

$$\begin{aligned} \underset{\boldsymbol{y}\in\mathbb{R}^m}{\text{maximize}} \quad & \boldsymbol{b}^T\boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c} \end{aligned}$$

2.2. Definition Given the **Primal Problem** in standard form:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

The corresponding **Dual Problem** is defined as:

$$\begin{aligned} \underset{\boldsymbol{y}\in\mathbb{R}^m}{\text{maximize}} \quad & \boldsymbol{b}^T\boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c} \end{aligned}$$

The dual of the dual problem is the primal problem.

2.3. Primal and Dual Relationship

Primal		Dual	
min	$\boldsymbol{c}^\top \boldsymbol{x}$	max	$\boldsymbol{b}^\top \boldsymbol{y}$
s.t.	$\boldsymbol{a}_i^\top \boldsymbol{x} \geq b_i, \quad i \in M_1,$	s.t.	$y_i \geq 0, \quad i \in M_1$
	$\boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i, \quad i \in M_2,$		$y_i \leq 0, \quad i \in M_2$
	$\boldsymbol{a}_i^\top \boldsymbol{x} = b_i, \quad i \in M_3,$		y_i free, $\quad i \in M_3$
	$x_j \geq 0, \quad j \in N_1,$		$\boldsymbol{A}_j^\top \boldsymbol{y} \leq c_j, \quad j \in N_1$
	$x_j \leq 0, \quad j \in N_2,$		$\boldsymbol{A}_j^\top \boldsymbol{y} \geq c_j, \quad j \in N_2$
	x_j free, $\quad j \in N_3,$		$\boldsymbol{A}_j^\top \boldsymbol{y} = c_j, \quad j \in N_3$

Lecture 11: Weak and Strong Duality

1. Weak Duality Theorem

Primal	Dual
$\min \quad \boldsymbol{c}^\top \boldsymbol{x}$	$\max \quad \boldsymbol{b}^\top \boldsymbol{y}$
s.t. $\quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$	s.t. $\quad \boldsymbol{A}^\top \boldsymbol{y} \leq \boldsymbol{c}$

If $\boldsymbol{x}$ is feasible for the primal problem and $\boldsymbol{y}$ is feasible for the dual, then:

$$\boldsymbol{b}^\top \boldsymbol{y} \leq \boldsymbol{c}^\top \boldsymbol{x}$$

- If the primal problem is unbounded (i.e., the 'optimal value' is $-\infty$), then the dual problem must be infeasible.
- If the dual problem is unbounded (i.e., the 'optimal value' is $+\infty$), then the primal problem must be infeasible.

2. Strong Duality Theorem

Optimality conditions for LP (sufficient): If $\boldsymbol{x}, \boldsymbol{y}$ satisfy: 1. $\boldsymbol{x}$ is primal feasible. 2. $\boldsymbol{y}$ is dual feasible. 3. The objective values coincide, i.e., $\boldsymbol{c}^\top \boldsymbol{x} = \boldsymbol{b}^\top \boldsymbol{y}$.

Then, $\boldsymbol{x}$ and $\boldsymbol{y}$ are optimal solutions of the primal and dual problems, respectively.

Strong Duality Theorem If a linear program has an optimal solution, so does its dual and the optimal values of the primal and dual problems are equal.

3. Optimality, Feasibility, and Boundedness

Thus, solving LPs is equivalent to solving the following linear inequality system (a feasibility problem): * $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ * $\boldsymbol{A}^\top \boldsymbol{y} \leq \boldsymbol{c}$ * $\boldsymbol{b}^\top \boldsymbol{y} = \boldsymbol{c}^\top \boldsymbol{x}$

P D	Finite Optimum	Unbounded	Infeasible
Finite Optimum	Possible		
Unbounded			Possible
Infeasible		Possible	Possible

4. Complementary Conditions

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be feasible solutions for the primal and dual problems, respectively. Then, $\boldsymbol{x}$ and $\boldsymbol{y}$ are optimal if and only if:

$$x_j(\boldsymbol{c}_j - \boldsymbol{A}_j^\top \boldsymbol{y}) = 0, \quad \forall j = 1, \ldots, n$$

This implies that for each j: * Either $x_j = 0$ * Or the j-th dual constraint is active: $\boldsymbol{A}_j^\top \boldsymbol{y} = \boldsymbol{c}_j$

Complementary in Another Form: Sometimes, we write the dual problem (equivalently) as:

Primal		Dual	
min	$\boldsymbol{c}^\top \boldsymbol{x}$	max	$\boldsymbol{b}^\top \boldsymbol{y}$
s.t.	$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$	s.t.	$\boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}$
	$\boldsymbol{x} \geq \boldsymbol{0}$		$\boldsymbol{s} \geq \boldsymbol{0}$

We call $\boldsymbol{s}$ the dual slack variables. Then, the complementarity conditions can be written as:

$$x_i \cdot s_i = 0, \quad \forall i = 1, \ldots, n$$

In general: Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be feasible points of the primal and dual problems. Then $\boldsymbol{x}$ and $\boldsymbol{y}$ are optimal if and only if: * $y_i(\boldsymbol{a}_i^\top \boldsymbol{x} - b_i) = 0, \quad \forall i$ * $x_j(\boldsymbol{A}_j^\top \boldsymbol{y} - c_j) = 0, \quad \forall j$

Lecture 12: Interpreting the Dual & Sensitivity Analysis

1. Interpreting the Dual Problem

- The production planning problem.
- The multi-firm alliance problem.
- The alternative systems problem.

2. Sensitivity Analysis

How do the optimal solution and the optimal value change when the input $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ changes?

Consider the standard form LP:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Denote the associated optimal value by V.

If $\boldsymbol{A}$ and $\boldsymbol{c}$ are fixed, V can be viewed as a function of $\boldsymbol{b}$: $V(\boldsymbol{b})$. **Theorem: Differentiability of the Optimal Value Function** If the dual has a unique optimal solution $\boldsymbol{y}^*$, then $\nabla_{\boldsymbol{b}} V(\boldsymbol{b}) = \boldsymbol{y}^*$.

Similarly, if $\boldsymbol{A}$ and $\boldsymbol{b}$ are fixed, V can be viewed as a function of $\boldsymbol{c}$: $V(\boldsymbol{c})$.

Theorem: Differentiability of $V(\boldsymbol{c})$ If the primal has a unique optimal solution $\boldsymbol{x}^*$, then $\nabla_{\boldsymbol{c}} V(\boldsymbol{c}) = \boldsymbol{x}^*$.

Chapter 2: Integer Programming

MAT3007 Notes

Included lectures

- Lecture 14: Integer Optimization & LP Relaxation
- Lecture 15: Branch-and-Bound Method

Lecture 14: Integer Optimization & LP Relaxation

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{Z}^n \end{aligned}$$

IP 的最优解未必靠近 LP 的最优解: - LP 松弛提供的是连续空间中的最优点；- IP 需要的是离散点集中的最优点；- 两者的几何结构本质不同，因此简单取整通常不可靠。

1. LP Relaxation

虽然 LP 松弛不一定能直接给出好整数解，但它仍然非常有用：作为界。- 对最大化问题，LP relaxation 的最优值 v^{LP} 是 IP 最优值 v^{IP} 的上界。

$$v^{LP} \geq v^{IP}$$

- 对最小化问题，LP relaxation 的最优值 v^{LP} 是 IP 最优值 v^{IP} 的下界。

$$v^{LP} \leq v^{IP}$$

因为 LP 松弛是 "放宽约束''：原来 IP 只能在整数点里选，现在 LP 可以在更大的连续集合里选。

LP 与 IP 最优值之间的差距，叫做 Integrality gap: - 最大化问题：$\text{gap} = v^{LP} - v^{IP}$ - 最小化问题：$\text{gap} = v^{IP} - v^{LP}$

假设是最大化问题，- LP 最优解是 v^{LP} - 把某个解 round 成可行整数解，此时目标值是 $v^{Rounding}$ - 真正的数最优值是是 v^{IP} 有：

$$0 \leq v^{IP} - v^{Rounding} \leq v^{LP} - v^{Rounding}$$

因为虽然你不知道真正最优整数值 v^{IP}，但直到 v^{LP} 是上界，所以离整数最优解最多不会超过 $v^{LP} - v^{Rounding}$

如果 LP relaxation 的最优解恰好是整数，那么它一定也是原 IP 的最优解。

如果我们能保证所有 BFS 都是整数点，那么 LP 一定存在整数最优解。(根据 LP Fundamental Theorem，只要 LP 有最优解，就一定存在一个最优的 BFS)

什么条件能保证一个 LP 的所有 BFS 都是整数? **Total Unimodularity（全酉模性，TU）**矩阵 A 是 totally unimodular，如果它的每一个子矩阵的行列式都只可能取 $0, +1, -1$。

If A is TU and b is an integer vector, then every BFS is integer.

判断 TU 的充分条件：1. 每列至多有两个非零元素。2. 每个元素都属于 $\{0, +1, -1\}$。3. 行可以划分为两个不交集合 B 和 C，使得：- 如果某列有两个非零元素且符号相同，则它们必须在不同集合中。- 如果某列有两个非零元素且符号相反，则它们必须在同一集合中。

Lecture 15: Branch-and-Bound Method

1. Branch-and-Bound Method

若 LP 松弛最优解为 $\boldsymbol{x}^*$，而其中某个变量 x_i^* 不是整数，则建立两个子问题：1. $x_i \leq \lfloor x_i^* \rfloor$ 2. $x_i \geq \lceil x_i^* \rceil$

例如 $x_i^* = 3.75$，则建立两个子问题：1. $x_i \leq 3$ 2. $x_i \geq 4$

对于子问题，继续解 LP 松弛：- 如果某个子问题的 LP 最优解已经是整数，那么这个子问题就“解完了’’；- 如果仍然有分数变量，就继续分支；- 如果子问题不可行，就直接丢弃；- 如果这个子问题即使最优也不可能优于当前已知最好整数解，就也可以丢弃。

反复进行，就形成一棵分支树（branching tree）。

对于最大化问题：- 上界：LP relaxation 的最优值。- 下界：任何已知的整数可行解的目标值。

2. Pruning

如果某个分支的 LP 松弛最优值已经不超过当前下界，那么这个分支就没必要继续算了。

Chapter 3: Non-linear Programming

MAT3007 Notes

Included lectures

- Lecture 16: Non-linear Optimization: Introduction
- Lecture 17: Optimality Conditions for Unconstrained Problems
- Lecture 18: Optimality Conditions for Constrianed Problems
- Lecture 19: KKT Conditions
- Lecture 20: More on KKT Conditions
- Lecture 21: Convexity
- Lecture 22: Convex Optimization
- Lecture 23: Nonlinear Optimization Algorithms
- Lecture 24: Nonlinear Optimization Algorithms (Cont'd)

Lecture 16: Non-linear Optimization: Introduction

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & x \in \Omega \end{aligned}$$

1. First-Order Necessary Conditions (FONC)

如果 $\boldsymbol{x}^*$ 是一个局部最优解，并且 f 在 $\boldsymbol{x}^*$ 处可微，那么：

$$\nabla f(x^*) = 0$$

满足 FONC 的点叫做**驻点**（stationary point）。驻点可能是局部最优解，也可能是局部最差解，或者是一个鞍点。

所以，FONC 是局部最优解的必要条件，但不是充分条件。

Lecture 17: Optimality Conditions for Unconstrained Problems

Recall: 如果 $\boldsymbol{x}^*$ 是一个局部最优解，那么：

$$\nabla f(\boldsymbol{x}^*) = 0$$

满足这个条件的点叫做**驻点** (stationary point)。驻点可能是局部最优解，也可能是局部最差解，或者是一个鞍点。

1. Second-Order Necessary Conditions (SONC)

如果 $\boldsymbol{x}^*$ 是一个局部最小值：

1. $\nabla f(\boldsymbol{x}^*) = 0$;
2. For all $\boldsymbol{d} \in \mathbb{R}^n$, $\boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{d} \geq 0$, i.e. $\nabla^2 f(\boldsymbol{x}^*)$ 半正定

Def: Positive Semi-definiteness (PSD) A symmetric matrix A is positive semi-definite iff for all $\boldsymbol{d} \in \mathbb{R}^n$, $\boldsymbol{d}^T A \boldsymbol{d} \geq 0$.

SONC requires the Hessian matrix $\nabla^2 f(\boldsymbol{x}^*)$ to be PSD, in 1D, this is equivalent to $f''(x^*) \geq 0$.

SONC 依然是必要条件。

反例为 $f(x) = x^3$ 在 $x = 0$ 处满足 SONC，但不是局部最小值。

2. Some Useful Facts about PSD Matrices

- 通常只对对称矩阵讨论 PSD。(如果 A 不对称，用 $\frac{1}{2}(A+A^T)$ 代替，即 $x^T Ax = x^T \frac{1}{2}(A+A^T)x$)
- 一个对称矩阵 PSD iff 它的所有特征值都非负。
- 对于所有矩阵 A，$A^T A$ 都是 PSD。

3. Second-Order Sufficient Conditions (SOSC)

如果 $\boldsymbol{x}^*$ 满足：

1. $\nabla f(\boldsymbol{x}^*) = 0$;
2. For all $\boldsymbol{d} \neq \boldsymbol{0}$, $\boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{d} > 0$, i.e. $\nabla^2 f(\boldsymbol{x}^*)$ is positive definite (PD).

那么 $\boldsymbol{x}^*$ 是一个局部最小值。

因为是充分条件——不是所有局部最小值都满足 SOSC!

总结

对于极小值：

类型	条件
最小值 FONC	$\nabla f = 0$
SONC	Hessian ≥ 0
SOSC	Hessian > 0

对于极大值：

类型	条件
最大值 FONC	$\nabla f = 0$
SONC	Hessian ≤ 0
SOSC	Hessian < 0

4. Saddle Point

如果 $\boldsymbol{x}^*$ 满足：

1. $\nabla f(\boldsymbol{x}^*) = 0$;
2. For some $\boldsymbol{d}$, $\boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{d} < 0$; and for some $\boldsymbol{d}'$, $\boldsymbol{d}'^T \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{d}' > 0$.

那么 $\boldsymbol{x}^*$ 是一个鞍点。

5. 整体解题策略

1. 求梯度

$$\nabla f(\boldsymbol{x}) = \boldsymbol{0}$$

   找到所有候选点
2. 求 Hessian

$$\nabla^2 f(\boldsymbol{x})$$

Hessian	结论
PD	严格局部最小
ND	严格局部最大
不定	鞍点
PSD	不确定

3. 特殊情况
   - 如果函数是凸函数，那么满足 FONC 的点一定是全局最小值。
   - 如果只有一个 stationary point，那么它往往是全局最优解。

Lecture 18: Optimality Conditions for Constrianed Problems

1. Abstract FONC for Constrained Problems

约束优化里，最优性不是看所有方向''能不能下降，而是看所有可行方向'' 能不能下降。(如边界点。虽然不满足 FONC，但仍可能是局部最优解)

Feasible Direction 给定可行点 $x \in \Omega$，如果存在某个 $\bar{t} > 0$，使得 $\forall t \in [0, \bar{t}]$，有 $x + td \in \Omega$，则称 d 为在点 x 处的可行方向。

FONC for Constrained Problems 如果 x^* 是约束优化问题的局部最小值，那么对于所有在 x^* 处的可行方向 d，都有 $\nabla f(x^*)^T d \geq 0$。

Descent Direction 如果 $\nabla f(x^*)^T d < 0$，则称 d 为在 x^* 处的下降方向。

记在 x 处的可行方向集为 $S_\Omega(x)$，下降方向集为 $S_D(x)$，则 FONC 可以重新表述为：

$$S_\Omega(x^*) \cap S_D(x^*) = \emptyset$$

约束问题的一阶必要条件可以重新说成一句非常直观的话——**局部极小点处，不存在可行下降方向。**

2. FONC for Linearly Constrained Non-linear Problems

Consider a linearly constrained problem:

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \end{aligned}$$

FONC for Linearly Constrained Problems 如果 $\boldsymbol{x}^*$ 是一个局部最小值，那么存在一些 $y \in \mathbb{R}^m_+$ 使得：

$$\nabla f(\boldsymbol{x}^*) - \boldsymbol{A}^T \boldsymbol{y} = 0 y_i (a_i^T \boldsymbol{x}^* - b_i) = 0, \quad i = 1, \ldots, m$$

1. 第一个条件：**stationarity** 目标函数的梯度可以表示成活跃约束法向量的非负线性组合。也就是说，目标函数想继续下降的趋势，被约束边界 "顶住了''。
2. 第二个条件：**complementary slackness** 如果第 i 个约束不活跃（即 $a_i^T \boldsymbol{x}^* > b_i$），则对应的 y_i 必须为 0；如果 $y_i > 0$，则第 i 个约束必须活跃（即 $a_i^T \boldsymbol{x}^* = b_i$）。

Corollary: FONC for Linear Equality Constraints

$$\min \quad f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

如果 $\boldsymbol{x}^*$ 是一个局部最小值，那么存在一些 $\boldsymbol{y} \in \mathbb{R}^m$ 使得：

$$\nabla f(\boldsymbol{x}^*) - \boldsymbol{A}^T \boldsymbol{y} = 0$$

Lecture 19: KKT Conditions

1. General Inequality Constraints

Consider the nonlinear program:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \le 0, \quad i = 1, \dots, m \end{aligned}$$

No Feasible Descent 如果 $\boldsymbol{x}^*$ 是一个局部最小值,那么不存在一个向量 $\boldsymbol{d}$ 使得:1. $\nabla f(\boldsymbol{x}^*)^T \boldsymbol{d} < 0$; and 2. $\nabla g_i(\boldsymbol{x}^*)^T \boldsymbol{d} \le 0, \forall i \in A(\boldsymbol{x}^*)$. 其中 $A(\boldsymbol{x}^*) = \{i : g_i(\boldsymbol{x}^*) = 0\}$ 是 $\boldsymbol{x}^*$ 处的活跃约束集合。

Firtz-John (FJ) Conditions 如果 $\boldsymbol{x}^*$ 是一个局部最小值,那么存在一些 $\lambda_0 \ge 0$ 和 $\lambda_1, \dots, \lambda_m \ge 0$ 不全为 0，使得:

$$\lambda_0 \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^m \lambda_i \nabla g_i(\boldsymbol{x}^*) = 0 \lambda_i g_i(\boldsymbol{x}^*) = 0, \quad i = 1, \dots, m$$

缺点：允许 $\lambda_0 = 0$，此时条件就变得很弱了。

Karush-Kuhn-Tucker (KKT) Conditions 若果 $\boldsymbol{x}^*$ 是

$$\min \quad f(\boldsymbol{x}) \quad \text{s.t.} \quad g_i(\boldsymbol{x}) \le 0, i = 1, \dots, m$$

的局部最小值，且活跃约束的梯度

$$\{\nabla g_i(\boldsymbol{x}^*) : i \in A(\boldsymbol{x}^*)\}$$

线性无关，那么存在 $\lambda_i \ge 0$ 使得:

$$\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^m \lambda_i \nabla g_i(\boldsymbol{x}^*) = 0 \lambda_i g_i(\boldsymbol{x}^*) = 0, \quad i = 1, \dots, m$$

2. General Equality and Inequality Constraints

Consider the nonlinear program:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \le 0, \quad i = 1, \dots, m \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, \dots, p \end{aligned}$$

`STEP 1: Lagrangian` - 对每个不等式约束 $g_i(\boldsymbol{x}) \le 0$，引入一个拉格朗日乘子 $\lambda_i \ge 0$; - 对每个等式约束 $h_j(\boldsymbol{x}) = 0$，引入一个拉格朗日乘子 μ_j（没有非负限制）; 定义拉格朗日函数:

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

`STEP 2: KKT Conditions` 1. Main KKT Conditions/Stationarity：目标梯度被约束梯度抵消

$$\nabla_x L(x, \lambda, \mu) = \nabla f(x) + \sum_{i=1}^m \lambda_i \nabla g_i(x) + \sum_{j=1}^p \mu_j \nabla h_j(x) = 0$$

2. Dual Feasibility：不等式乘子非负（作为惩罚项）

$$\lambda_i \geq 0, \quad i = 1, \ldots, m$$

3. Complementarity Conditions：只有活跃约束才可能有影响

$$\lambda_i g_i(x) = 0, \quad i = 1, \ldots, m$$

4. Primal Feasibility：原问题必须可行

$$g_i(x) \leq 0, \quad i = 1, \ldots, mh_j(x) = 0, \quad j = 1, \ldots, p$$

3. Constraint Qualification

KKT 条件的必要性依赖于一个重要的前提——**约束资格（constraint qualification, CQ）**。最常用的 CQ 是**线性独立约束资格（LICQ）**，即所有活跃约束的梯度向量线性无关。

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

线性无关。

Lecture 20: More on KKT Conditions

- KKT conditions is first-order necessary conditions (FONC) for general constrained optimization problems.
- KKT 点只是候选最优点，不保证真最优
- 如果 LICQ 成立，乘子唯一

1. 应用 KKT 的流程

STEP 1：检查 LICQ STEP 2：写出 KKT 条件 STEP 3：从互补条件入手分析，找到 KKT 点

补充：- 若 CQ 成立，则全局最优解一定是 KKT 点 - 若 CQ 成立且 KKT 点唯一，则该点必为全局最优 - 若问题是凸优化，则 KKT 条件会更强（必要且充分）

Lecture 21: Convexity

Recap: Applying KKT

General Strategy: - Check LICQ - Derive KKT conditions - Start by Complementary Slackness then find KKT points

Unconstrained	Constrained
FONC: x^* local min (+CQ) $\implies \nabla f(x^*) = 0$	KKT conditions
SONC: x^* local min (+CQ) $\implies \nabla^2 f(x^*)$ is PSD	?
SOSC: x^* local min (+CQ) $\implies \nabla^2 f(x^*)$ is PD	?

Convexity

Def:

- Convex set

  A set $\Omega \subseteq \mathbb{R}^n$ is convex if for any $x, y \in \Omega$ and $\lambda \in [0,1]$, we have $\lambda x + (1-\lambda)y \in \Omega$.

- Convex conbination

  For any $x_1, ..., x_n$ and $\lambda_1, ..., \lambda_n \geq 0$ such that $\sum_{i=1}^n \lambda_i = 1$, we have $\sum_{i=1}^n \lambda_i x_i$ is a convex combination of $x_1, ..., x_n$.

- Convex function

  A function $f : \Omega \to \mathbb{R}$ is convex if for any $x, y \in \Omega$ and $\lambda \in [0,1]$, we have $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$.

- Concave function

  A function $f : \Omega \to \mathbb{R}$ is concave if for any $x, y \in \Omega$ and $\lambda \in [0,1]$, we have $f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda) f(y)$.

Characterization of Convexity

- Convexity by Hessian > If f is twice differentiable, then f is convex iff its Hessian is PSD, i.e., $d^T \nabla^2 f(x) d \geq 0$ for all x and d.
- Lemmas > Sum Rule > If $a_1, ..., a_n \geq 0$ and $f_1, ..., f_n$ are convex, then $f = \sum_{i=1}^n a_i f_i$ is convex. > > Composition Rule > If g is convex and $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, then $f(x) = g(Ax + b)$ is convex. > > Pointwise maximum > If f_i are convex for $i = 1, ..., n$, then $f(x) = \max_{i=1,...,n} f_i(x)$ is convex. > Pointwise Minimum > If f_i are concave for $i = 1, ..., n$, then $f(x) = \min_{i=1,...,n} f_i(x)$ is concave.

Lecture 22: Convex Optimization

以下两类问题都叫凸优化：- 在凸可行域上最小化凸函数 - 在凸可行域上最大化凹函数

凸性：局部最优 = 全局最优

Proof: 反证法。设 x^* 是局部最优解，但不是全局最优解，则存在 y 使得 $f(y) < f(x^*)$。由于 f 是凸函数，我们有：

$$f(\lambda x^* + (1-\lambda)y) \leq \lambda f(x^*) + (1-\lambda)f(y) < f(x^*)$$

这说明在 x^* 附近存在更小的点，与 x^* 是局部最优解矛盾。

1. Stationarity and Global Optimality

Consider the constrained optimization problem:

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

其中 - f 是凸函数 - g_i 是凸函数 - h_j 是线性函数

那么任何 KKT 点都是全局最优解。

注意：在凸优化里，只要 f 和 Ω 满足凸性条件，KKT 点就一定存在且全局最优，不再依赖于 CQ。

2. Convex Constraints

如何判断约束是否构成凸集?

Lemma： Let g be a convex function. Then, for any c, the region $\Omega = \{x : g(x) \leq c\}$ is a convex set. In this case, we say that $g(x) \leq c$ is a convex constraint.

- $g(x) \leq 0$ and g convex $\Rightarrow$ g convex constraint
- $g(x) \geq 0$ and g concave $\Rightarrow$ g convex constraint
- Linear constraint is convex constraint
- 一个约束即使看起来不属于上面标准形式，它定义出来的集合仍然可能是凸集。

也就是说，上述判定法是充分条件，不是必要条件。

3. Hidden Convexity

有时目标函数或约束的凸性不是直接写在表面上的，需要经过变换或重新理解可行域.

e.g.

$$\begin{aligned} \max_{x,y,z} \quad & xyz \\ \text{s.t.} \quad & x + 2y + 3z \leq 3 \\ & x, y, z \geq 0 \end{aligned}$$

xyz not concave，但可以通过对数变换得到一个凸优化问题：

$$\begin{aligned} \max_{x,y,z} \quad & \log(xyz) = \log x + \log y + \log z \\ \text{s.t.} \quad & x + 2y + 3z \leq 3 \\ & x, y, z \geq 0 \end{aligned}$$

Lecture 23: Nonlinear Optimization Algorithms

Start with an unconstrained optimization problem:

$$\min_x f(x)$$

- Bisection method
- GD Method
- Newton's method

Def: Convergence $\{x^k\}$ converges to x^* iff for every $\epsilon > 0$, there exists a positive integer K s.t. $||x^k - x^*||_2 < \epsilon$ for all $k \geq K$.

1. 一维无约束优化

$f : \mathbb{R} \to \mathbb{R}$ 二分法（Bisection method）：局部最小点满足：$g(x) = f'(x) = 0$，即找导函数的零点。这时问题变成了一个 root-finding problem，可以用二分法来求解。

假设能找到两个点 x_l 和 x_r 使得 $g(x_l) < 0$ 且 $g(x_r) > 0$，则根据连续性，存在 $x^* \in (x_l, x_r)$ 使得 $g(x^*) = 0$。二分法的核心思想是不断缩小区间 (x_l, x_r) 来逼近 x^*。

定义中点 $x_m = \frac{x_l + x_r}{2}$：- 情况 1：如果 $g(x_m) = 0$，则 x_m 就是我们要找的点。- 情况 2：如果 $g(x_m) < 0$，说明零点在区间 (x_m, x_r) 内，因此我们将 x_l 更新为 x_m。- 情况 3：如果 $g(x_m) > 0$，说明零点在区间 (x_l, x_m) 内，因此我们将 x_r 更新为 x_m。

重复此过程，直到 $|x_r - x_l| < \epsilon$ or $|g(x_m)| < \epsilon$，此时 x_m 就是一个近似的局部最小点。

为了得到一个 ϵ-近似的局部最小点，最多需要 $\lceil \log_2 \left(\frac{b-a}{\epsilon}\right) \rceil$ 次迭代，其中 $[a, b]$ 是初始区间。

通过对 f' 是用二分法，我们可以找到一个近似 stationary point。当 f 是凸函数时，这个 stationary point 也是全局最优解。

2. Higher-dimensional Unconstrained Optimization

$$x^{k+1} = x^k + \alpha_k d^k$$

其中 d^k 是一个 descent direction，α_k 是 step size。

每一步包含两个核心问题：- 往哪个方向走？- 走多远？

GD method:

$$d^k = -\nabla f(x^k) x^{k+1} = x^k - \alpha_k \nabla f(x^k)$$

Stopping criterion: $||\nabla f(x^k)||_2 < \epsilon$.

Step size α_k 的选取：- 固定步长：$\alpha_k = \alpha$ （简单，便宜，但太大会震荡，太小会收敛慢）- 精确线搜索：$\alpha_k = \arg\min_{\alpha > 0} f(x^k - \alpha \nabla f(x^k))$ （每步都找到最优步长，但计算代价大）

【例】$f(x) = b^T x + \frac{1}{2} x^T A x$，其中 A 是 PSD 矩阵。梯度：$\nabla f(x) = b + Ax$ 在 x^k 处的梯度下降方向：$d^k = -\nabla f(x^k) = -b - Ax^k$ 沿着这个方向的函数值：

$$\begin{aligned} f(x^k + \alpha d^k) &= b^T (x^k + \alpha d^k) + \frac{1}{2} (x^k + \alpha d^k)^T A (x^k + \alpha d^k) \\ &= f(x^k) + \alpha \nabla f(x^k)^T d^k + \frac{1}{2} \alpha^2 d^{kT} A d^k \end{aligned}$$

因为 A 正定，所以 $d^{kT} A d^k > 0$，因此 $f(x^k + \alpha d^k)$ 是 α 的二次函数。通过求导可以找到最优的 α_k：

$$\alpha_k = -\frac{\nabla f(x^k)^T d^k}{(d^k)^T A d^k}$$

Lecture 24: Nonlinear Optimization Algorithms (Cont'd)

1. Armijo Line Search

除了 Exact Line Search，还有其他的 step size 选取方法：**Backtracking Line Search/Armijo Line Search** 背景：已经找到下降方向 d^k，但不知道合适的步长 α_k。给定参数 $\sigma, \gamma \in (0,1)$，从候选步长集合中选择最大的可接受步长：

$$\{1, \sigma, \sigma^2, \sigma^3, ...\}$$

使得：

$$f(x^k + \alpha_k d^k) - f(x^k) \leq \gamma \alpha_k \nabla f(x^k)^T d^k$$

Armijo condition 的意义：要下降的足够多。1. 从 $\alpha = 1$ 开始 2. 检查 $f(x^k + \alpha d^k) - f(x^k) \leq \gamma \alpha \nabla f(x^k)^T d^k$ - 如果满足：接受 α 作为步长 - 如果不满足：将 α 更新为 $\sigma\alpha$，继续检查，直到满足条件

2. Convergence of GD

1. Lipschitz Continuous Gradient 如果存在 $L > 0$ 使得：

$$||\nabla f(x) - \nabla f(y)||_2 \leq L||x - y||_2, \quad \forall x, y$$

则称 f 的梯度是 Lipschitz continuous 的，L 是 Lipschitz constant。记作 $f \in C_L^{1,1}$。L 控制梯度变化的有多快，L 越大，函数曲率可能越大，梯度变化越剧烈，因此步长的选取需要更谨慎（通常更小）。

【例】$f(x) = \frac{1}{2}x^T Ax + b^T x + c$。梯度：$\nabla f(x) = Ax + b$。所以：

$$||\nabla f(x) - \nabla f(y)|| = ||Ax + b - (Ay + b)|| = ||A(x - y)|| \leq ||A||_{op}||x - y||$$

因此，Lipschitz constant 可以取 $L = ||A||_{op}$。其中 $||A||_{op}$ 是矩阵 A 的 operator norm，定义为：

$$||A||_{op} = \sqrt{\lambda_{\max}(A^T A)} = max_{||d||=1}||Ad||$$

对于对称正定矩阵 A，$||A||_{op}$ 等于 A 的最大特征值 $\lambda_{\max}(A)$。

Theorem: Lipschitz Continuity via Hessians 如果 f 二阶连续可微，则以下条件等价：- $f \in C_L^{1,1}$ - $||\nabla^2 f(x)||_{op} \leq L$ for all x

Descent Lemma 如果 $f \in C_L^{1,1}$，则对于任意 x, y，有：

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

令 $y = x^{k+1} = x^k - \bar{\alpha}\nabla f(x^k)$，那么：

$$y - x^k = -\bar{\alpha}\nabla f(x^k)$$

代入 Descent Lemma：

$$f(x^{k+1}) \leq f(x^k) - \bar{\alpha}\|\nabla f(x^k)\|^2 + \frac{L}{2}\bar{\alpha}^2\|\nabla f(x^k)\|^2$$

整理得：

$$f(x^{k+1}) \leq f(x^k) - \left(\bar{\alpha} - \frac{L}{2}\bar{\alpha}^2\right)\|\nabla f(x^k)\|^2$$

为了保证 $f(x^{k+1}) < f(x^k)$，需要满足：

$$\bar{\alpha} - \frac{L}{2}\bar{\alpha}^2 > 0$$

解这个不等式，得到：

$$0 < \bar{\alpha} < \frac{2}{L}$$

结论：对于 Lipschitz continuous gradient 的函数，选择步长 $\bar{\alpha} \in (0, \frac{2}{L})$ 可以保证每一步梯度下降都会使函数值降低。

Linear Convergence 如果存在 $\eta \in (0,1)$ 和 $\ell \geq 0$ 使得：

$$||x^{k+1} - x^*|| \leq \eta ||x^k - x^*||, \quad \forall k \geq \ell$$

则称 $\{x^k\}$ 线性收敛到 x^*，收敛率为 η。

Strong Convexity 假设 f 有 L-Lipschitz continuous gradient，并且存在 $\mu > 0$ 使得：

$$\mu ||d||^2 \leq d^T \nabla^2 f(x) d \leq L||d||^2, \quad \forall x, d$$

则称 f 是 μ-strongly convex 的。此时，问题有唯一解 x^*，并且 GD method 以线性速率收敛到 x^*，收敛率为 $\eta = 1 - 2M\mu$，不同的线搜索方法会有不同的 M。函数值满足：

$$f(x^{k+1}) - f(x^*) \leq \eta (f(x^k) - f(x^*))$$

- 固定步长：$M = \bar{\alpha}(1 - \frac{L\bar{\alpha}}{2})$. - 精确线搜索：$M = \frac{1}{L}$. - Armijo line search: $M = \gamma \min\{1, \frac{2\sigma(1-\gamma)}{L}\}$.
- 最优固定步长：$\bar{\alpha} = 1/L$, 或者用 exact line search 时，可以得到较优的收敛率 $\eta = 1 - \frac{\mu}{L}$.

定义条件数：$\kappa = \frac{L}{\mu}$，则收敛率可以写成 $\eta = 1 - \frac{1}{\kappa}$. 说明条件数越大，收敛率越慢。

距离误差的预测 rate 为 $\bar{\eta} = \frac{\kappa - 1}{\kappa + 1}$，当 κ 很大时，$\bar{\eta} \approx 1 - \frac{2}{\kappa}$，比函数值的预测 rate 更快。