1 Lecture 1: Two-Player Zero-Sum Finite Matrix Games

Game Elements in This Lecture

Players	$N = 2$ (two players) **[NEW]**
Strategy	Pure strategies $\Omega_i = \{e_{i1}, \ldots, e_{im_i}\}$ (finite) **[NEW]**
Cost	Zero-sum: $J_1 + J_2 = 0$ **[NEW]**

Key Questions Introduced:

- What is a game? (Triple: $\mathcal{N}$, Ω, J)
- What is an equilibrium? (Saddle-point, security strategies)
- How to compute? (Minimax principle)

1.1 Terminology & Notation

1.1.1 Game Definition

A **game** $\mathcal{G}$ consists of 3 components:

1. **Players** $\mathcal{N} = \{1, \ldots, N\}$
2. **Action sets** Ω_i for player i:
   - Finite $\Rightarrow$ matrix game
   - Infinite $\Rightarrow$ continuous-action game
3. **Strategy space** $\Omega = \prod_{i \in \mathcal{N}} \Omega_i$:
   - Pure strategy: $u_i \in \Omega_i$ (action = strategy in normal-form)
   - Strategy profile: $\mathbf{u} = (u_1, \ldots, u_N) \in \Omega$

1.1.2 Cost Function

The **cost function** $J_i : \Omega \to \mathbb{R}$ depends on all players' actions:

$$J_i(\mathbf{u}) = J_i(u_i, u_{-i})$$

where u_{-i} denotes the actions of all players except i.

1.2 Normal Form Representation

1.2.1 Normal Form

A game in **normal form** is represented as $\mathcal{G} = (\mathcal{N}, \Omega_i, J_i)$ where:

- $\mathcal{N} = \{1, \ldots, N\}$ is the set of players
- Ω_i is player i's **pure action set**, and $\Omega = \Omega_1 \times \cdots \times \Omega_N$ the **action profile** space
- $J_i : \Omega \to \mathbb{R}$ is player i's **cost**

1.3 Two-Player Zero-Sum Finite (Matrix) Games

1.3.1 Setting

Players: $\mathcal{N} = \{1, 2\}$ (denoted P_1, P_2)

Action sets (finite): $|\Omega_i| = m_i < \infty$

Define the index set of actions for player i as $M_i = \{1, 2, \ldots, m_i\}$.

We encode pure actions as one-hot vectors:

$$\Omega_i = \{e_{i1}, \ldots, e_{im_i}\} \subset \mathbb{R}^{m_i}$$

where e_{ij} is the j-th standard basis vector in $\mathbb{R}^{m_i}$.

Outcome: When player P_i selects action $j \in M_i$, we write $u_i = e_{ij}$.

1.3.2 Zero-Sum Condition

$$J_1(u_1, u_2) + J_2(u_1, u_2) = 0, \qquad \forall\,(u_1, u_2) \in \Omega_1 \times \Omega_2$$

Therefore, the costs are negatives of each other: $J_2 = -J_1$, so we can define $J := J_1 = -J_2$.

Simplified notation:

$$\mathcal{G}(\mathcal{N}, \Omega_i, J_i) \to \mathcal{G}(\mathcal{N}, \Omega_i, J)$$

1.3.3 Cost Matrix Representation

Given **finite actions**, we express the cost function as:

$$J(u_1, u_2) = u_1^\top A\, u_2$$

where $A \in \mathbb{R}^{m_1 \times m_2}$ is P_1's cost matrix.

For pure actions (j, k), where player P_1 chooses action $j \in M_1$ and player P_2 chooses action $k \in M_2$:

$$J(u_1, u_2) = e_{1j}^\top A\, e_{2k} = [A]_{j,k} = a_{jk}$$

This is called a **matrix game**.

1.4 Solution Concepts for 2-Player Zero-Sum Games

1.4.1 Security Strategies

Players adopt conservative strategies that minimize worst-case outcomes.

Definition (P$_1$ Security Strategy):

P_1 (row player) chooses a **security strategy** to minimize worst-case cost:

$$j^* \in \arg\min_{j \in M_1} \max_{k \in M_2} a_{jk}$$

P_1's **security level** (upper value):

$$J_U := \min_{j \in M_1} \max_{k \in M_2} a_{jk}$$

Definition (P$_2$ Security Strategy):

P_2 (column player) chooses a **security strategy** to maximize worst-case gain (equivalently, minimize P_1's cost):

$$k^* \in \arg\max_{k \in M_2} \min_{j \in M_1} a_{jk}$$

P_2's **security level** (lower value):

$$J_L := \max_{k \in M_2} \min_{j \in M_1} a_{jk}$$

Proposition: In any matrix game, $J_L \le J_U$.

1.4.2 Minimax (Security) Pair

Definition: We call $(j^*, k^*) \in M_1 \times M_2$ a **minimax pair** (or **security pair**) if:

- $j^* \in \arg\min_j \max_k a_{jk}$ (security strategy for P_1)
- $k^* \in \arg\max_k \min_j a_{jk}$ (security strategy for P_2)

Warning: In pure strategies, this pair **need not be an equilibrium**. After observing the opponent's choice, a player may have a better response (ex-post regret). This happens whenever $J_L < J_U$.

1.4.3 Saddle Point Solution

Definition: A pair (j^*, k^*) is a **saddle point** if and only if:

$$a_{j^*k} \le a_{j^*k^*} \le a_{jk^*} \quad \forall j \in M_1,\ \forall k \in M_2$$

Interpretation:

- $a_{j^*k^*} \le a_{jk^*}$ for all j: j^* minimizes P_1's cost in column k^*
- $a_{j^*k} \le a_{j^*k^*}$ for all k: k^* maximizes P_1's cost in row j^*

The **saddle-point value** is:

$$J^* = a_{j^*k^*}$$

This entry is simultaneously **minimum in its column** (no regret for P_1) and **maximum in its row** (no regret for P_2).

1.5 Example: Matching Pennies (Pure Strategies)

1.5.1 Game Setup

Each player has two strategies $\Omega_i = \{H, T\}$ (Heads or Tails).

Rules:

- If both choose the same action, P_2 pays P_1 \$1
- If they choose different actions, P_1 pays P_2 \$1

Vector representation:

$$H \mapsto e_{i1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad T \mapsto e_{i2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Cost matrix (P_1's perspective):

$$A = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

where rows correspond to P_1's actions $\{H, T\}$ and columns to P_2's actions $\{H, T\}$.

1.5.2 Analysis

Example computation: If P_1 chooses H (i.e., $u_1 = e_{11} = [1, 0]^\top$) and P_2 chooses T (i.e., $u_2 = e_{22} = [0, 1]^\top$):

$$J(u_1, u_2) = [1, 0] \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = [1, 0] \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 1$$

P_1 incurs cost $+1$ (pays \$1 to P_2).

Step 1: Find Security Strategies

For P_1: Calculate $J_U := \min_j \max_k a_{jk}$

For row $j = 1$ (action H): $\max_k a_{1k} = \max\{-1, 1\} = 1$

For row $j = 2$ (action T): $\max_k a_{2k} = \max\{1, -1\} = 1$

Therefore: $J_U = \min\{1, 1\} = 1$

P_1's security strategy: $j^* \in \{1, 2\}$ (both actions are security strategies) with upper value $J_U = 1$

For P_2: Calculate $J_L := \max_k \min_j a_{jk}$

For column $k = 1$ (action H): $\min_j a_{j1} = \min\{-1, 1\} = -1$

For column $k = 2$ (action T): $\min_j a_{j2} = \min\{1, -1\} = -1$

Therefore: $J_L = \max\{-1, -1\} = -1$

P_2's security strategy: $k^* \in \{1, 2\}$ (both actions are security strategies) with lower value $J_L = -1$

Observation: Since $J_L = -1 < J_U = 1$, we have a **gap**, indicating **no saddle point exists** in pure strategies.

Step 2: Check for Equilibrium

Security strategy pairs: $\{(1, 1), (1, 2), (2, 1), (2, 2)\}$

Let's check candidate $(j^*, k^*) = (1, 2)$ (i.e., P_1 plays H, P_2 plays T):

Does P_1 have regret? Given P_2 plays $k = 2$ (column 2):

$$a_{12} = 1 > a_{22} = -1$$

P_1 could reduce cost from 1 to -1 by switching to row 2 (action T). **P_1 has regret.**

Does P_2 have regret? Given P_1 plays $j = 1$ (row 1):

$$a_{11} = -1 < a_{12} = 1$$

P_2 already maximizes P_1's cost at column 2. **P_2 has no regret.**

Conclusion: Since P_1 has regret, $(1, 2)$ is **not an equilibrium**.

By symmetry, **all four combinations** of security strategies are minimax pairs, but **none is a saddle point**.

1.6 Key Observations

1.6.1 Summary for Matching Pennies

1. **Security strategies exist:** Both players have (multiple) security strategies
2. **Minimax pairs exist:** Any combination (j^*, k^*) of security strategies forms a minimax pair
3. **No saddle point:** None of these pairs is an equilibrium—at least one player has ex-post regret
4. **Gap:** $J_L = -1 < 1 = J_U$ indicates no pure-strategy saddle point

Motivating Question for Next Lecture:
What if we want to find an equilibrium when no pure-strategy equilibrium exists?
Answer Preview:
We introduce **randomization** (mixed strategies) by replacing one-hot pure action vectors with **probability distributions** over actions.