A Course in Graduate Algorithms **Lecture 15**

Zero-Sum Games

Designed by Prof. Bo Waggoner for the University of Colorado-Boulder *Updated: 2023*

Two-player zero sum games are not only interesting and important in game theory, but also as useful models throughout mathematics and computer science. We will define these games in general and state and prove von Neumann's Min-Max Theorem for finite action spaces, using no-regret learning.[1]

Objectives:

- Define two-player zero-sum games.
- Understand *mixed* (randomized) strategies versus *pure* (non-randomized) strategies.
- Understand the statement of the Min-Max Theorem for mixed strategies and its leader-responder interpretation.
- Know how to use online no-regret learning to compute equilibria in two-player zero-sum games.

1 Two-Player Zero-Sum Games

1.1 Definitions

Mathematically, a *game* is a model of interaction where strategic players select actions, then an outcome occurs as a function of their actions. While they capture board games and similar games played for fun or sport, they can also model situations such as competition among firms, routing decisions in a traffic network, job markets, and more. A **two-player game** is defined by the following:

- Two *action sets*, A for player 1, call her Alice, and B for player 2, call him Bob.
- Two *utility functions*, $u(a, b)$ for Alice and $v(a, b)$ for Bob.

Alice chooses an action $a \in A$ while Bob simultaneously chooses $b \in B$. Alice's goal is to maximize her utility $u(a, b)$, while Bob's goal is to maximize $v(a, b)$. It is a **finite game** if A and B are finite sets.

Zero-sum. In general, the utility functions can be arbitrary, so Alice and Bob are not necessarily opposed. An important special case is when their interests are exactly opposed:

Definition 1. A two-player game (A, B, u, v) is **zero-sum** if, for all $a \in A, b \in B$,

$$u(a, b) = -v(a, b).$$

In this case, we can specify the game by (A, B, u).

Another way to put it is that $u(a, b) + v(a, b) = 0$, i.e. Alice's and Bob's utilities always sum to zero. In a zero-sum game, the better off Alice is, the worse off Bob is.

[1]Notes inspired by those of Aaron Roth at the University of Pennsylvania.

Examples. Rock-paper-scissors can be described as the following game. The action sets are $A = B = \{\text{rock, paper, scissors}\}$. It is a zero-sum game, i.e. $v(a, b) = -u(a, b)$. The utility function u is summarized in the below matrix, where Alice is the *row player* (her strategy is to pick a row) and Bob is the *column player*.

	rock	paper	scissors
rock	0	-1	1
paper	1	0	-1
scissors	-1	1	0

For example, when Alice plays rock and Bob plays scissors, Alice's utility is $u(\text{rock,scissors}) = 1$, given by the top row and third column. The matrix or *normal-form* representation will be a handy way to describe zero-sum games.

Here is another zero-sum game, a *battle of the bands*. Alice knows how to play the Guitar and the Drums, while Bob knows how to play the Violin and the Harmonica. Each of them has to pick one instrument for the battle of the bands. As we can see, Guitar beats Violin by a lot, as Alice gets 3 utility and Bob gets -3. But Violin beats Drums by quite a bit. Harmonica is a bit better than Guitar but worse than Drums.

	Violin	Harmonica
Guitar	3	-1
Drums	-2	1

1.2 Equilibrium

If the game consisted of just one player, Alice, then it would be an *optimization problem*. Alice would pick a to maximize her utility, and we could call this her *optimal* action. However, when playing against Bob, it is not clear what *optimal* means. If Bob plays b_1, then Alice might have one optimal action $a_1 = \text{argmax}_a\, u(a, b_1)$. But if Bob plays b_2, Alice's optimal action (also called a *best response*) could be $a_2 = \text{argmax}_a\, u(a, b_2)$.

Without knowing what Bob will play, Alice's best response is not well-defined. But similarly, without knowing what Alice will play, Bob's best response is not well-defined. Therefore, to analyze the game, we need a *solution concept*. The standard approach in game theory is the idea of an equilibrium: a pair of strategies where each player is already best-responding to the other, and neither could benefit from switching.

Definition 2. In a two-player game (A, B, u, v), a **pure-strategy equilibrium** is a pair $a^* \in A, b^* \in B$ such that:

- $a^* \in \text{argmax}_{a \in A}\, u(a, b^*)$, i.e. Alice's action is a best-response to b^*; and
- $b^* \in \text{argmax}_{b \in B}\, v(a^*, b)$, i.e. Bob's action is a best response to a^*.

We can view an equilibrium as a prediction of what might happen when two players play a game. Or, as a suggestion or guidance for a reasonable way to play the game. As the name suggests, equilibria are related to *stability*: even if opponents do happen to know how the other is playing, they will continue playing the equilibrium strategy. For example, if a game is played repeatedly many times, then an equilibrium is a prediction of a stable point that could be played over and over. If it is not an equilibrium, one of the players should want to switch the strategy next time they play.

Mixed-strategy equilibrium. Many games have no pure-strategy equilibrium. For example, you can check that this is the case for rock-paper-scissors and battle of the bands as defined above (Exercise 1). In such cases, one approach is to suppose that the players can *randomize* over their actions. A probability distribution over actions is also called a *mixed strategy*, as opposed to a *pure strategy* of a single action.

For example, in rock-paper-scissors, it makes sense to pick an action from some distribution so that the opponent doesn't know exactly what we will play.

We use Δ_X to denote the set of probability distributions on X.

Definition 3. Given a two-player game (A, B, u, v), its **mixed extension** is the game $(\Delta_A, \Delta_B, \bar{u}, \bar{v})$ where

- $\bar{u}(p, q) := \mathbb{E}_{a \sim p} \mathbb{E}_{b \sim q} u(a, b)$ and
- $\bar{v}(p, q) := \mathbb{E}_{a \sim p} \mathbb{E}_{b \sim q} v(a, b)$.

From now on, we will drop the bars and just write $u(p, q)$. We may also write $u(a, q) := \mathbb{E}_{b \sim q} u(a, b)$ and $u(p, b) := \mathbb{E}_{a \sim p} u(a, b)$.

Definition 4. In a two-player game (A, B, u, v), a **mixed-strategy equilibrium** is an equilibrium of the mixed extension, i.e. a pair $p^* \in \Delta_A$ and $q^* \in \Delta_B$ such that $p^* \in \text{argmax}_{p \in \Delta_A} u(p, q^*)$ and $q^* \in \text{argmax}_{q \in \Delta_B} v(p^*, q)$.

In a zero-sum game, we can equivalently write Bob's condition as $q^* \in \text{argmin}_{q \in \Delta_B} u(p^*, q)$, i.e. Bob is trying to *minimize* Alice's expected utility.

Generally, when one sees the term *equilibrium* or *Nash equilibrium*, it refers to mixed strategies rather than pure strategies. It is a fact – Nash's theorem – that a mixed-strategy equilibrium always exists for any finite game. Sometimes we can also prove existence of equilibrium in a pure-strategy context where A and B are nice convex sets, such as intervals on the real line. One can think of the mixed extension and turning a discrete finite game into a more nicely-behaved "convex game" so that equilibria always exist. However, we will not discuss more general convex games in this lecture.

Exercise 1. Prove that battle of the bands, as defined above, has no pure-strategy equilibrium.

Hint: Take each pair of actions (a, b) and show that pair is not an equilibrium.

Exercise 2. Verify that $p^* = q^* = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ is a mixed-strategy equilibrium of rock-paper-scissors.

2 The Min-Max Theorem

2.1 Order of play

Two-player zero-sum games have very special properties not shared by other kinds of games. We will see that they have a *value* that Alice can gaurantee herself regardless of what strategy Bob plays, and vice versa. This implies a surprising fact, that in any equilibrium, Alice must obtain that value (and analogously for Bob), i.e. there are not better or worse equilibria. We will also see that the players can commit to strategies simultaneously or one after the other, in either order, and the equilibria are the same. Let us now consider this last property.

Suppose we change the game so that the players go one at a time:

1. Alice has to first pick her mixed strategy $p \in \Delta_A$ and commit to it.
2. Bob learns p and then gets to decide on his mixed strategy $q \in \Delta_B$.
3. As usual, Alice's payoff is $u(p, q)$ and Bob's is $-u(p, q)$.

What is optimal play in this game? Well, for any choice p of Alice, Bob should try to respond with a q that minimizes her utility, i.e. Bob solves $\min_q u(p, q)$. Therefore, Alice should commit to a p that maximizes this utility, i.e. Alice solves

$$\max_p \min_q u(p, q).$$

Now, imagine we switch the game up so that Alice plays second. Bob first commits to q, then Alice learns q and gets to decide on p. In this case, we can use the same argument: for any q, Alice should solve $\max_p u(p,q)$, so Bob should solve

$$\min_q \max_p u(p,q).$$

Which scenario is better for Alice? Intuitively, the second one: she gets to wait and pick p once she has full information about what her opponent is doing. We can formalize this:

Lemma 1. *It is (weakly) better for Alice to play second than first, i.e. in any zero-sum game,*

$$\max_p \min_q u(p,q) \le \min_q \max_p u(p,q).$$

Proof. Let $p^* \in \text{argmax}_p \min_q u(p,q)$ be an optimal strategy for Alice when playing first. Then she can guarantee herself at least as much utility when playing second: she simply ignores what Bob commits to, and plays p^*. Mathematically,

$$\begin{aligned} \max_p \min_q u(p,q) &= \min_q u(p^*,q) \\ &\le \min_q \max_p u(p,q). \end{aligned}$$

□

Now the question is: *how much* better is it to go second? Suprisingly, the Min-Max Theorem will show that it is not better at all. Alice can do exactly as well going first as second. We will show this next.

Exercise 3. Consider the game where Alice plays second. Bob plays some mixed strategy q and Alice solves for an optimal response $p \in \Delta_A$. Argue that Alice has an optimal response that is a pure strategy $a \in A$, or in other words,

$$\min_q \max_{p \in \Delta_A} u(p,q) = \min_q \max_{a \in A} u(a,q).$$

Hint: use that any optimal p is a distribution over some subset of pure strategies $a_1, \dots, a_k$. Recall the definition of $u(p,q)$.

2.2 The Min-Max Theorem

In fact, Alice does *no* better by playing second than she did by playing first.

Theorem 1. *In a finite two-player zero-sum game (A, B, u),*

$$\max_p \min_q u(p,q) = \min_q \max_p u(p,q).$$

Proof. Define

$$\begin{aligned} v_1 &= \max_p \min_q u(p,q) && \text{Alice's utility when playing first,} \\ v_2 &= \min_q \max_p u(p,q) && \text{Alice's utility when playing second.} \end{aligned}$$

In Lemma 1, we showed $v_1 \le v_2$. It remains to show $v_1 \ge v_2$. In fact, we will show that for all $\alpha > 0$, $v_1 \ge v_2 - \alpha$. This implies $v_1 \ge v_2$.

Our proof will use online no-regret learning. Suppose in rounds $t = 1, \dots, T$:

1. Alice picks a mixed strategy p^t.

2. Bob sees p^t and best-responds, choosing strategy $q^t = \text{argmin}_q\, u(p^t, q)$.
3. Alice draws $a \sim p^t$ and Bob draws $b \sim q^t$.
4. Alice's utility is $u(a, b)$. Her expected utility is $u(p^t, q^t)$.

We can cast this as an online learning problem for Alice with finite action set A. Alice will utilize the Multiplicative Weights algorithm (MW). Define

$$\text{Alice} = \frac{1}{T} \sum_{t=1}^{T} u(p^t, q^t) \tag{1}$$

$$\text{OPT} = \max_{a \in A} \frac{1}{T} \sum_{t=1}^{T} u(a, q^t). \tag{2}$$

We can modify the utility so that it satisfies the assumptions for MW, and its regret guarantee gives the following (see Exercise 4):

Lemma 2. *Let $M = 2\max_{a,b} |u(a, b)|$. If Alice plays according to MW, then her average per-round utility satisfies*

$$\text{OPT} - \textit{Alice} \le 2\sqrt{\frac{2M \ln |A|}{T}}.$$

Next, we have the following claims.

Lemma 3. $v_2 \le \text{OPT}$.

Proof. Let $\hat{q} = \frac{1}{T} \sum_{t=1}^{T} q^t$. Notice (Exercise 5) that for any p, $u(p, \hat{q}) = \frac{1}{T} \sum_{t=1}^{T} u(p, q^t)$.

$$\begin{aligned}
v_2 &= \min_q \max_p u(p, q) \\
&\le \max_p u(p, \hat{q}) \\
&= \max_p \frac{1}{T} \sum_{t=1}^{T} u(p, q^t) \\
&= \max_{a \in A} \frac{1}{T} \sum_{t=1}^{T} u(a, q^t) && \text{(Exercise 3)} \\
&= \text{OPT}.
\end{aligned}$$

Intuitively, in v_2, Bob chooses q optimally knowing that Alice then gets to best-respond. In OPT, Bob has chosen some sequence of strategies $q^1, \ldots, q^T$ and we consider the best response of Alice in hindsight. The best Bob could have done is choosing $q^t = q$ for all t. □

Lemma 4. $v_1 \ge$ *Alice.*

Proof. In v_1, Alice chooses p optimally, knowing that Bob will best-respond. In "Alice", she is just choosing a sequence $p^1, \ldots, p^T$ according to MW, and Bob will best-respond to each.

$$\begin{aligned}
\text{Alice} &= \frac{1}{T} \sum_{t=1}^{T} \max_q u(p^t, q) \\
&\le \frac{1}{T} \sum_{t=1}^{T} \min_p \max_q u(p, q) \\
&= v_1.
\end{aligned}$$

□

Now we are ready to complete the proof of the Min-Max Theorem. Let any $\alpha > 0$ be given. We must show that $v_2 - v_1 \le \alpha$. Choose $T = \left\lceil \frac{8M \ln |A|}{\alpha^2} \right\rceil$.

$$
\begin{aligned}
v_2 - v_1 &\le \text{OPT} - v_1 && \text{(Lemma 3)} \\
&\le \text{OPT} - \text{Alice} && \text{(Lemma 4)} \\
&\le 2\sqrt{\frac{2M \ln |A|}{T}} && \text{(Lemma 2)} \\
&\le \alpha.
\end{aligned}
$$

Since this holds for all $\alpha > 0$, we have $v_2 - v_1 \le 0$ and therefore $v_2 \le v_1$ as desired. □

Exercise 4. Prove Lemma 2, using the fact that MW guarantees regret at most $2\sqrt{2T \ln(N)}$ with N actions.

Hint: to apply MW, we should have losses in $[0, 1]$ *rather than utilities. Try letting* $M = \max_{a,b} 2|u(a, b)|$ *and defining* $\ell_a^t = \frac{1}{2} - \frac{u(a,q^t)}{M}$*. Show that* $\ell_a^t \in [0, 1]$*, then use the definition of MW's regret to prove the lemma.*

Exercise 5. Let $\hat{q} = \frac{1}{T} \sum_{t=1}^{T} q^t$. Show that for any p, $u(p, \hat{q}) = \frac{1}{T} \sum_{t=1}^{T} u(p, q^t)$. Notice the left side is interpreted as playing p against the distribution $\hat{q}$, while the right side is the average utility from playing p over T rounds against the strategies $q^1, \dots, q^T$.

3 Corollaries

We can obtain a number of remarkable facts from the Min-Max Theorem and its proof. We will first state the results, then prove them all together.

Corollary 1. *Any two-player zero-sum finite game has a* ***value***

$$v^* := \max_p \min_q u(p, q) = \min_q \max_p u(p, q).$$

In any equilibrium of the game, Alice's expected utility is v^* *and Bob's is* $-v^*$*.*

Corollary 2. *Given a two-player zero-sum finite game, for any* $p^* \in \text{argmax}_p \min_q u(p, q)$ *and any* $q^* \in \text{argmin}_q \max_p u(p, q)$*, the pair* (p^*, q^*) *is an equilibrium.*

Corollary 3. *An* α*-approximate equilibrium of a two-player zero-sum finite game can be computed in time polynomial in* $|A|, |B|, \frac{1}{\alpha}$*. In particular, using the Multiplicative Weights algorithm as in the proof of Theorem 1, the pair* $\hat{p} = \frac{1}{T} \sum_{t=1}^{T} p^t$ *and* $\hat{q} = \frac{1}{T} \sum_{t=1}^{T} q^t$ *form an* α*-approximate equilibrium for a large enough choice of* T*.*

Proof. If Alice plays p^*, then by definition of $\max_p \min_q u(p, q)$, her expected utility is at least v^* regardless of what Bob plays. Therefore, Alice can *guarantee* utility at least v^* by playing p^*. Therefore, in any equilibrium, Alice's utility is at least v^*.

On the other hand, by the same logic, Bob can guarantee himself utility at least $-v^*$ by playing q^*. So in any equilibrium, Alice's utility is at most v^*. Corollary 1 follows.

Next, Corollary 2 follows because in (p^*, q^*), Alice's utility is $v^* = u(p^*, q^*)$. By definition of q^*, there is no strategy Alice can switch to that increases her utility, i.e. $p^* \in \text{argmax}_p u(p, q^*)$. Similarly, $q^* \in \text{argmin}_q u(p^*, q)$. So (p^*, q^*) are an equilibrium.

Finally, we argue that MW actually computes *approximate equilibria*, i.e. that $\min_q u(\hat{p}, q) \ge v^* - \alpha$ (so Alice is approximately best-responding) and $\max_p u(p, \hat{q}) \le v^* + \alpha$ (so Bob is as well). First, we can view $\hat{p}$ as an average utility for playing from each p^t with probability $\frac{1}{T}$, just as in Exercise 5. Second, Alice only does better in this scenario than if Bob gets to respond to each p^t individually:

$$\begin{aligned}\min_q u(\hat{p}, q) &= \min_q \frac{1}{T} \sum_{t=1}^{T} u(p^t, q) \\ &\geq \frac{1}{T} \sum_{t=1}^{T} \min_{q^t} u(p^t, q^t) \\ &= \text{Alice}.\end{aligned}$$

Putting this together with the proof of Theorem 1, and choosing T large enough,

$$\begin{aligned}\min_q u(\hat{p}, q) &\geq \text{Alice} \\ &\geq \text{OPT} - \alpha \\ &\geq v^* - \alpha.\end{aligned}$$

So $\hat{p}$ is an α-best response for Alice. Similarly,

$$\begin{aligned}\max_p u(p, \hat{q}) &= \max_p \frac{1}{T} \sum_{t=1}^{T} u(p, q^t) \\ &= \max_{a \in A} \frac{1}{T} \sum_{t=1}^{T} u(a, q^t) \\ &= \text{OPT} \\ &\leq v^* + \alpha.\end{aligned}$$

So $\hat{q}$ is an α-best response for Bob. □

4 Applications

4.1 Bounds for algorithms

Consider an optimization problem, such as maxizing a matching in a graph. It could be an online problem. For each input size n, there are a set of possible instances B and a set of possible algorithms A. Generally, these are finite (if extremely large) sets if we fix n. The performance of algorithm $a \in A$ on instance $b \in B$ is $u(a, b)$.

For example, in online bipartite matching, we know that the greedy algorithm a^* can achieve $u(a*, b) \geq \frac{1}{2}$ for all instances b, or in other words,

$$\min_b u(a^*, b) \geq \frac{1}{2}.$$

Often, we would like to bound the performance of any *randomized algorithm*. Since we can view a randomized algorithm as a distribution $p \in \Delta_A$, we wish to compute

$$\max_{p \in \Delta_A} \min_{b \in B} u(p, b).$$

By Lemma 1, and the observation of Exercise 3 that the inner minimization can be either over mixed or pure strategies, we get

$$\max_{p \in \Delta_A} \min_{b \in B} u(p, b) \leq \min_{q \in \Delta_B} \max_{a \in A} u(a, q). \tag{3}$$

In this context, Equation 3 is known as Yao's Lemma or Yao's minimax principle. On the left side, we fix a randomized algorithm p and consider its worst-case performance on any input. On the right side, we fix a distribution over inputs q, and consider the best performance of any deterministic algorithm.

Therefore, to upper-bound the best possible performance of any randomzied algorith, we can simply give a distribution q over inputs and consider the best possible determinstic algorithm that knows and responds to that distribution. For example, in online matching, we can consider the "Z" graph where the first arriving left-side vertex is connected to both right-side vertices, and the second arriving left-side vertex is randomly connected to one of the right-side vertices. This distribution is q. The best possible deterministic algorithm is to simply match the first arrival to its first neighbor, and match the second arrival if possible. This gives $\max_a u(a, q) = \frac{3}{4}\text{OPT}$. By Yao's Lemma, the best possible performance guarantee of any randomized algorithm is therefore $\frac{3}{4}$.

A similar argument, extended to general input sizes, shows that no randomized algorithm can exceed $1 - \frac{1}{e} \approx 0.6321\ldots$ for online bipartite matching.

Note that Yao's Lemma (3) sometimes holds more generally that the Min-Max Theorem, e.g. for infinite games. However, often the Min-Max Theorem also applies and (3) is an equality. This is (roughly) the case in online bipartite matching, where there exists a randomized algorithm achieving $1 - \frac{1}{e}$.

4.2 Machine learning

There are many places where zero-sum games arise in machine learning. One example is in *Generative Adversarial Networks* (GANs), which are used to generate images and text. The idea there is to set up a game between a *learner* (Alice) and *discriminator* (Bob). Alice's strategy space A consists of, say, images of faces. Bob's strategy space B consists of classifiers that take in an image and classify it as either real or fabricated.

The game proceeds as follows:

1. Alice produces an image $a \in A$, while Bob simultaneously produces a classifier $b \in B$.
2. We flip a coin.
    - If heads, we give the classifier the fabricated image a.
    - If tails, we give the classifier a real image a' drawn randomly from a dataset.
3. Alice's utility is 1 if Bob's classifier predicts incorrectly, or 0 if Bob's classifier predicts correctly.
4. Bob's utility is the negative of Alice's, i.e. it is a zero-sum game.

It was found that by training algorithms for Alice and Bob simultaneously, the "Alice" or learner algorithm would learn to generate images that are very similar to real images. So by setting up and solving a zero-sum game, we are able to accomplish a seemingly-unrelated machine learning task.