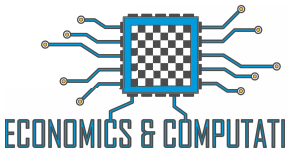

Harvard CS 1360, Spring 2025
Instructor: Ariel Procaccia
Scribe: Sadie Zhao

Minimax Theorem via No-Regret Learning

Lecture 19

In this lecture, we present a proof of the minimax theorem for two-player zero-sum games using tools from online learning, specifically no-regret algorithms. We begin by motivating the concept of regret in decision-making under uncertainty, and formalize it through the weighted majority and randomized weighted majority algorithms. The latter algorithm guarantees that the average performance approaches that of the best fixed action in hindsight. By applying randomized no-regret learning to repeated play in zero-sum games, we yield a simple and intuitive proof of von Neumann's minimax theorem.

1 Recall: Minimax Theorem

We begin by recalling the classical result in game theory:

Theorem 1 (von Neumann, 1928)**.** *Every two-player zero-sum game has a unique value v such that:*

- *Player 1 can guarantee a utility of at least v,*
- *Player 2 can guarantee a utility of at least $-v$.*

This ensures that optimal mixed strategies exist for both players.

2 No-Regret Learning

2.1 Motivation

Suppose a player must choose one of n actions each day (e.g., routes to work), and observes the incurred cost after making a choice. The goal is to *perform nearly as well as the best fixed action in hindsight.* Formally, suppose we have a sequence of time steps $t = 1, 2, \ldots, T$. At each time t, the algorithm selects an action a_t, and incurs a loss (or receives a reward) $\ell_t(a_t)$. Let A be the set of all possible actions.

Regret when minimizing losses:

$$\text{Regret}(T) = \sum_{t=1}^{T} \ell_t(a_t) - \min_{a \in A} \sum_{t=1}^{T} \ell_t(a)$$

Regret when maximizing rewards:

$$\text{Regret}(T) = \max_{a \in A} \sum_{t=1}^{T} r_t(a) - \sum_{t=1}^{T} r_t(a_t)$$

2.2 Formal Model

The interaction is modeled as a cost matrix, and at each time step $t = 1, 2, \cdots, T$,

- The algorithm selects a row i_t (an action).
- The adversary selects a column j_t (the environment).
- The algorithm receives the cost $c(i_t, j_t)$ for the chosen cell and observes the column j_t.

Assume all costs lie in the interval $[0, 1]$.

2.3 Regret

Let T denote the number of rounds. The *average regret* is defined as:

$$\text{Regret}(T) = \frac{1}{T}\sum_{t=1}^{T} c(i_t^{\text{alg}}, j_t) - \min_i \left(\frac{1}{T}\sum_{t=1}^{T} c(i, j_t)\right)$$

In other words, the *average regret* is the average per-day cost of algorithm — the average per-day cost of best fixed row in hindsight.

Definition 1 (No-Regret Algorithm)**.** An algorithm is said to be no-regret if the regret tends to zero as $T \to \infty$.

Remark 1. Note that we are not competing with adaptive strategy, just the best *fixed* row in hindsight.

2.4 Deterministic Algorithms: Examples

Consider the cost matrix:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Alternating strategy: incurs constant regret ($\Theta(1)$).
  - If the algorithm chooses alternating rows at each round, the adversary can choose alternating columns so that $\frac{1}{T}\sum_{t=1}^{T} c(i_t^{\text{alg}}, j_t) = 1$ and $\min_i \left(\frac{1}{T}\sum_{t=1}^{T} c(i, j_t)\right) = \frac{1}{2}$.
- Greedy strategy (choose action with lowest cumulative cost so far): This amounts to alternating rows like before, and still incurs $\Theta(1)$ regret.

This illustrates the fundamental limitation of deterministic algorithms in adversarial settings. In fact, in this specific example, the average regret is always at least $\frac{1}{2}$. This is because in each round, regardless of the algorithm's choice, the adversary can select a column that incurs a cost of 1. Meanwhile, the best fixed action in hindsight incurs an average cost of at most $\frac{1}{2}$, because exactly one of the two actions incurs a cost at each round (so one of them must have incurred a cost in at most half of the rounds).

3 The Weighted Majority Algorithm

3.1 Expert Advice Model

We receive advice from n experts over time and wish to predict outcomes (e.g., weather) as accurately as the best expert in hindsight.

3.2 Algorithm

The idea is that experts are penalized every time they make a mistake.

Weighted Majority Algorithm

- Each expert begins with weight 1.
- Predict using the weighted majority vote.
- After each round, penalize incorrect experts by halving their weight.

3.3 Analysis

Let:

- M: total number of mistakes made by the algorithm,
- m: number of mistakes made by the best expert,
- W: total weight of all experts (initially $W = n$).

Each time the algorithm makes a mistake, at least half of the weighted vote was incorrect. So, at least half the total weight is on incorrect experts, and their weights are halved. This implies the total weight drops by at least $\frac{1}{4}$ after each mistake.

Therefore, after M mistakes:

$$W \leq n \cdot \left(\frac{3}{4}\right)^M.$$

Meanwhile, the best expert, who has made m mistakes, has weight:

$$w_{\text{best}} = \left(\frac{1}{2}\right)^m.$$

Since the best expert's weight is a part of the total weight:

$$\left(\frac{1}{2}\right)^m \leq n \cdot \left(\frac{3}{4}\right)^M.$$

Taking logarithms of both sides:

$$-m \log 2 \leq \log n + M \log\left(\frac{3}{4}\right)$$

It follows that

$$-m \log 2 \leq \log n + M(\log 3 - \log 4) = \log n + M\left(\log 3 - 2\log 2\right),$$

and therefore

$$M \leq \frac{m \log 2 + \log n}{2 \log 2 - \log 3}$$

Numerically, since $\log 2 \approx 0.693$ and $\log 3 \approx 1.098$, this gives:

$$M \leq 2.41(m + \log n)$$

Conclusion: The number of mistakes of the algorithm is close to 2.5 times the number of mistakes of the best expert, but this is still not no-regret as $M - m$ can be larger than m, and m could be on the order of T — as it was in the Example of Section 2.4.

3.4 Slight Modification

Modified Weighted Majority Algorithm

- Each expert begins with weight 1.
- Predict using the weighted majority vote.
- After each round, penalize incorrect experts by removing ε fraction of their weight.

Question: Is there an ε that would guarantee $M \leq (1 + \delta)m$ for a small $\delta > 0$. The answer is no, as the argument of Section 2.4 for the limitation of deterministic algorithm still holds, and in that example we had $M/m = 2$.

4 Randomized Weighted Majority

The idea is to predict proportionally to weights so we can smooth out the worst case.

Randomized Weighted Majority Algorithm:

- Start with all experts having weight 1.
- If the total weight of experts predicting $+$ is w_+ and the total weight of experts predicting $-$ is w_-, then:
  - Predict $+$ with probability $\frac{w_+}{w_+ + w_-}$,
  - Predict $-$ with probability $\frac{w_-}{w_+ + w_-}$.
- Penalize mistakes by removing an ε fraction of weight from each expert who predicted incorrectly.

Theorem 2. *For suitable ε, the randomized weighted majority algorithm has average regret at most $\frac{2\sqrt{T \ln n}}{T} \to 0$.*

5 Minimax via No-Regret

5.1 Setup

In a zero-sum game G, denote:

- V_C as the smallest reward (to the row player) the column player can guarantee if she commits first,
- V_R as the largest reward (to the row player) the row player can guarantee if she commits first.

First, we can see that $V_C \geq V_R$ since if the column player goes the second, they can better respond to the action chosen by the row player and minimize the reward (to the row player). The minimax theorem says equality holds.

5.2 Proof Sketch

We want to show that $V_C \leq V_R$. Assume for contradiction that $V_C > V_R$.

- Shift and scale the rewards in the game so that payoffs lie in $[-1, 0]$ and can now be interpreted as costs, to match the results on no-regret learning. Notice that these operations shift and scale V_C and V_R, so if it was the case that $V_C > V_R$, this still holds. Let $V_C = V_R + \delta$ for some $\delta > 0$.
- Suppose the game is played repeatedly; in each round the row player commits, and the column player responds.
- Let the row player play RWM, and let the column player respond optimally to current mixed strategy.
- Then, after T steps:
  - $U_{\text{ALG}} \geq U_{\text{best row in hindsight}} - 2\sqrt{T \log n}$ by Theorem 2.
  - $U_{\text{ALG}} \leq T \cdot V_R$ since V_R is the largest reward the row player can guarantee if they commit first, and here they are committing to the RWM strategy in each round.
  - **Claim:** Best row in hindsight $\geq T \cdot V_C$

    Proof.
    - Suppose the column player played strategies $s_1, s_2, \ldots, s_T$ over the T rounds.
    - Define a mixed strategy y for the column player that plays each s_t with probability $\frac{1}{T}$ (multiplicities possible).

* Let x be the row player's best response to y. Then

$$V_C \leq u_1(x, y) = \frac{1}{T}\left(u_1(x, s_1) + \cdots + u_1(x, s_T)\right)$$

because the column player is committing first, so they are doing worse than under V_C (and the row player is doing better). Moreover,

$$u_1(x, s_1) + \cdots + u_1(x, s_T) \leq \text{best row in hindsight.}$$

In fact, x might as well be the best row in hindsight (and equality holds above), as it is a best response to y and any pure strategy in the support of a best response is itself a best response.

Putting the two inequalities together proves the claim. □

– It follows that:

$$T \cdot V_R \geq T \cdot V_C - 2\sqrt{T \log n} \quad \Rightarrow \quad \delta T \leq 2\sqrt{T \log n}$$

Contradiction for large T.