No-Regret Learning and Zero-Sum Games

Instructor: Thomas Kesselheim

We have seen a couple of applications of the experts framework. Today, we will get to know another one. To some extend, we could actually have stated the earlier results as applications of today's result.

1 Zero-Sum Games

A zero-sum game is a special case of a two-player game. The game is represented by a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. Player 1 (the *row player*) chooses a row index i; player 2 (the *column player*) chooses a column index j. Given these choices, the row player has to pay $A_{i,j}$ units of money to the column player. (The amount can also be negative.)

Example 28.1. *The famous game* Rock-Paper-Scissors *is represented by the matrix*

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix} .$$

We allow players to randomize their strategies. That is, the row player may choose a vector $\mathbf{x} = (x_1, \ldots, x_m)$, $\sum_{i=1}^m x_i = 1$; the column player may choose a vector $\mathbf{y} = (y_1, \ldots, y_n)$, $\sum_{j=1}^n y_i = 1$. We denote the respective sets of feasible vectors by Δ_m and Δ_n. (These sets of so-called mixed strategies are called the m- or n-dimensional simplex.) Note that the expected outcome can be represented as a vector-matrix-vector product $\mathbf{x}^\top \mathbf{A} \mathbf{y}$.

2 The Minimax Theorem

It seems to be a clear advantage to choose the probabilities only after the other player has done so. But, the main result for zero-sum games, the *minimax theorem*, states that this is actually not true if we allow probability distributions.

Theorem 28.2. *For every matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$*

$$\max_{\mathbf{y} \in \Delta_n} \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A} \mathbf{y} = \min_{\mathbf{x} \in \Delta_m} \max_{\mathbf{y} \in \Delta_n} \mathbf{x}^\top \mathbf{A} \mathbf{y} .$$

Observe that by the order of the maximum and the minimum on the left-hand side the column player moves first and then the row player moves. On the right-hand side, first the row player commits to her probability vector and only then the column player chooses hers, possibly depending on the row player's choice. So, the theorem states that does not matter if one of the two players moves first or both move simultaneously. The quantity $\lambda = \max_{\mathbf{y} \in \Delta_n} \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A} \mathbf{y}$ is called the *value* of the game.

As we have already realized, it is always better to move second. Showing "$\leq$" in Theorem 28.2 is straightforward and the statement that really needs a proof is that also "$\geq$" holds.

3 An Experts Algorithm as Row Player

We will now prove the minimax theorem. Recall that an experts algorithm computes probability vectors $\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(T)}$. So, if we have m experts, this could be possible choices of probability vectors for the row player. Our goal is to use the no-regret property to show the "$\geq$" part of equality.

We still have to define the cost of an expert i. To this end, we use that, in step t, the algorithm deterministically determines the vector $\mathbf{p}^{(t)}$ before seeing $\ell^{(t)}$. Therefore, $\ell^{(t)}$ can be the game outcome provided that the column player moves after the row player. That is, let $\mathbf{y}^{(t)}$ maximize $(\mathbf{p}^{(t)})^\top \mathbf{A}\mathbf{y}^{(t)}$ and set $\ell^{(t)} = \mathbf{A}\mathbf{y}^{(t)}$.

By this definition

$$\sum_{i=1}^{m} p_i^{(t)} \ell_i^{(t)} = (\mathbf{p}^{(t)})^\top \ell^{(t)} = (\mathbf{p}^{(t)})^\top \mathbf{A}\mathbf{y}^{(t)} \ .$$

We always let the column player move second. So, the outcome for her is at least as good as if the row player chose the minimum, that is,

$$(\mathbf{p}^{(t)})^\top \mathbf{A}\mathbf{y}^{(t)} = \max_{\mathbf{y} \in \Delta_n} (\mathbf{p}^{(t)})^\top \mathbf{A}\mathbf{y} \geq \min_{\mathbf{x} \in \Delta_m} \max_{\mathbf{y} \in \Delta_n} \mathbf{x}^\top \mathbf{A}\mathbf{y} \ ,$$

where the equality follows from the definition of $\mathbf{y}^{(t)}$ and the inequality because any possible term is at least the minimum.

Now, we use the regret definition. It says that

$$\text{Regret}^{(T)} = \sum_{t=1}^{T} \sum_{i=1}^{m} p_i^{(t)} \ell_i^{(t)} - \min_i \sum_{t=1}^{T} \ell_i^{(t)} \ .$$

Let us understand the term $\min_i \sum_{t=1}^{T} \ell_i^{(t)}$. We use that the minimum is upper-bounded by any weighted average. Therefore, we have for all $\mathbf{x} \in \Delta_m$

$$\min_i \sum_{t=1}^{T} \ell_i^{(t)} \leq \sum_{i=1}^{m} x_i \sum_{t=1}^{T} \ell_i^{(t)} = \sum_{t=1}^{T} \sum_{i=1}^{m} x_i \ell_i^{(t)} \ .$$

Furthermore, by the definition of $\ell_i^{(t)}$, we have

$$\sum_{i=1}^{m} x_i \ell_i^{(t)} = \mathbf{x}^\top \mathbf{A}\mathbf{y}^{(t)} \ .$$

Note that this holds for all $\mathbf{x}$, so overall

$$\min_i \sum_{t=1}^{T} \ell_i^{(t)} \leq \min_{\mathbf{x} \in \Delta_m} \sum_{t=1}^{T} \mathbf{x}^\top \mathbf{A}\mathbf{y}^{(t)} \ .$$

Furthermore,

$$\min_{\mathbf{x} \in \Delta_m} \sum_{t=1}^{T} \mathbf{x}^\top \mathbf{A}\mathbf{y}^{(t)} = \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A} \left(\sum_{t=1}^{T} \mathbf{y}^{(t)} \right) \leq T \cdot \max_{\mathbf{y} \in \Delta_n} \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A}\mathbf{y} \ .$$

In combination

$$\frac{1}{T} \text{Regret}^{(T)} \geq \min_{\mathbf{x} \in \Delta_m} \max_{\mathbf{y} \in \Delta_n} \mathbf{x}^\top \mathbf{A}\mathbf{y} - \max_{\mathbf{y} \in \Delta_n} \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A}\mathbf{y} \ .$$

If the row player uses a no-regret algorithm, we have $\text{Regret}^{(T)} = o(T)$, that is for every $\epsilon > 0$ there is a T that guarantees $\frac{1}{T}\text{Regret}^{(T)} \leq \epsilon$, this then means

$$\max_{\mathbf{y} \in \Delta_n} \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A} \mathbf{y} \geq \min_{\mathbf{x} \in \Delta_m} \max_{\mathbf{y} \in \Delta_n} \mathbf{x}^\top \mathbf{A} \mathbf{y} - \epsilon \ .$$

As this guarantee holds for all $\epsilon > 0$, it also has to hold for $\epsilon = 0$.

4 Max Flow as a Zero-Sum Game

Zero-sum games also help us to see the Max-Flow Problem and our algorithm which uses an experts algorithm in a new light. Note that if there is a flow of value F^*, there is a way to assign probabilities $(x^*_P)_{P \in \mathcal{P}}$ to the s-t paths such that any edge e fulfills $\sum_{P : e \in P} x^*_P \leq \frac{c_e}{F^*}$.

Inspired by this observation, consider the zero-sum game in which the row player chooses paths and the column player chooses edges. If the row player chooses path P and the column player chooses edge e, the transfer should be $A_{P,e} = \frac{F^*}{c_e}$ if $e \in P$ and 0 otherwise. This way, we have

$$((\mathbf{x}^*)^\top \mathbf{A})_e = \sum_{P \in \mathcal{P}} \mathbf{x}^*_P \mathbf{A}_{P,e} = \sum_{P : e \in P} \mathbf{x}^*_P \frac{F^*}{c_e} \leq 1 \quad \text{for all } e \in E$$

and therefore also

$$(\mathbf{x}^*)^\top \mathbf{A} \mathbf{y} \leq 1 \quad \text{for all } \mathbf{y} \in \Delta_m \ .$$

By these considerations, it follows that

$$\max_{\mathbf{y} \in \Delta_n} \min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top \mathbf{A} \mathbf{y} = 1$$

if and only if there is a flow of value F^*.

Our flow algorithm tries to find exactly this pair $\mathbf{x}, \mathbf{y}$. It does so by using exactly the row player's strategy of an experts algorithm mentioned above.