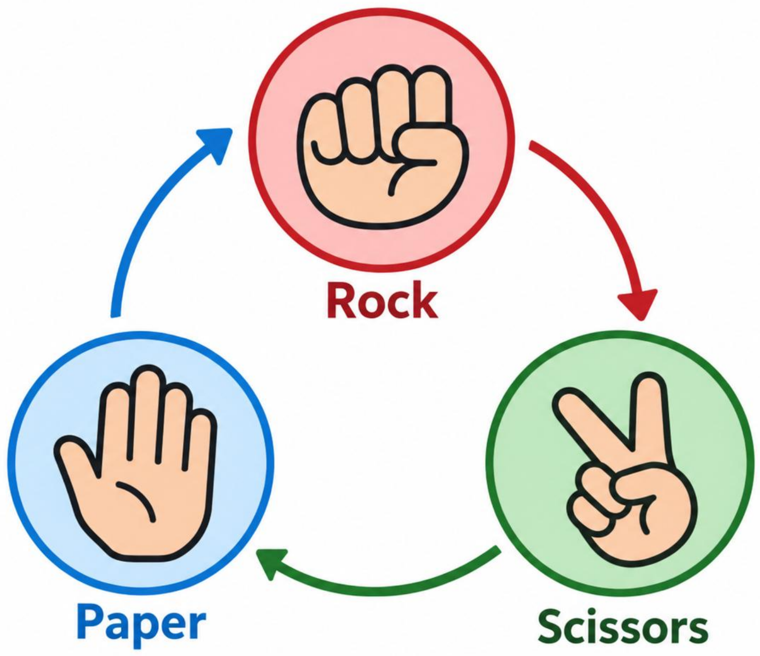
# Two-Player Zero-Sum Games

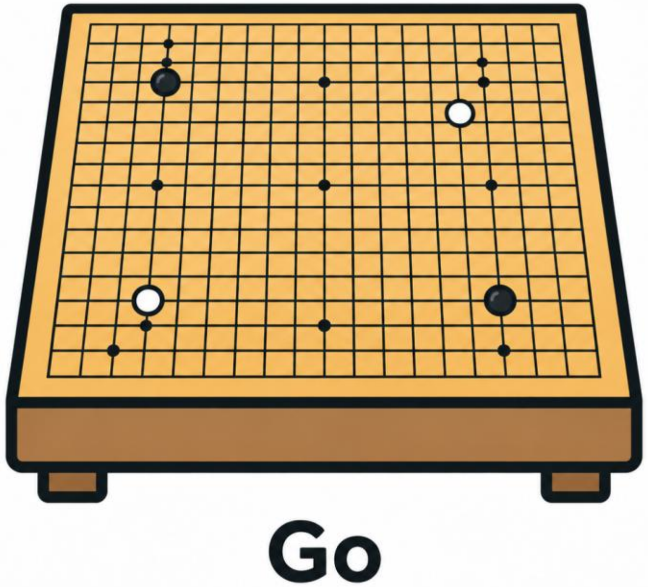
Chen-Yu Wei

# References

- Lecture 4 of Haipeng’s course
- Lecture 8 here (for boosting)

# Two-Player Zero-Sum Games



# Two-Player Zero-Sum Games

$$f(x, y)$$

Min-player

$x \in \mathcal{X}$

Max-player

$y \in \mathcal{Y}$

$\mathcal{X} = \{$ Rock, paper, scissor $\}$

$\mathcal{X} = \Delta_3$

# Game Matrix

$$G = \begin{bmatrix} 1 & -1 & 0 & -3 \\ 2 & 3 & 1 & -5 \end{bmatrix}$$

cost matrix for row-player

Column player

$G =$ row player

| |  |  |  | |
|---|---|---|---|---|
|  | 0 | ~~1~~ 0.99 | -1 | → 0.99 |
|  | -1 | 0 | 1 | → 1 |
|  | 1 | -1 | 0 | → 1 |

$$\sum_{i=1}^{3}\sum_{j=1}^{3} p(i)\,q(j)\,G(i,j) = p^T G q$$

$$\boxed{\begin{array}{l} p \in \Delta_N,\ q \in \Delta_M \\ G \in \mathbb{R}^{N\times M} \end{array}}$$

If row-player uses $p \in \Delta_3$, column player uses $q \in \Delta_3$

# Discussion (Sep 16)

Given a game matrix, (assuming the opponent will do their best), what is the best we can do?

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

# Compare the Two Values

$\max_{y} f(x,y)$

min     max

$$\boxed{\min_x \underbrace{\max_y f(x,y)}_{}} \geq \boxed{\max_y \underbrace{\min_x f(x,y)}_{}}$$

$\leq$

$=$

depend on f

$g(x)$ = my cost if I choose $x$

$h(y)$

Protocol: min-player has to tell max-player their decision

min :( max :)

If we (min-player) have to play under this protocol,

then by choosing

$$x^* = \arg\min_x g(x) = \arg\min_x \left(\max_y f(x,y)\right)$$

my cost is at most

$$\min_x \max_y f(x,y)$$

**Proof.**

$$\text{my cost} = f(x^*, \hat{y})$$

$$\leq \max_y f(x^*, y)$$

$$= \min_x \max_y f(x,y)$$

min :) max :(

Max has to tell min their decision

their reward $\geq \max_y \min_x f(x,y)$

# Compare the Two Values

let $\boxed{x^* = \arg\min_{x} \left( \max_{y} f(x,y) \right)}$

$$\min_{x \in \mathcal{X}} \left( \max_{y \in \mathcal{Y}} f(x,y) \right) = \max_{y \in \mathcal{Y}} \left( f(x^*, y) \right)$$

$$\geq \max_{y \in \mathcal{Y}} \left( \min_{x \in \mathcal{X}} f(x,y) \right)$$

https://www.youtube.com/watch?v=RmgLPeASjXs Minimax Theorem Informal Illustration
(but this is actually not the minimax theorem…)

# Compare the Two Values

Sometimes this ineq is strict:

① $\mathcal{X} = \{\text{rock}, \text{paper}, \text{scissor}\} = \mathcal{Y}$

② $f(x,y) = |x-y|$, $\mathcal{X} = \mathcal{Y} = [0,1]$

Convex in $x$ & $y$

0 $x$ 1

Game values when both players do their best:

$$\min_x \max_y \; f(x,y) \quad \geq \quad \max_y \min_x f(x,y)$$

**Protocol:** min-player has to reveal their decision ($x$) to max-player

$$\min_{x\in[0,1]} \left( \max_{y\in[0,1]} |x-y| \right) = \min_{x\in[0,1]} \left( \underbrace{\max\{x, 1-x\}}_{0.5} \right) = 0.5$$

**Protocol:** max-player has to reveal their decision ($y$) to min-player

$$\max_{y\in[0,1]} \left( \underline{\min_{x\in[0,1]} |x-y|} \right) = \max_{y\in[0,1]} (0) = 0$$

# Minimax Theorem $\boxed{\mathcal{X} = \mathcal{Y} = \Delta_3}$

$$N \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}_{M}$$

**(John von Neumann's) Minimax Theorem (1928).**

Let $G \in \mathbb{R}^{N \times M}$ . Then

$$\min_{p \in \Delta_N} \max_{q \in \Delta_M} p^\top G q \overset{\geq}{\underset{\leq}{=}} \max_{q \in \Delta_M} \min_{p \in \Delta_N} p^\top G q$$

$$\boxed{\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x,y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x,y)}$$

if $\mathcal{X}, \mathcal{Y}$ are bounded convex sets, and $f(x,y)$ is convex in $x$, concave in $y$

$\forall y,\ f(\underset{\sim}{x}, y)$ is a convex func of $x$

# Discussion

Given a game matrix, (assuming the opponent will do their best), what is the best we can do?

$$p^* = \underset{p}{\text{argmin}} \left( \max_{\xi} \; p^T G \xi \right) \quad = \quad \boxed{\max_{\xi} \left( \min_{p} \; p^T G \xi \right)}$$

$$= \underset{p}{\text{argmin}} \left( \max_{j} \; (p^T G)_j \right) \qquad \boxed{\max_{\xi} \left( \min_{i} \; (G\xi)_i \right)}$$

# Proof by running algorithm

$\Delta_N$ : space of distribution over $N$ actions

It suffices to prove

$$\boxed{\min_{p\in\Delta_N} \max_{\xi\in\Delta_M} p^T G \xi \le \max_{\xi\in\Delta_M} \min_{p\in\Delta_N} p^T G \xi}$$

① Create an expert-learning problem

② run no_regret algonit

③ get regret guarantee →

$$p^T G \xi = \sum_{i=1}^{N} \sum_{j=1}^{M} p(i)\xi(j)\, G(i,j)$$

$$= \sum_{i=1}^{N} p(i) \left( \sum_{j=1}^{M} \xi(j)\, G(i,j) \right)$$

$(G\xi)_i$

$p$ For $t=1,2,\cdots,T$:

Generate $p_t \in \Delta_N$

Define loss $\ell_t = G\xi_t$ $\quad$ (loss $= p_t^T G \xi_t$)

$\xi$ For $t=1,2,\cdots,T$

Generate $\xi_t \in \Delta_M$

Define reward $r_t = G^T p_t$

$\langle \xi_t, r_t \rangle = p_t^T G \xi_t$

Regret for min-player

use exponential weights

$$\sum_{t=1}^{T} p_t^T \underbrace{\ell_t}_{= G q_t} - \min_p \sum_{t=1}^{T} p^T \ell_t \le 2\sqrt{T \log N}$$

$$\max_q \sum_{t=1}^{T} q^T \underbrace{r_t}_{G^T p_t} - \sum_{t=1}^{T} q_t^T r_t \le 2\sqrt{T \log M}$$

$$\Rightarrow \sum_{t=1}^{T} p_t^T G q_t - \min_p \sum_{t=1}^{T} p^T G q_t \le 2\sqrt{T \log N}$$

$$\Downarrow$$

$$\max_q \sum_{t=1}^{T} p_t^T G q - \sum_{t=1}^{T} p_t^T G q_t \le 2\sqrt{T \log M}$$

$+$

$$\Downarrow$$

$$\boxed{\max_q \sum_{t=1}^{T} p_t^T G q - \min_p \sum_{t=1}^{T} p^T G q_t \le 2\sqrt{T \log N} + 2\sqrt{T \log M}}$$

$$\mathrm{Reg} = o(T)$$

$$\max_q \frac{1}{T}\sum_{t=1}^{T} p_t^T G q - \min_p \frac{1}{T}\sum_{t=1}^{T} p^T G q_t \le \frac{2\sqrt{T\log N} + 2\sqrt{T\log M}}{T}$$

$$\bar{p} = \frac{1}{T}\sum_{t=1}^{T} p_t \qquad \bar{q} = \frac{1}{T}\sum_{t=1}^{T} q_t$$

Contradiction:

If $\min\max - \max\min = V_G > 0$

then let $T$ large enough so that

$$\sqrt{\frac{\log M}{T}} + \sqrt{\frac{\log N}{T}} < V_G$$

$$\Rightarrow \max_q \bar{p}^T G q - \min_p p^T G \bar{q} \le O\left(\sqrt{\frac{\log N}{T}} + \sqrt{\frac{\log M}{T}}\right)$$

we know $\max_q \bar{p}^T G q \ge \min_p \max_q p^T G q$. $\quad \min_p p^T G \bar{q} \le \max_q \min_p p^T G q$

$$\Rightarrow \min_p \max_q p^T G q - \max_q \min_p p^T G q \le O\left(\sqrt{\frac{\log N}{T}} + \sqrt{\frac{\log M}{T}}\right)$$

$\le 0$?

$$\frac{\mathrm{Reg}}{T} = \frac{o(T)}{T} \to 0$$

# Nash Equilibrium

**Def** Consider game matrix $G \in \mathbb{R}^{N\times M}$

($\varepsilon$-approximate Nash equilibrium)

We call $(p_*, q_*) \in \Delta_N \times \Delta_M$ a Nash Equilibrium iff

$$\begin{cases} ① & p_*^T G q_* \le p^T G q_* + \varepsilon \quad \forall p \in \Delta_N \\ ② & p_*^T G q_* \ge p_*^T G q - \varepsilon \quad \forall q \in \Delta_M \end{cases}$$

$p_* G q \le \max_q p_*^T G q = p_*^T G q_* = \min_p p^T G q_* \le p^T G q_*$

**Thm.** Let $p_* \in \arg\min_p \left( \max_q p^T G q \right)$ $\quad q_* \in \arg\max_q \min_p p^T G q$

Then $(p_*, q_*)$ is a NE

**pf.** We want to show $p_*^T G q \le p_*^T G q_* \le p^T G q_* \quad \forall p, q$

And know $\max_q \min_p p^T G q \le \min_p p^T G q_* \le p_*^T G q_* \le \max_q p_*^T G q = \min_p \max_q p^T G q$

By the choice of $p_*$

# Nash Equilibrium

$\bar{p}, \bar{\xi}$ satisfies

$$\max_{\xi} \bar{p}^T G \xi \le \min_{p} p^T G \bar{\xi} + \overbrace{\sqrt{\frac{\log N}{T}} + \sqrt{\frac{\log M}{T}}}^{o(T)}$$

$$\max_{\xi} \bar{p}^T G \xi \le \min_{p} p^T G \bar{\xi} + o(T) \le \bar{p}^T G \bar{\xi} + o(T)$$

$$\bar{p}^T G \bar{\xi} \le$$

$$\Rightarrow \begin{cases} \bar{p}^T G \bar{\xi} \ge \max_{\xi} \bar{p}^T G \xi - o(T) \qquad \sqrt{\frac{\log(MN)}{T}} \le \varepsilon \Rightarrow T \ge \frac{\log MN}{\varepsilon^2} \\ \bar{p}^T G \bar{\xi} \le \min_{p} p^T G \bar{\xi} + o(T) \end{cases}$$

# Another Proof of the Minimax Theorem

For $t=1,2,\ldots,T$

Choose $p_t \in \underset{p}{\operatorname{argmin}}\ p^T G \xi_t$

$\left( i_t \in \underset{i}{\operatorname{argmin}}\ (G\xi_t)_i \right)$

For $t=1,2,\cdots,T$:

① Generate $\xi_t \in \Delta_M$

② Define reward vector

$r_t = G^T p_t$

③ Update $\xi_t \to \xi_{t+1}$ using $r_t$

Useful when

Minimizer has huge # actions

but, somehow, obtaining

$\underset{p}{\operatorname{argmin}}\ p^T G \xi_t$

is easy

---

Regret of max-player: $\max_{\xi} \sum_{t=1}^{T} p_t^T G \xi - \sum_{t=1}^{T} p_t^T G \xi_t \leq o(T) = \sqrt{\frac{\log M}{T}} \Rightarrow T \geq \frac{\log M}{\varepsilon^2}$

$\sum_{t=1}^{T} \left( \min_{p} p^T G \xi_t \right)$ (by the choice of $p_t$)

$\leq \min_{p} \sum_{t=1}^{T} p^T G \xi_t$

$\max_{\xi} \sum_{t=1}^{T} p_t G \xi - \min_{p} \sum_{t=1}^{T} p^T G \xi_t \leq$

Follow the same proof as before to show

$(\bar{p}, \bar{\xi})$ is approx-NE

# Application: Boosting

**Setting:** Binary classification

**Question:** Suppose we can always find a classifier that is only slightly better than random (i.e., 50% accuracy). Can we use this **weak learner** to produce an accurate classifier?

# Setting

**Training data:** $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N) \in \mathcal{F} \times \{-1,1\}$

**Classifier:** $h\colon \mathcal{F} \to \{-1,1\}$

Loss of a classifier $h$ on a training sample $(x, y)$:

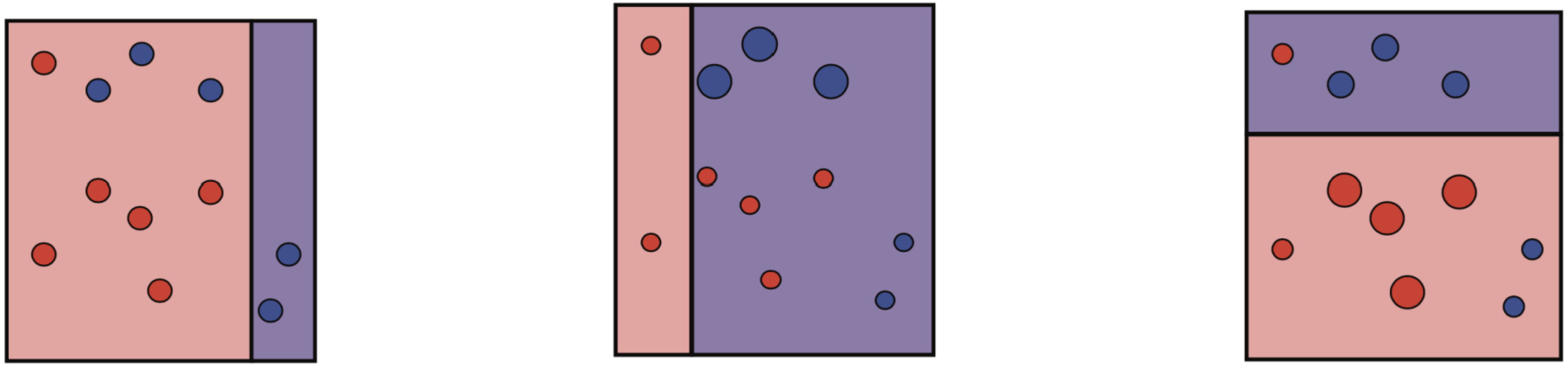
$$\ell((x,y), h) = \mathbb{I}\{h(x) \neq y\}$$

# What Does "Weak Learner" Mean?

Assume : $\forall p \in \Delta_N$, $\exists h \in \mathcal{H}$, such that

$$\sum_{i=1}^{N} p(i) \underbrace{\mathbb{1}\left(h(x_i) \neq y_i\right)}_{\ell((x_i, y_i), h)} \leq \frac{1}{2} - \gamma$$

# What Does “Weak Learner” Mean?



# Applying Minimax Theorem

$\ell((x_i, y_i), h)$

$\|$

$$\max_{p \in \Delta_N} \left( \min_{h \in \mathcal{H}} \sum_{i=1}^{N} p(i)\, \ell(i, h) \right) \le \frac{1}{2} - \gamma$$

$\forall p \quad \exists h$

$$= \max_{p \in \Delta_N} \left( \min_{q \in \Delta_H} \sum_{i=1}^{N} \sum_{h \in H} p(i)\, q(h)\, \ell(i, h) \right) \le \frac{1}{2} - \gamma$$

# From No-Regret Learning to Boosting

**Input:** Training dataset $(x_1, y_1), (x_2, y_2), \ldots., (x_N, y_N)$

Initialize a no-regret algorithm **ALG** over $\Delta_N$

For $t = 1, \ldots, T$:

Let **ALG** output $p_t \in \Delta_N$

Train a classifier $h_t = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \; \mathbb{E}_{i \sim p_t}\big[1[h(x_i) \neq y_i]\big]$

Define **reward** $r_t(i) = 1[h_t(x_i) \neq y_i]$.

Feed reward vector $r_t$ to **ALG**.

# Boosting with Exponential Weights

**Input:** Training dataset $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$

For $t = 1, \dots, T$:

Train a classifier $h_t = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \ \mathbb{E}_{i \sim p_t}\left[1[h(x_i) \neq y_i]\right]$

Update $p_{t+1}(i) \propto p_t(i) \exp(\eta \, 1[h_t(x_i) \neq y_i])$

# Adaboost

(Yoav Freund and Robert Schapire, 1995)

**Input:** Training dataset $(x_1, y_1), (x_2, y_2), \dots., (x_N, y_N)$

For $t = 1, \dots, T$:

Train a classifier $h_t = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \; \mathbb{E}_{i \sim p_t}[1[h(x_i) \neq y_i]]$ Let $\epsilon_t = \mathbb{E}_{i \sim p_t}[1[h_t(x_i) \neq y_i]]$

Update $p_{t+1}(i) \propto p_t(i) \exp(\alpha_t \, 1[h_t(x_i) \neq y_i])$ with $\alpha_t = \ln \frac{1-\epsilon_t}{\epsilon_t}$

# Example

Link