Equilibrium Computation in Normal-form Games: Linear Programming and Online Learning

15 888 **Computational Game Solving** (Fall 2025)

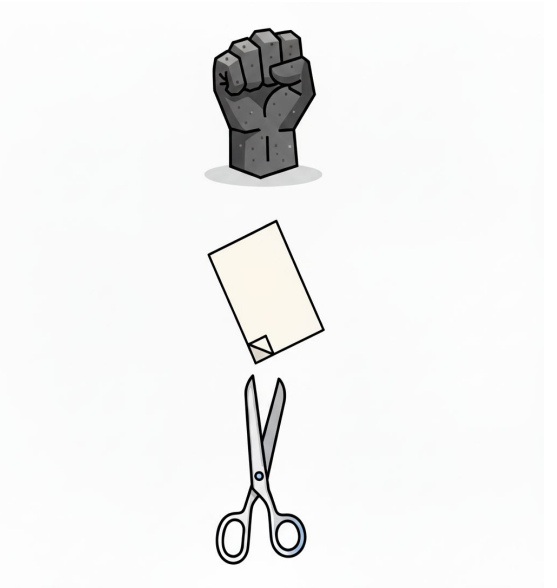
Ioannis Anagnostides

Today’s lecture

- Solving (two-player) zero-sum games through linear programming
  - Polynomial-time algorithms
  - Proof of the minimax theorem
- Simple iterative algorithms for zero-sum games
  - Best-response dynamics
  - Fictitious play
- Online learning and regret minimization
  - Basic learning algorithms (FTRL, MD, RM)
  - Connections to game-theoretic equilibria

Recap on normal-form games

- We think of them as *simultaneous-move* interactions
- Each player selects an action from a finite set
- **Utility function** maps actions to a real value
- Players can randomize by specifying distributions
- Players maximize expected utility
- Any finite game can be represented in normal form
- The normal-form is often inefficient (more on this in the next lecture)



Zero-sum games

- Two players with exactly **opposing** interests
- The row player wants to solve

$$\min_{\boldsymbol{x} \in \Delta(\mathcal{A}_1)} \max_{\boldsymbol{y} \in \Delta(\mathcal{A}_2)} \boldsymbol{x}^\top \mathbf{A} \boldsymbol{y} = \sum_{a_1 \in \mathcal{A}_1} \sum_{a_2 \in \mathcal{A}_2} \boldsymbol{x}[a_1] \mathbf{A}[a_1, a_2] \boldsymbol{y}[a_2].$$

- The row player first specifies a mixed strategy, and then the column player responds optimally to that strategy

Minimax through linear programming

$$\min_{\boldsymbol{x}\in\Delta(\mathcal{A}_1)} \max_{\boldsymbol{y}\in\Delta(\mathcal{A}_2)} \boldsymbol{x}^\top \mathbf{A}\boldsymbol{y} = \sum_{a_1\in\mathcal{A}_1} \sum_{a_2\in\mathcal{A}_2} \boldsymbol{x}[a_1]\mathbf{A}[a_1, a_2]\boldsymbol{y}[a_2].$$

- Equivalent to $\min_{\boldsymbol{x}\in\Delta(\mathcal{A}_1)} \max_{a_2\in\mathcal{A}_2} \boldsymbol{x}^\top \mathbf{A}[:, a_2]$

- The column player can always respond optimally through a *pure strategy*
- What if we introduce an auxiliary variable? $t \geq \boldsymbol{x}^\top \mathbf{A}[:, a_2]$ for all $a_2 \in \mathcal{A}_2$

Linear programming formulation

LP for row player

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & t \geq \boldsymbol{x}^\top \mathbf{A}[:, a_2] \text{ for all } a_2 \in \mathcal{A}_2, \\
& \sum_{a_1 \in \mathcal{A}_1} \boldsymbol{x}[a_1] = 1, \\
& \boldsymbol{x} \geq 0.
\end{aligned}
$$

Linear programming formulation

LP for row player

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & t \geq \boldsymbol{x}^\top \mathrm{A}[:, a_2] \text{ for all } a_2 \in \mathcal{A}_2, \\
& \sum_{a_1 \in \mathcal{A}_1} \boldsymbol{x}[a_1] = 1, \\
& \boldsymbol{x} \geq 0.
\end{aligned}
$$

LP for column player

$$
\begin{aligned}
\text{maximize} \quad & t \\
\text{subject to} \quad & t \leq \boldsymbol{y}^\top \mathrm{A}[a_1, :] \text{ for all } a_1 \in \mathcal{A}_1, \\
& \sum_{a_2 \in \mathcal{A}_2} \boldsymbol{y}[a_2] = 1, \\
& \boldsymbol{y} \geq 0.
\end{aligned}
$$

Linear programming formulation

LP for row player

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & t \geq \boldsymbol{x}^\top \mathbf{A}[:, a_2] \text{ for all } a_2 \in \mathcal{A}_2, \\
& \sum_{a_1 \in \mathcal{A}_1} \boldsymbol{x}[a_1] = 1, \\
& \boldsymbol{x} \geq 0.
\end{aligned}
$$

LP for column player

$$
\begin{aligned}
\text{maximize} \quad & t \\
\text{subject to} \quad & t \leq \boldsymbol{y}^\top \mathbf{A}[a_1, :] \text{ for all } a_1 \in \mathcal{A}_1, \\
& \sum_{a_2 \in \mathcal{A}_2} \boldsymbol{y}[a_2] = 1, \\
& \boldsymbol{y} \geq 0.
\end{aligned}
$$

These are duals!

Proof of the minimax theorem

Linear programming duality implies

$$\min_{\boldsymbol{x}\in\Delta(\mathcal{A}_1)} \max_{\boldsymbol{y}\in\Delta(\mathcal{A}_2)} \boldsymbol{x}^\top \mathbf{A}\boldsymbol{y} = \max_{\boldsymbol{y}\in\Delta(\mathcal{A}_2)} \min_{\boldsymbol{x}\in\Delta(\mathcal{A}_1)} \boldsymbol{x}^\top \mathbf{A}\boldsymbol{y} = v.$$

Theorem. There is a polynomial-time algorithm for finding minimax (or Nash) equilibria in zero-sum games.

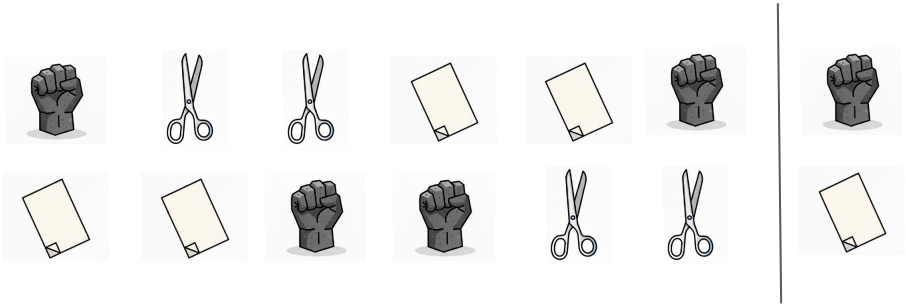
Are we done? Unfortunately, LP solvers scale poorly in large games

Iterative algorithms: best-response dynamics

- Simple idea: best-respond to each other’s strategy
- Does it converge?

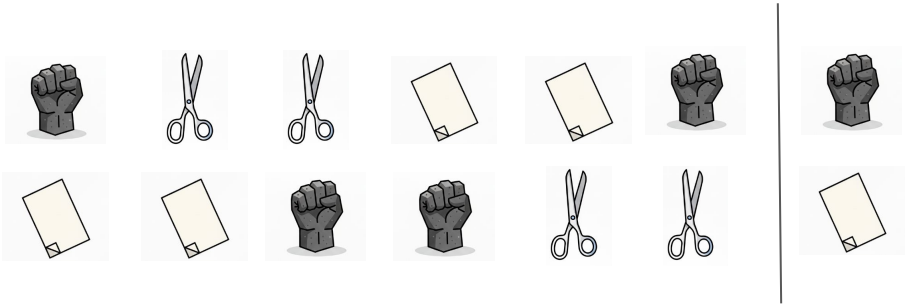
Iterative algorithms: best-response dynamics

- Simple idea: best-respond to each other’s strategy
- Does it converge?
- Some games don’t even have pure-strategy equilibria



Iterative algorithms: best-response dynamics

- Simple idea: best-respond to each other’s strategy
- Does it converge?
- Some games don’t even have pure-strategy equilibria
- What about the average?
- What if winning by paper is twice more valuable?



Iterative algorithms: fictitious play

- A more sophisticated algorithm is **fictitious play**
- It best-responds not to the previous strategy, but to the **average strategy** of the opponent so far
- Basic opponent modeling
- Julia Robinson showed it always converges
- But the rate of convergence can be exponentially slow (Daskalakis and Pan, 2014)
- Open question beyond adversarial tiebreaks

Iterative algorithms: fictitious play

- A more sophisticated algorithm is **fictitious play**
- It best-responds not to the previous strategy, but to the **average strategy** of the opponent so far
- Basic opponent modeling
- Julia Robinson showed it always converges
- But the rate of convergence can be exponentially slow (Daskalakis and Pan, 2014)
- Open question beyond adversarial tiebreaks
- If our opponent knows we are using fictitious play, are we **exploitable**?

Online learning

- A learner interacts with an environment over a sequence of rounds
- The learner first selects a mixed strategy
- The environment provides as feedback some utility vector $\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle$

Online learning

- A learner interacts with an environment over a sequence of rounds
- The learner first selects a mixed strategy
- The environment provides as feedback some utility vector $\langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle$

Regret is the most common measure of performance

$$\mathrm{Reg}^{(T)} \coloneqq \max_{\boldsymbol{x} \in \Delta(\mathcal{A})} \left\{ \sum_{t=1}^{T} \langle \boldsymbol{x}, \boldsymbol{u}^{(t)} \rangle \right\} - \sum_{t=1}^{T} \langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle.$$

☞ *Regret can be negative!*

0 1 0 1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0 1 0 1

Is fictitious play no-regret?

$$\boldsymbol{x}^{(t)} \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta^m} \sum_{\tau=1}^{t-1} \langle \boldsymbol{x}, \boldsymbol{u}^{(\tau)} \rangle.$$

Aka. “follow the leader”

What happens if fictitious play encounters the following sequence of utilities?

ε 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0

Is fictitious play no-regret?

$$\boldsymbol{x}^{(t)} \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta^m} \sum_{\tau=1}^{t-1} \langle \boldsymbol{x}, \boldsymbol{u}^{(\tau)} \rangle.$$

Aka. “follow the leader”

What happens if fictitious play encounters the following sequence of utilities?

ε 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0

Theorem. There is a sequence of utilities such that fictitious play incurs linear regret.

Follow the regularized leader (FTRL)

A small tweak to fictitious play:

$$\boldsymbol{x}^{(t)} = \operatorname*{argmax}_{\boldsymbol{x} \in \Delta^m} \left\{ \sum_{\tau=1}^{t-1} \langle \boldsymbol{x}, \boldsymbol{u}^{(\tau)} \rangle - \frac{1}{\eta} \mathcal{R}(\boldsymbol{x}) \right\}$$

- $\mathcal{R}$ is a *strictly convex* regularizer
- η is the learning rate

Multiplicative weights update:

Let $\mathcal{R} : \boldsymbol{x} \mapsto \sum_{a \in \mathcal{A}} \boldsymbol{x}[a] \ln \boldsymbol{x}[a]$

$$\boldsymbol{x}^{(t)}[a] \propto \exp\left(\eta \sum_{\tau=1}^{t-1} \boldsymbol{u}^{(\tau)}[a]\right)$$

Euclidean regularization:

Let $\mathcal{R} : \boldsymbol{x} \mapsto \frac{1}{2} \sum_{a \in \mathcal{A}} (\boldsymbol{x}[a])^2 = \frac{1}{2} \|\boldsymbol{x}\|_2^2$

$$\boldsymbol{x}^{(t)} = \Pi_{\Delta(\mathcal{A})}\left(\eta \sum_{\tau=1}^{t-1} \boldsymbol{u}^{(\tau)}\right)$$

FTRL has no-regret

Theorem. Under any sequence of utilities, the regret of FTRL is bounded as

$$\mathrm{Reg}^{(T)} \leq \frac{R}{\eta} + \eta \sum_{t=1}^{T} \|\boldsymbol{u}^{(t)}\|_*^2.$$

- R is the range of the regularizer, *strongly convex* w.r.t. $\|\cdot\|$
- $\|\cdot\|_*$ is the *dual* norm

By selecting the learning rate optimally,

$$\mathrm{Reg}^{(T)} \leq 2B\sqrt{RT}.$$

For MWU,

$$\mathrm{Reg}^{(T)} \leq 2\sqrt{T \ln m}.$$

Online mirror descent

Another class of online algorithms:

$$\boldsymbol{x}^{(t)} \coloneqq \operatorname*{argmax}_{\boldsymbol{x}\in\Delta(\mathcal{A})} \left\{ \langle \boldsymbol{x}, \boldsymbol{u}^{(t-1)} \rangle - \frac{1}{\eta} \mathcal{B}_{\mathcal{R}}(\boldsymbol{x}, \boldsymbol{x}^{(t-1)}) \right\}.$$

Online gradient descent:

$$\boldsymbol{x}^{(t)} = \Pi_{\Delta(\mathcal{A})}(\boldsymbol{x}^{(t-1)} + \eta \boldsymbol{u}^{(t-1)}).$$

We measure distance through the Bregman divergence:

$$\mathcal{B}_{\mathcal{R}}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \mathcal{R}(\boldsymbol{x}) - \mathcal{R}(\boldsymbol{x}') - \langle \nabla \mathcal{R}(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle$$

☞ For some regularizers, FTRL is equivalent to MD

☞ The regret bound of MD is similar to that of FTRL

Regret matching

- MWU can be expressed as

$$\boldsymbol{x}^{(t)}[a] \propto \exp\left(\eta \sum_{\tau=1}^{t-1}(\boldsymbol{u}^{(\tau)}[a] - \langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle)\right) = \exp(\eta \boldsymbol{r}^{(t-1)}[a])$$

- FTRL with Euclidean regularization can be expressed as

$$\boldsymbol{x}^{(t)} := \Pi_{\Delta(\mathcal{A})}(\eta \boldsymbol{r}^{(t-1)})$$

- What about $\boldsymbol{x}^{(t)} \propto \max(\boldsymbol{r}^{(t-1)}, \boldsymbol{0}) := [\boldsymbol{r}^{(t-1)}]^+$

Regret matching

Algorithm 1: Regret matching (RM)

Initialize cumulative regrets $\boldsymbol{r}^{(0)} \coloneqq \mathbf{0}$;
for $t = 1, \dots, T$ **do**
  Define $\boldsymbol{\theta}^{(t)} \coloneqq \max(\boldsymbol{r}^{(t-1)}, \mathbf{0})$;
  if $\boldsymbol{\theta}^{(t)} = \mathbf{0}$ **then**
    Let $\boldsymbol{x}^{(t)} \in \Delta(\mathcal{A})$ be arbitrary
  else
    Compute $\boldsymbol{x}^{(t)} \coloneqq \boldsymbol{\theta}^{(t)} / \|\boldsymbol{\theta}^{(t)}\|_1$;
  Output strategy $\boldsymbol{x}^{(t)} \in \Delta(\mathcal{A})$;
  Observe utility $\boldsymbol{u}^{(t)} \in [0, 1]^{\mathcal{A}}$;
  $\boldsymbol{r}^{(t)} \coloneqq \boldsymbol{r}^{(t-1)} + \boldsymbol{u}^{(t)} - \langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle \mathbf{1}$;

Algorithm 2: Regret matching$^+$ (RM$^+$)

Initialize cumulative regrets $\boldsymbol{r}^{(0)} \coloneqq \mathbf{0}$;
for $t = 1, \dots, T$ **do**
  Define $\boldsymbol{\theta}^{(t)} \coloneqq \boldsymbol{r}^{(t-1)}$;
  if $\boldsymbol{\theta}^{(t)} = \mathbf{0}$ **then**
    Let $\boldsymbol{x}^{(t)} \in \Delta(\mathcal{A})$ be arbitrary
  else
    Compute $\boldsymbol{x}^{(t)} \coloneqq \boldsymbol{\theta}^{(t)} / \|\boldsymbol{\theta}^{(t)}\|_1$;
  Output strategy $\boldsymbol{x}^{(t)} \in \Delta(\mathcal{A})$;
  Observe utility $\boldsymbol{u}^{(t)} \in [0, 1]^{\mathcal{A}}$;
  $\boldsymbol{r}^{(t)} \coloneqq [\boldsymbol{r}^{(t-1)} + \boldsymbol{u}^{(t)} - \langle \boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)} \rangle \mathbf{1}]^+$;

Analysis of RM

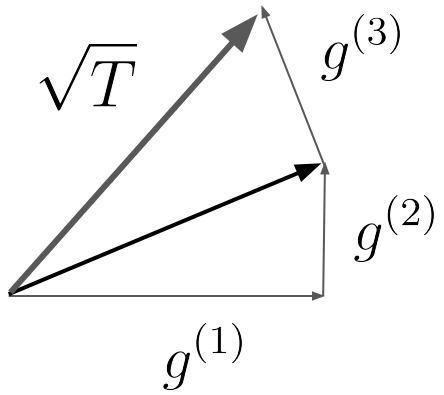
Theorem. For any sequence of utilities, the regret of RM is at most $\sqrt{mT}$.

Instantaneous regret: $g^{(t)} = u^{(t)} - \langle x^{(t)}, u^{(t)} \rangle \mathbf{1}$

Instantaneous regret is always **orthogonal** to the regret accumulated thus far

$$\langle x^{(t)}, g^{(t)} \rangle = 0 = \langle [r^{(t-1)}]^+, g^{(t)} \rangle$$

Pythagorean lemma:



Self-play in zero-sum games

- Both players follow a no-regret algorithm
- This is called “self-play”

Theorem. If both players have no-regret, the **average strategies** converge to a minimax equilibrium.

$$\mathrm{Reg}_1^{(T)} = \max_{\boldsymbol{x}' \in \Delta(\mathcal{A}_1)} \sum_{t=1}^{T} \langle \boldsymbol{x}' - \boldsymbol{x}^{(t)}, -\mathbf{A}\boldsymbol{y}^{(t)} \rangle = \sum_{t=1}^{T} \langle \boldsymbol{x}^{(t)}, \mathbf{A}\boldsymbol{y}^{(t)} \rangle - T \min_{\boldsymbol{x}' \in \Delta(\mathcal{A}_1)} \langle \boldsymbol{x}', \mathbf{A}\bar{\boldsymbol{y}}^{(T)} \rangle.$$

Similarly,

$$\mathrm{Reg}_2^{(T)} = \max_{\boldsymbol{y}' \in \Delta(\mathcal{A}_2)} \sum_{t=1}^{T} \langle \boldsymbol{y}' - \boldsymbol{y}^{(t)}, \mathbf{A}^\top \boldsymbol{x}^{(t)} \rangle = T \max_{\boldsymbol{y}' \in \Delta(\mathcal{A}_2)} \langle \boldsymbol{y}', \mathbf{A}^\top \bar{\boldsymbol{x}}^{(T)} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{x}^{(t)}, \mathbf{A}\boldsymbol{y}^{(t)} \rangle.$$

Self-play in general-sum games

Definition 4.10 (Coarse correlated equilibrium). A correlated distribution $\boldsymbol{\mu} \in \Delta(\mathcal{A}_1 \times \cdots \times \mathcal{A}_n)$ is an *ϵ-coarse correlated equilibrium* if for any player $i \in [n]$ and deviation $a_i' \in \mathcal{A}_i$,

$$\mathbb{E}_{(a_1,\ldots,a_n)\sim\mu} u_i(a_1, \ldots, a_n) \geq \mathbb{E}_{(a_1,\ldots,a_n)\sim\mu} u_i(a_i', a_{-i}) - \epsilon.$$

Theorem. If all players follow a no-regret algorithm, the average correlated distribution of play converges to a coarse correlated equilibrium.