Lecture 17: Zero-Sum Games

Objectives of this lecture

- Introduce the concept of zero-sum games.
- Prove that zero-sum games have a unique game value (though optimal strategies need not be unique).
- State Yao's minimax principle.

1 Philosophy

Zero-sum games are a special type of two-player game with many applications. In a zero-sum game, one player's gain is exactly the other player's loss. Player 1 wants to maximize the expected payoff, while player 2 wants to minimize it.

By using linear programming duality, we can prove that optimal play in zero-sum games is very structured: there is a number v, called the *value of the game*, such that player 1 has a randomized strategy guaranteeing payoff at least v, and player 2 has a randomized strategy guaranteeing payoff at most v.

If you have seen Nash equilibria, one interpretation of today's lecture is that every finite zero-sum game has a Nash equilibrium. Even stronger, all equilibria have the same expected payoff. This common payoff is the game value.

2 Assumed Knowledge

1. Linear programming duality.
2. Matrix and vector notation.

3 Zero-Sum Games

3.1 Basic setup

For this lecture, let A and B be finite strategy spaces.

A function $p : A \times B \to \mathbb{R}$ is a *payoff function*. This means that if player 1 plays $a \in A$ and player 2 plays $b \in B$, then player 1 earns $p(a, b)$ and player 2 loses $p(a, b)$.

Randomized strategy. Both players may play randomized strategies. This means that they choose distributions D_A and D_B over A and B, respectively. Then the expected payoff (or *game value*) of these strategies is

$$\mathrm{val}(D_A, D_B) := \mathbb{E}_{a\sim D_A}\mathbb{E}_{b\sim D_B}[p(a,b)].$$

Player 1 wants to maximize this value, and player 2 wants to minimize it.

Equilibrium. We say that D_A and D_B are an equilibrium for the game if

- for all $b \in B$, it holds that $\mathbb{E}_{a\sim D_A}[p(a,b)] \geq \mathrm{val}(D_A, D_B)$, and
- for all $a \in A$, it holds that $\mathbb{E}_{b\sim D_B}[p(a,b)] \leq \mathrm{val}(D_A, D_B)$.

In other words, player 1 cannot improve by changing strategies once D_B is fixed, and player 2 cannot improve by changing strategies once D_A is fixed. By linearity, it is enough to check pure deviations.

3.2 Example: Rock, Paper, Scissors

Player 1 \ Player 2	Rock	Paper	Scissors
Rock	$(0,0)$	$(-1,1)$	$(1,-1)$
Paper	$(1,-1)$	$(0,0)$	$(-1,1)$
Scissors	$(-1,1)$	$(1,-1)$	$(0,0)$

In this case, the equilibrium is for both players to play rock, paper, and scissors with equal probabilities $(1/3, 1/3, 1/3)$. The game value is 0.

3.3 Existence of equilibrium

In this section we will prove that every finite zero-sum game has an equilibrium and a unique game value. The proof uses linear programming duality.

Reduction to games with positive payoffs. For a technical reason, it is easiest to first restrict to games with positive payoffs (that is, all $p(a,b) > 0$). This is without loss of generality, because adding the same constant to every payoff shifts the game value by that constant but does not change the optimal strategies.

Algebraic expression for game value. Define

$$\Delta_A = \{x \in \mathbb{R}^A_{\geq 0} : \sum_{a \in A} x(a) = 1\},$$

the set of distributions over A. Define Δ_B similarly.

Let us write the strategy of player 1 as $x \in \Delta_A$ and the strategy of player 2 as $y \in \Delta_B$. Let $P \in \mathbb{R}^{A \times B}$ be the payoff matrix, where $P(a, b) = p(a, b)$.

Then the game value of these strategies is

$$\mathrm{val}(x, y) := x^\top P y = \sum_{a \in A} \sum_{b \in B} x(a) y(b) p(a, b).$$

This exactly captures sampling a from x, sampling b from y, and then taking the resulting payoff.

We want to compare the following two quantities:

$$\max_{x \in \Delta_A} \min_{b \in B} \sum_{a \in A} x(a) p(a, b) \qquad \text{and} \qquad \min_{y \in \Delta_B} \max_{a \in A} \sum_{b \in B} y(b) p(a, b).$$

The zero-sum game theorem says that these two quantities are equal.

Writing down the LP. Suppose player 1 commits to a mixed strategy x first. Then player 2 chooses a pure strategy b minimizing the payoff. So player 1 wants to solve

$$\max_{x \in \Delta_A} \min_{b \in B} \sum_{a \in A} x(a) p(a, b).$$

Introduce a variable V for the guaranteed payoff. Then this becomes the linear program

$$\max V \quad \text{such that} \quad \begin{cases} x(a) \geq 0 \text{ for all } a \in A, \\ \sum_{a \in A} x(a) = 1, \\ \sum_{a \in A} x(a) p(a, b) \geq V \text{ for all } b \in B. \end{cases}$$

Because all payoffs are positive, the optimum has $V > 0$. If we rescale by setting $\hat{x}(a) := x(a)/V$, then the LP is equivalent to

$$\min \mathbf{1}^\top \hat{x} \qquad \text{such that} \qquad \hat{x} \geq 0, \quad P^\top \hat{x} \geq \mathbf{1}.$$

Its optimal value is $1/V$.

By LP duality, the dual program is

$$\max \mathbf{1}^\top y \qquad \text{such that} \qquad y \geq 0, \quad P y \leq \mathbf{1}.$$

Let $y^\star$ be an optimal dual solution, and let $t = \mathbf{1}^\top y^\star = 1/V$. Define

$$z := V y^\star.$$

Then $z \geq 0$ and

$$\mathbf{1}^\top z = V \cdot \mathbf{1}^\top y^\star = V \cdot \frac{1}{V} = 1,$$

so $z \in \Delta_B$. Also,

$$Pz = VPy^\star \leq V\mathbf{1}.$$

This means that for every pure strategy $a \in A$,

$$\sum_{b \in B} z(b)p(a,b) \leq V.$$

So if player 2 plays the mixed strategy z, then every pure strategy of player 1 has expected payoff at most V.

On the other hand, the primal strategy x guarantees payoff at least V against every pure strategy of player 2. Therefore,

$$\max_{x \in \Delta_A} \min_{b \in B} \sum_{a \in A} x(a)p(a,b) = \min_{y \in \Delta_B} \max_{a \in A} \sum_{b \in B} y(b)p(a,b).$$

Any optimal pair of mixed strategies forms an equilibrium, and the common payoff is the unique *game value*. The optimal strategies themselves need not be unique.

Notes. Computing the value of a zero-sum game is essentially equivalent to solving a linear program. Similarly, the fact that zero-sum games have equilibria is closely tied to LP duality itself.

3.4 Yao's Minimax Principle

This section is mostly for enrichment. Soon in the course we will study algorithms in more restricted models. For example, consider an algorithm that receives a stream of integers and wants to estimate the number of distinct integers encountered. The algorithm also wants to use very little space (much less than storing the whole stream).

Instead of designing algorithms, let us think about the opposite direction: can we prove that such an algorithm requires nontrivial space? More generally, how can we prove lower bounds, especially when the algorithms are allowed to be randomized?

More formal setup. Let $\mathscr{A}$ be a class of algorithms (for example, all algorithms with space at most some bound). A *randomized algorithm* can be viewed as a distribution over deterministic algorithms in $\mathscr{A}$.

Let $\mathscr{T}$ be a collection of possible inputs (or instances). Suppose we want to prove a statement such as the following:

Every randomized algorithm supported on $\mathscr{A}$ fails on some input in $\mathscr{T}$ with probability at least 0.01.

Obvious approach. Take an arbitrary randomized algorithm over $\mathscr{A}$, and exhibit an input in $\mathscr{T}$ on which the algorithm succeeds with probability less than 0.99.

Alternate approach. Exhibit a distribution over $\mathscr{T}$ such that every deterministic algorithm in $\mathscr{A}$ succeeds with probability less than 0.99 when the input is drawn from that distribution. Then, by averaging, every randomized algorithm must fail on some input with probability at least 0.01.

You might ask: could the first approach work while the second one does not? The answer is no. This is exactly an application of the zero-sum game theorem. Think of it this way: player 1 chooses a distribution over algorithms (that is, a randomized algorithm), and player 2 chooses a distribution over inputs. The payoff is the probability that the algorithm outputs the correct answer. Then the zero-sum game theorem says that the two approaches above yield the same value. This is known as *Yao's minimax principle.*