6.S890: Topics in Multiagent Learning

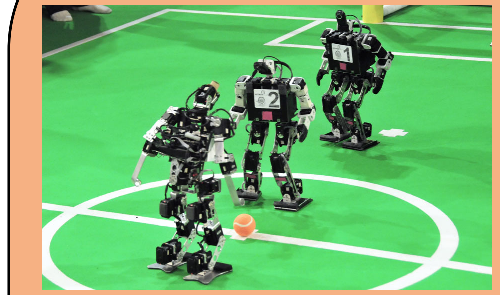
Lecture 20

Fall 2023

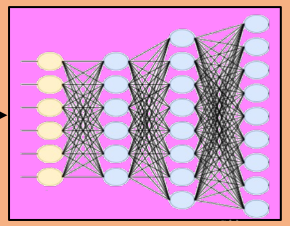
Context: Increasing Interest in Multi-Agent Learning



Multi-player Game-Playing:
- Superhuman Chess, Go, Poker, Gran Turismo
- Good StarCraft, Diplomacy

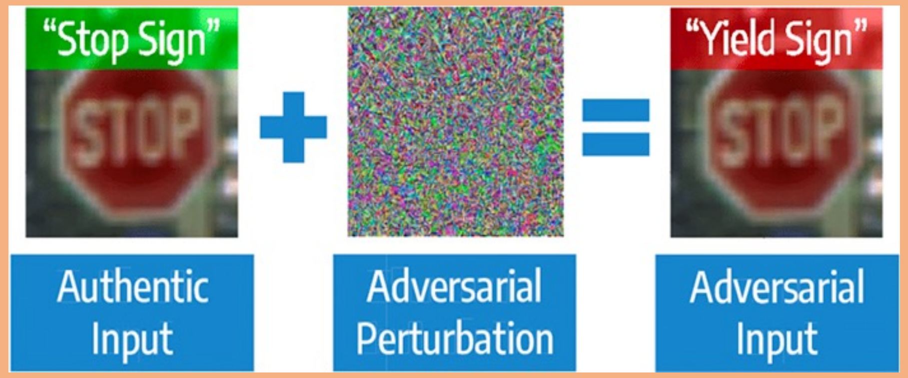


- Multi-robot interactions
- Autonomous driving
- Automated Economic policy design

$Z \sim \mathcal{N}(0, I) \longrightarrow$  $\longrightarrow$  $\sim P_{\text{target}}$

boring randomness | neural network | interesting randomness

Generative Adversarial Networks (GANs)
synthetic data generation



Adversarial Training
robustifying models against adversarial attacks

Important Caveats…

- (I) Strategic Behavior does not emerge from standard training
- (II) Naively trained models can be manipulated
- (III) Training without regard to the presence of other agents can lead to undesirable (e.g. collusive) consequences
- (IV) The optimization workhorse of Deep Learning (a.k.a. gradient descent) struggles in multi-agent settings
- (V) Finally Game Theory (namely the existence of Nash equilibrium and other types of equilibrium) breaks

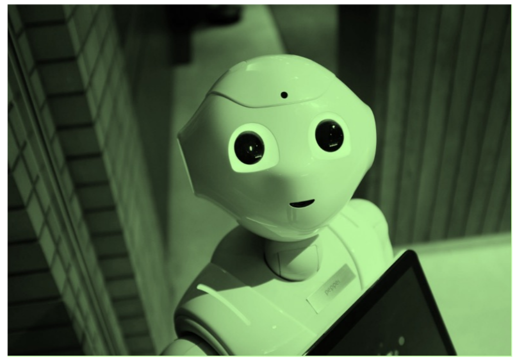
Important Caveats…

- (I) Strategic Behavior does not emerge from standard training
- (II) Naively trained models can be manipulated
- (III) Training without regard to the presence of other agents can lead to undesirable (e.g. collusive) consequences
- (IV) The optimization workhorse of Deep Learning (a.k.a. gradient descent) struggles in multi-agent settings
- (V) Finally Game Theory (namely the existence of Nash equilibrium and other types of equilibrium) breaks

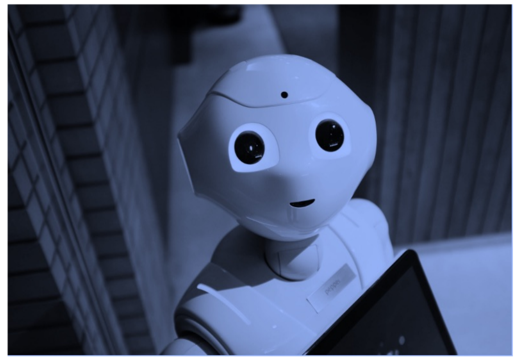
Today: Rather than imposing extra structure, or going after local equilibria, accept that strategy-sets might be infinite, e.g. represented by DNNs, or non-parametric & that utilities might be non-concave

- Go for full generality
- Characterize when eq existence/computation might be possible

Recall Setting: Infinite/Non-Parametric Games

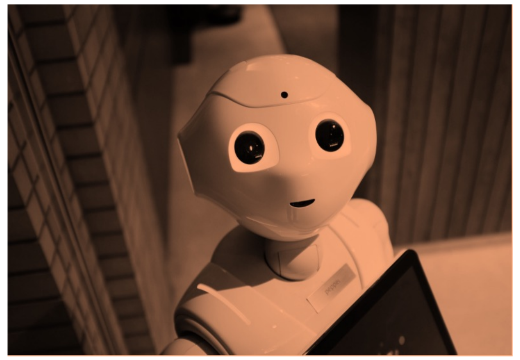


action: $x_1 \in \mathcal{X}_1$
goal: max $u_1(x_1, \dots, x_n)$



action: $x_2 \in \mathcal{X}_2$
goal: max $u_2(x_1, \dots, x_n)$

…



action: $x_n \in \mathcal{X}_n$
goal: max $u_n(x_1, \dots, x_n)$

- Action sets $\mathcal{X}_i$: high-dimensional or infinite-dimensional/non-parametric
- Utilities u_i : arbitrary functions $u_i : \times_i \mathcal{X}_i \to \mathbb{R}$
- Questions I want to ask:

Under what conditions do there exist **global** Nash/Correlated/Coarse Correlated Equilibria?

Are there simple methods converging to equilibria in a finite number of steps?

- For Q1: I hope that the answer depends on some complexity measure of the u_i's that I can identify
- For Q2: by "simple" I want that each step can be executed efficiently

Obstacle to Eq Existence: "Guess the larger number" Game

Player 2 (max player)

Player 1 (min player)

	1	**2**	**3**	**4**	**…**
1	1	1	1	1	…
2	-1	1	1	1	…
3	-1	-1	1	1	…
4	-1	-1	-1	1	…
…	…	…	…	…	…

A two-player zero-sum game where:

- $\mathcal{X}_1 = \mathcal{X}_2 = \mathbb{N}$
- $u_1(x_1, x_2) = -u_2(x_1, x_2) = 1_{x_1 \geq x_2} - 1_{x_1 < x_2}$
- (so table shows utility of Player 2)

Fact: "Guess the larger number" game has no Nash equilibrium (not even a very coarse approximate one).

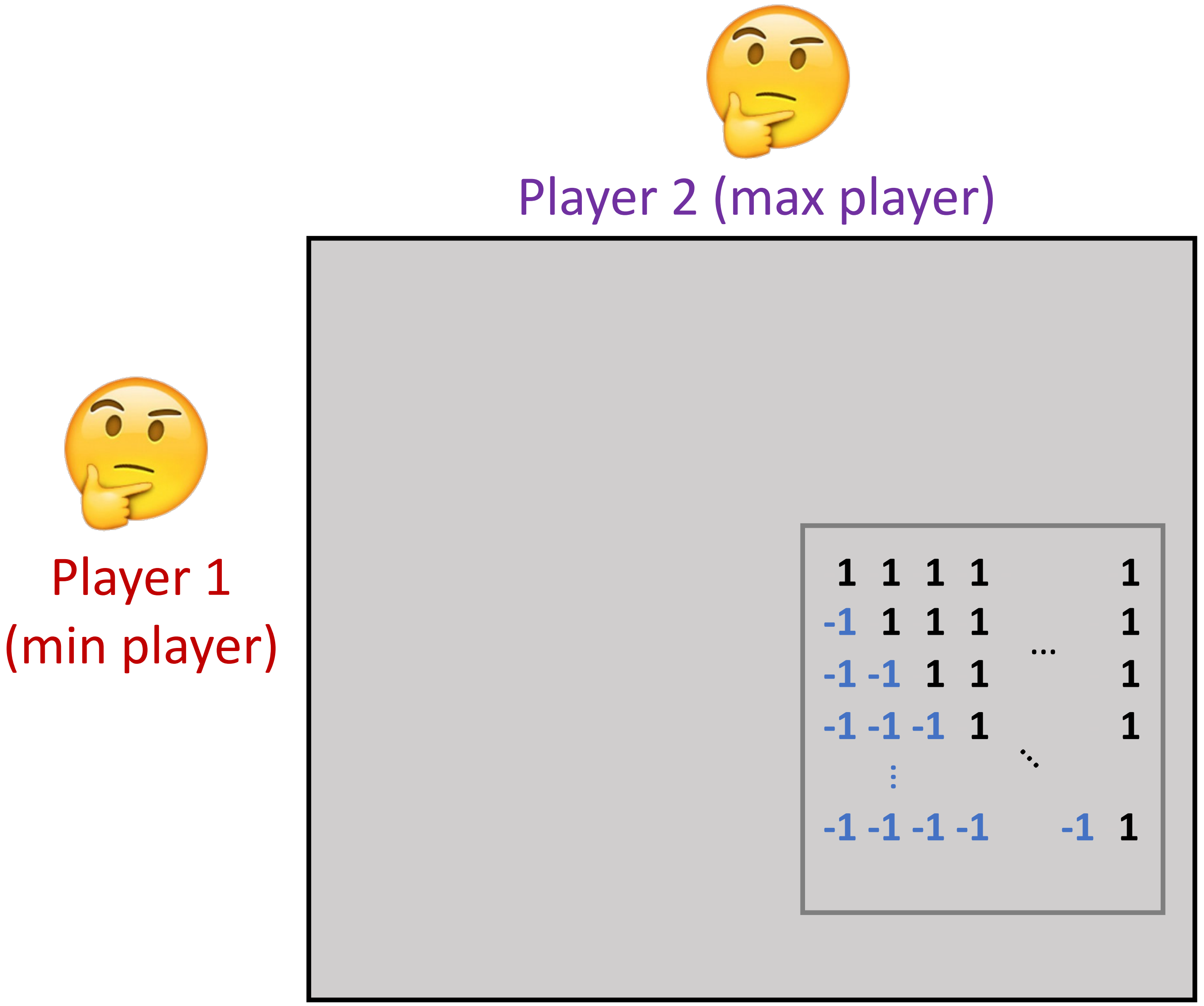
So "Guess the larger number game" is an obstacle to the existence of Nash equilibrium.

What if we exclude it?

What if we exclude “Guess the larger number”?

- Surprising fact: “Guess the larger number” game is the only obstacle to the existence of Nash equilibrium in $\{-1,1\}$-valued two-player zero-sum games!

Theorem [Hanneke-Livni-Moran’21]: If an (infinite) $\{-1,1\}$-valued two-player zero-sum game has no subgame which is “Guess the larger number,” then it has an ϵ-approximate Nash equilibrium for all $\epsilon > 0$.



G: $\{-1,1\}$-valued two-player zero-sum game

Threshold dimension of G: size of largest threshold sub-matrix

[Hanneke-Livni-Moran’21]: **Tr(G)** finite $\Rightarrow$ Minimax Eq exists

Claim: **Tr(G)** finite $\Leftrightarrow$ **Littlestone dimension** of G finite*

*: define Littlestone dimension of G in next slide

[Parenthesis: Littlestone dimension of a Game

Littlestone dimension of a Game

- G: a multiplayer $\{\pm 1\}$-valued game with utilities $u_i: \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \{\pm 1\}$
- For each player, consider the function class $H_i \coloneqq \{u_i(x_i, \cdot) \mid x_i \in \mathcal{X}_i\}$
  - H_i contains binary classifiers mapping each x_{-i} to ± 1
- **Littlestone dimension** of G is $\max_i \{\mathbf{Ldim}(H_i)\}$

Littlestone dimension of a Concept Class H of binary classifiers, mapping $\mathcal{X}$ to $\{\pm 1\}$

- TL;DR:
  - $\mathbf{Ldim}(H)$: characterizes whether and how well (in terms of regret) classifiers can be online learned from a sequence of adversarial data
- Specifically suppose that for $t = 1, \dots, T$:
  - learner chooses distribution p_t over $h_t \in H$
  - adversary chooses $(x_t, b_t) \in \mathcal{X} \times \{\pm 1\}$ (with knowledge of learner's distribution)
  - learner samples $h_t \sim p_t$ and experiences loss $\ell(h_t(x_t), b_t)) = \frac{1 - h_t(x_t) \cdot b_t}{2}$ (i.e. 1 if prediction is wrong ow 0)
- **[Rakhlin-Sridharan-Tewari'15, Hanneke-Livni-Moran'21]:** Can guarantee expected regret $\tilde{O}\left(\sqrt{T \cdot \mathbf{Ldim}(H)}\right)$ (which may be finite even when *H* is infinite!)

]

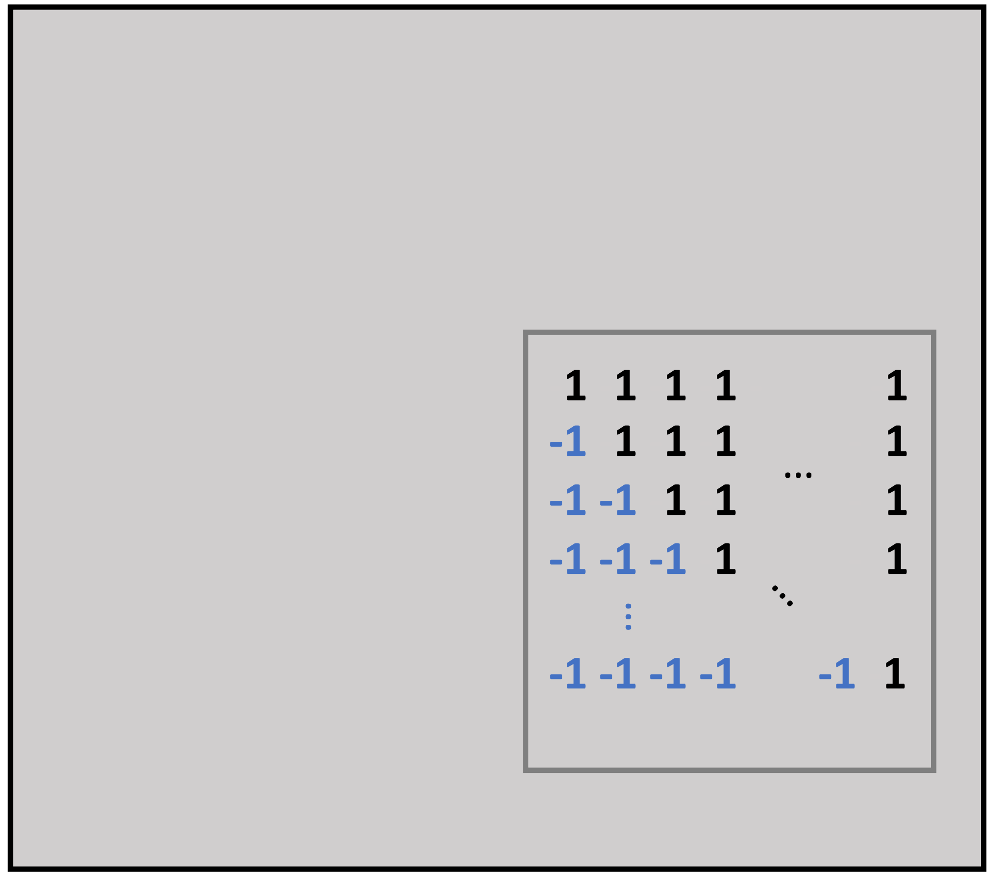
What if we exclude “Guess the larger number”?

- Surprising fact: “Guess the larger number” game is the only obstacle to the existence of Nash equilibrium in $\{-1,1\}$-valued two-player zero-sum games!

Theorem [Hanneke-Livni-Moran’21]: If an (infinite) $\{-1,1\}$-valued two-player zero-sum game has no subgame which is “Guess the larger number,” then it has an ϵ-approximate Nash equilibrium for all $\epsilon > 0$.

Player 2 (max player)

Player 1 (min player)



G: $\{-1,1\}$-valued two-player zero-sum game

Threshold dimension of G: size of largest threshold sub-matrix

[Hanneke-Livni-Moran’21]: Tr(G) finite $\Rightarrow$ Minimax Eq exists

Claim: Tr(G) finite $\Leftrightarrow$ **Littlestone dimension** of G finite

Littlestone dimension of G: $\max\{\mathbf{Ldim}(H_1), \mathbf{Ldim}(H_2)\}$

where $H_1 :=$ {rows of G viewed as binary classifiers over $\mathcal{X}_2$}
$H_2 :=$ {columns of G viewed as binary classifiers of $\mathcal{X}_1$}

$\mathbf{Ldim}(H)$: characterizes online learnability of H (from stream of examples)

Suggests: perhaps equilibria can be found through learning…

- hold that thought

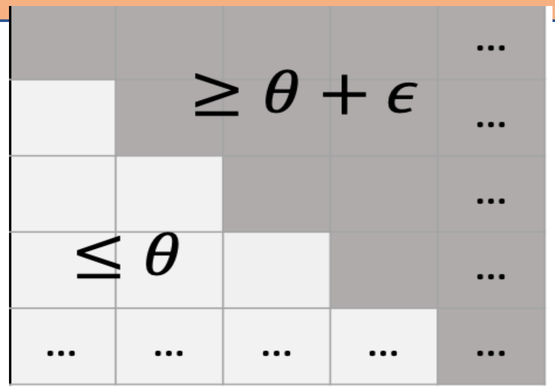
How about real-valued games?

- Surprising fact: "Guess the larger number" game is the only obstacle to the existence of Nash equilibrium in $\{-1,1\}$-valued two-player zero-sum games!

[Hanneke-Livni-Moran'21]: If an (infinite) $\{-1,1\}$-valued two-player zero-sum game has no subgame which is "Guess the larger number" (a.k.a. has finite Tr(G) $\Leftrightarrow$ finite Lit(G)) then it has an ϵ-approximate Nash eq for all $\epsilon > 0$.

[Daskalakis-Golowich'21] (Real-valued generalization of the above; informal):
If an (infinite) real-valued two-player zero-sum game has no subgame which is **ϵ-close to** some "scaling" of "Guess the larger number," then it has $O(\epsilon)$-approximate Nash equilibrium.

Formal result: requires finiteness of **ϵ-Fat Threshold** or **ϵ-sequential fat shattering dimension** (which are respectively generalizations of threshold dimension and Littlestone dimension to real-valued functions).

- **Def:** **ϵ-FatTr**(G) is the largest subgame satisfying  for some θ.

- **Def:** ϵ**-seqFat**$(G) = \max_i \epsilon$**–seqFat**(H_i) where $H_i := \{u_i(x_i, \cdot) \mid x_i \in \mathcal{X}_i\}$

[Rakhlin-Sridharan-Tewari'15

- TL;DR: ϵ–seqFat(H) characterizes online learnability of concept class H; achievable regret: $O(\epsilon \cdot T) + \tilde{O}\left(\sqrt{T \cdot \epsilon\text{–seqFat}(H)}\right)$

Next Question: Equilibrium Learning?

Question: Can we get equilibrium learning dynamics for *binary games* with finite Littlestone dimension?

- **challenge:** standard no-regret learning algorithms have cumulative T-round regret: $\sqrt{\log(\#\text{ actions})\,T}$

[Hanneke, Livni, Moran'21] There is a no-regret learning algorithm so that if each player uses it then their regret is $\tilde{O}(\mathrm{Ldim}^{1/2} \cdot T^{1/2})$; even in multi-player general-sum *binary* games.

- **remark:** no explicit dependence on # actions; note that $\mathrm{Ldim} \leq \log(\#\text{actions})$ always

[Daskalakis-Golowich, '21]: There is a no-regret learning algorithm so that if each player uses it then their regret is $\tilde{O}(\mathrm{Ldim}^{3/4} \cdot T^{1/4})$; even in multi-player general-sum *binary* games.

- **remark:** when $\#\text{actions}$ finite, rate dependence on T matches **[Syrgkanis et al'15]** obtained through optimistic methods (although not quite the near-optimal $\mathrm{poly}(\log T)$ rates of **[Daskalakis-Fishelson-Golowich'21, …]**)

Corollary: For the above algorithm, in the two-player zero-sum binary game setting, the empirical averages of each player's iterates are a $\tilde{O}(\mathrm{Ldim}^{3/4} \cdot T^{-3/4})$-approximate Nash equilibrium.
In the multi-player general-sum binary game setting, the empirical averages of the players' joint strategy profiles are an $\tilde{O}(\mathrm{Ldim}^{3/4} \cdot T^{-3/4})$-approximate Coarse Correlated Equilibrium.

Big Practical Issue: All known learning algorithms use the so-called "SOA oracle" which is very inefficient!

- also, missing online learning algorithms for CCE in multi-player *real-valued* games (exist non-constructive algos)
- also, no understanding of when CE exists (in binary or real-valued settings)

Meanwhile what do people do in practice?

Double Oracle Algorithm **[McMahan-Gordon-Blum’03]** (also used in PSRO)

<u>Double Oracle Algorithm</u>

Setting: two-player zero-sum game $G = (\mathcal{A}, \mathcal{B}, u)$, $\mathcal{A}$: minimizer’s strategies
Input: nonempty finite subsets $A_0 \subseteq \mathcal{A}$, $B_0 \subseteq \mathcal{B}$, and $\varepsilon \geq 0$

Let t: $= 0$
repeat
    Find a minimax equilibrium (p_t^*, q_t^*) of subgame (A_t, B_t, u)
    Find some $a_{t+1} \in \mathrm{BR}_{\mathcal{A}}(q_t^*)$ and $b_{t+1} \in \mathrm{BR}_{\mathcal{B}}(p_t^*)$
    Let $A_{t+1} := A_t \cup \{a_{t+1}\}$ and $B_{t+1} := B_i \cup \{b_{t+1}\}$
    $t \coloneqq t + 1$
end if $u(p_t^*, b_{t+1}) - u(a_{t+1}, q_t^*) \leq \varepsilon$

Output: ε-equilibrium (p_t^*, q_t^*) of game G

Question (also asked in **[Gemp et al.’22]**)**:** under what conditions does this end in finite time?
How about multi-player/general-sum generalizations of this algorithm?
[Assos-Atttias-Dagan-Daskalakis-Fishelson’23]: provide answers to both questions!

Computing equilibrium "practically"

- **Setting:** an infinite zero-sum game (+ extensions to general-sum in our paper)
- **Goal:** compute a minimax equilibrium using an easy-to-compute oracle
- We'll assume access to two oracles:
  - **Best-response (aka ERM) oracle:** given a finitely-supported mixed strategy of the opponent, returns a best response
  - **Value oracle:** given strategies for both players, output the utility

Theorem [Assos, Attias, Dagan, Daskalakis, Fishelson '23]: There is a (variation to the double-oracle) algorithm that computes an ϵ-minimax equilibrium using a best-response oracle for both players, in time $2^{O(\mathbf{Ldim}/\epsilon^2)}$ (if the game has binary values) and time $2^{O(\epsilon\text{-}\mathbf{seqFat}/\epsilon^2)}$ (if the games has general values)

Theorem [Hazan, Koren '16]: For any d there exists a two-player zero-sum binary game with $\mathrm{Ldim} = d$, such that any algorithm that accesses the game solely via best-response and value oracles, requires $2^{\mathrm{Ldim}/2}$ oracle calls to compute an $\epsilon = 1/4$ minimax equilibrium.

- **What's the point of our result?**

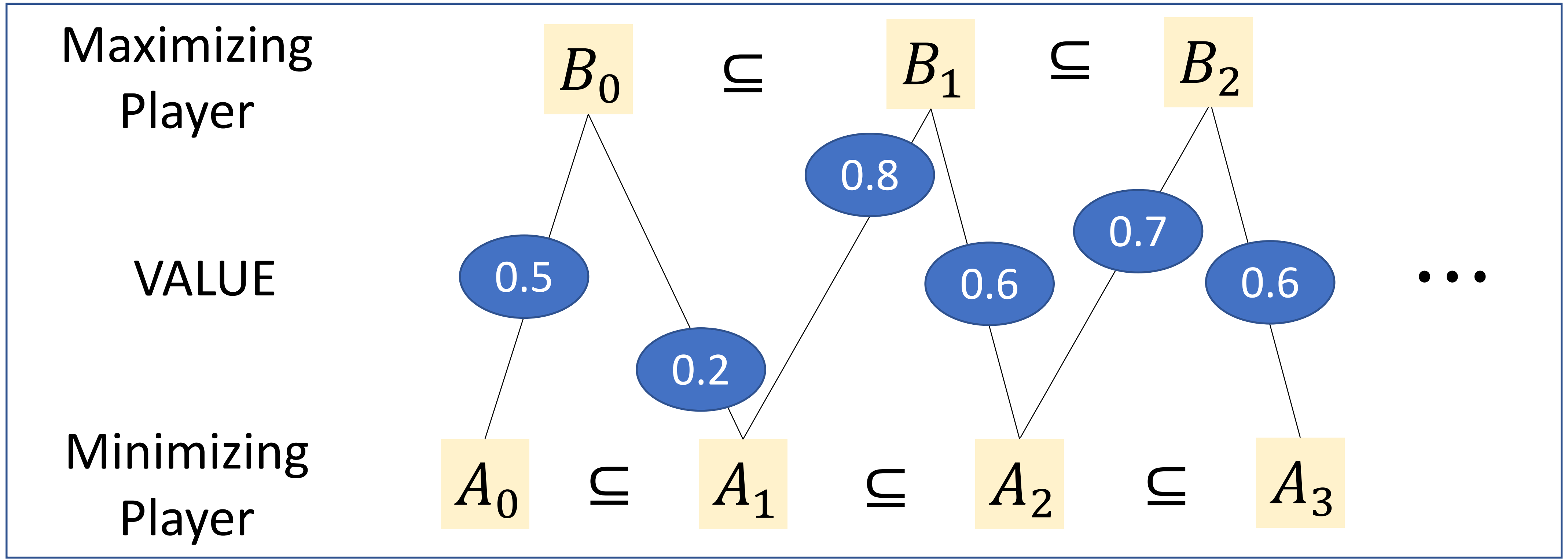
  good per iteration complexity (assuming ERM oracle)!

Algorithm: a variant of Double-Oracle

- A "Turn-based" Double Oracle algorithm
- The algorithm computes action sets $A_0 \subseteq A_1 \subseteq \cdots \subseteq \mathcal{A}$ for the minimizing player (player w/ action set $\mathcal{A}$) and $B_0 \subseteq B_1 \subseteq \cdots \subseteq \mathcal{B}$ for the maximizing player (player w/ action set $\mathcal{B}$) such that

$$\mathrm{Val}(A_{t+1}, B_t) = \mathrm{Val}(\mathcal{A}, B_t) \leq \mathrm{Val}(A_t, B_t) - \epsilon$$
$$\mathrm{Val}(A_{t+1}, B_{t+1}) = \mathrm{Val}(A_{t+1}, \mathcal{B}) \geq \mathrm{Val}(A_{t+1}, B_t) + \epsilon$$

- Each iteration is implemented using **Best-Response** and **Value** oracle calls
- **Central Claim:** The algorithm is guaranteed to terminate after $2^{O(\mathbf{Ldim}/\epsilon^2)}$ (binary-valued games) or $2^{O(\epsilon\text{-}\mathbf{seqFat}/\epsilon^2)}$ (real-valued games) iterations!
  - An ϵ-minimax equilibrium can be computed from there!



Computing B_{t+1} (similarly A_{t+1})

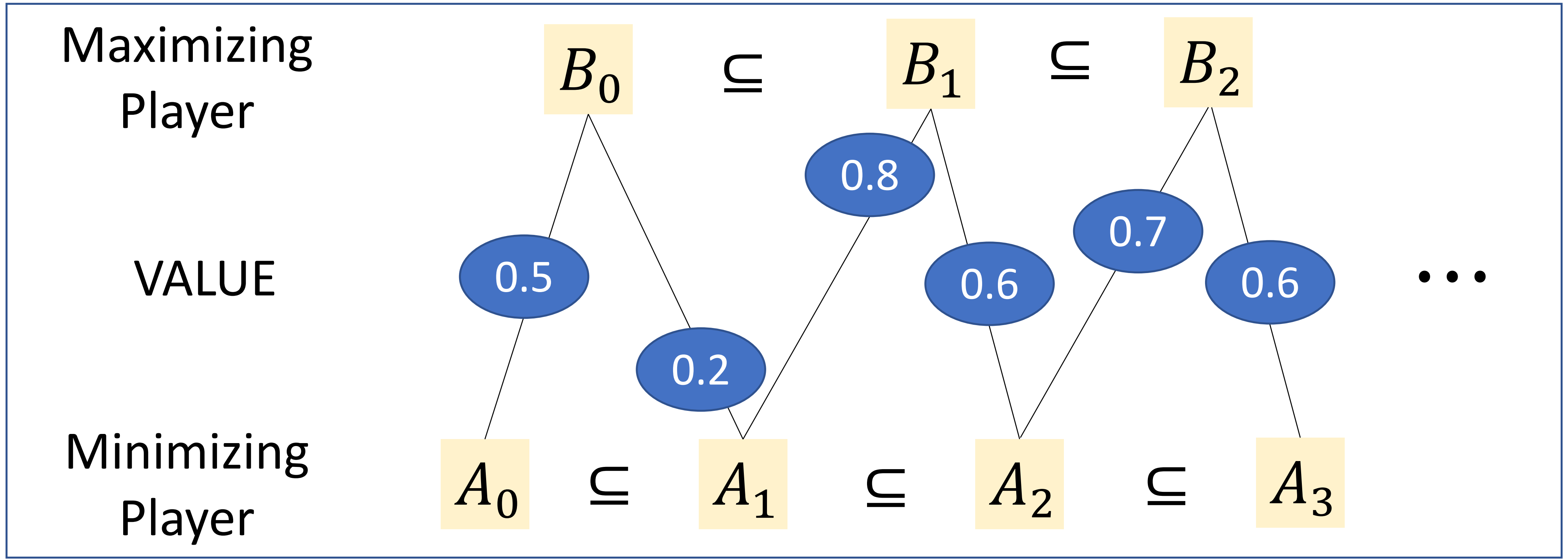
- Alternatingly, over multiple rounds, Player A updates her randomization over A_{t+1} (which is finite!) using a no-regret learning algorithm, and Player B plays her best-response over the full set $\mathcal{B}$ (using ERM oracle!) against A's average history so far (i.e. runs Be-The-Leader algorithm)
- $B_{t+1} \leftarrow B_t \cup \{\text{actions played by Player B}\}$

Algorithm: a variant of Double-Oracle

- A "Turn-based" Double Oracle algorithm
- The algorithm computes action sets $A_0 \subseteq A_1 \subseteq \cdots \subseteq \mathcal{A}$ for the minimizing player (player w/ action set $\mathcal{A}$) and $B_0 \subseteq B_1 \subseteq \cdots \subseteq \mathcal{B}$ for the maximizing player (player w/ action set $\mathcal{B}$) such that

$$\mathrm{Val}(A_{t+1}, B_t) \approx \mathrm{Val}(\mathcal{A}, B_t) \leq \mathrm{Val}(A_t, B_t) - \epsilon$$
$$\mathrm{Val}(A_{t+1}, B_{t+1}) \approx \mathrm{Val}(A_{t+1}, \mathcal{B}) \geq \mathrm{Val}(A_{t+1}, B_t) + \epsilon$$

- Each iteration is implemented using **Best-Response** and **Value** oracle calls
- **Central Claim:** The algorithm is guaranteed to terminate after $2^{O(\mathbf{Ldim}/\epsilon^2)}$ (binary-valued games) or $2^{O(\epsilon\text{-}\mathbf{seqFat}/\epsilon^2)}$ (real-valued games) Iiterations!
  - An ϵ-minimax equilibrium can be computed from there!



Computing B_{t+1} (similarly A_{t+1})

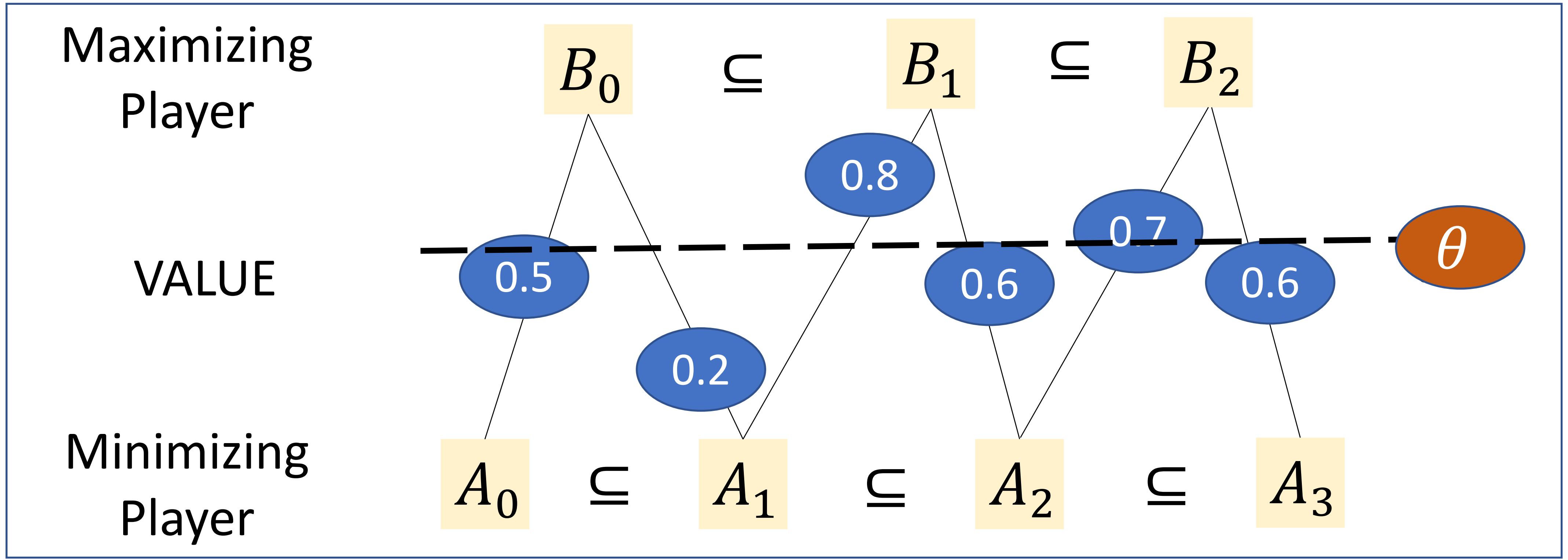
- Alternatingly, over multiple rounds, Player A updates her randomization over A_{t+1} (which is finite!) using a no-regret learning algorithm, and Player B plays her best-response over the full set $\mathcal{B}$ (using ERM oracle!) against A's average history so far (i.e. runs Be-The-Leader algorithm)
- $B_{t+1} \leftarrow B_t \cup \{\text{actions played by Player B}\}$

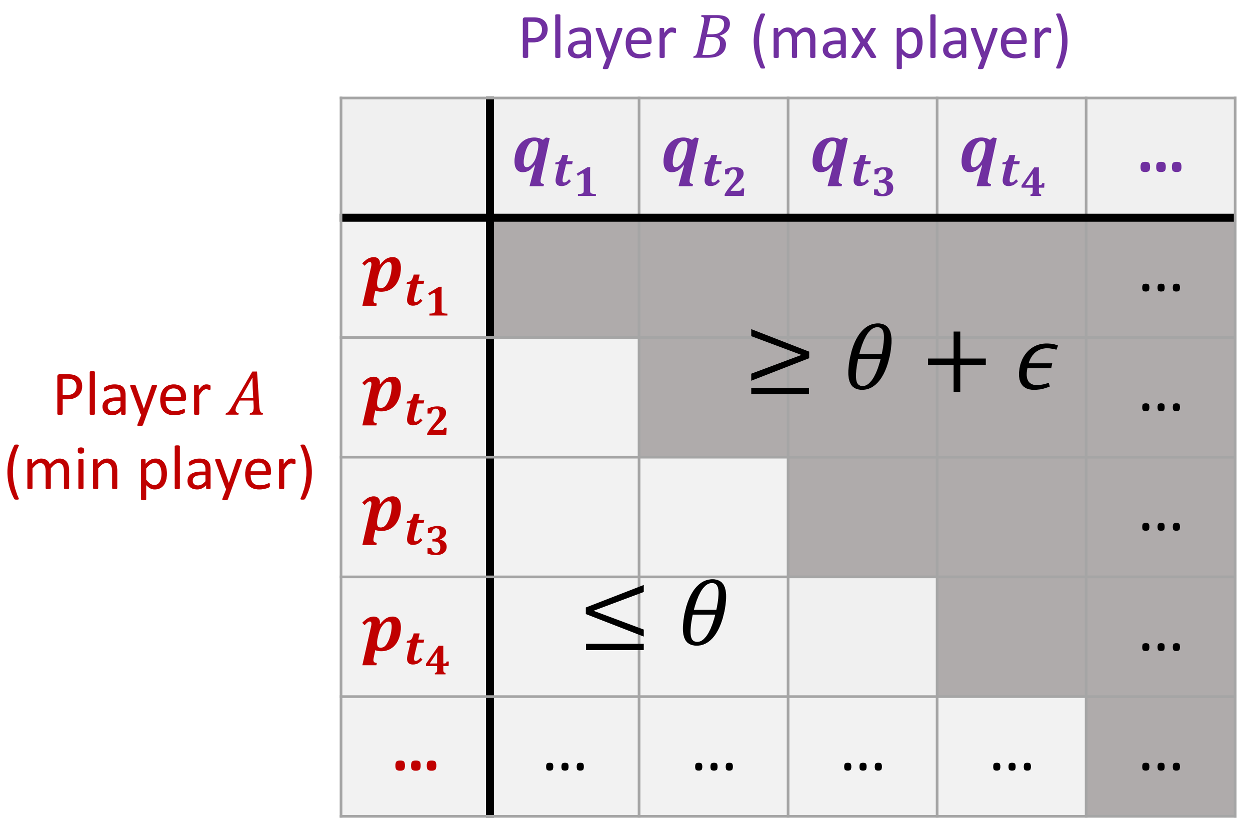
Analyzing the game: binary case

- Assume that the algorithm proceeds for T iterations (want to show must be finite)
- **Claim:** can find $t_1, t_2, \ldots, t_{k=\Theta(T\epsilon)}$ and a threshold θ such that

$$\mathrm{Val}\left(A_{t_i}, B_{t_j}\right) \leq \theta \ \text{ if } i > j\,; \mathrm{Val}\left(A_{t_i}, B_{t_j}\right) \geq \theta + \epsilon \ \text{ if } i \leq j$$

- Hence, there exists an ϵ-separated "guess-the-larger-number" subgame of **mixed** strategies $p_{t_1}, \ldots, p_{t_k}$, $q_{t_1}, \ldots, q_{t_k}$, where (p_{t_i}, q_{t_i}) is minmax strategy of the finite subgame
  - By **[Hanneke-Livni-Moran '21]**, **[Assos, Attias, Dagan, Daskalakis, Fishelson '23]**, there exists a guess-the-larger-number subgame of **pure** strategies of size about $\log k$.* (*if time permits)
  - Since threshold dimension is bounded (a.k.a. Littlestone is bounded), the size of this subgame is bounded.
  - This yields a bound on the number of iterations





Analyzing the game: binary case

Large

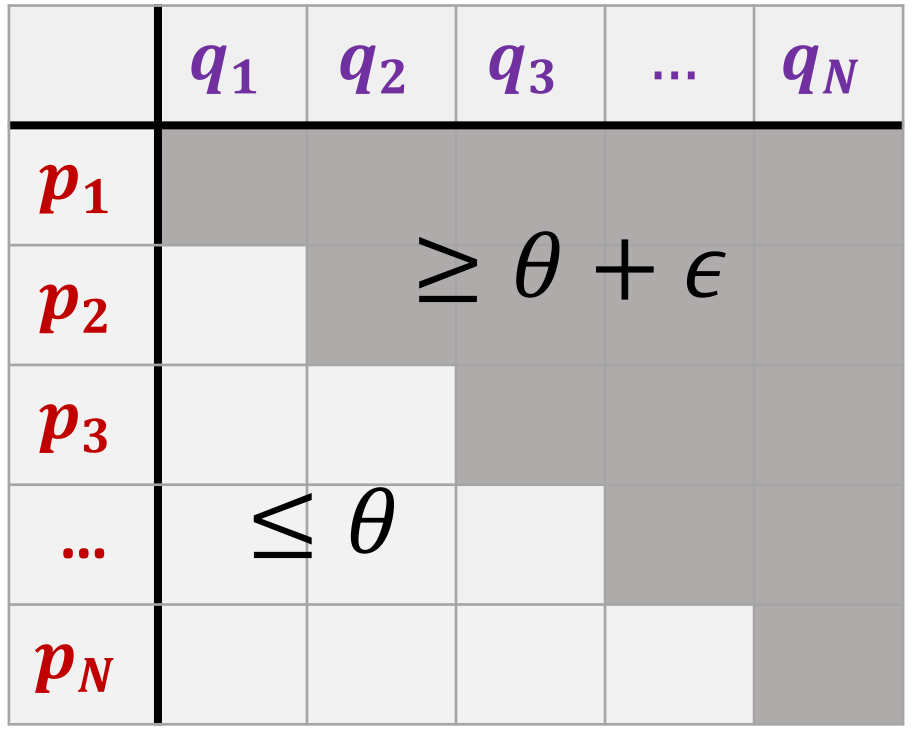
	q_1	q_2	q_3	…	q_N
p_1					
p_2			$\geq \theta + \epsilon$		
p_3					
…		$\leq \theta$			
p_N					

$\Longrightarrow$

Large-ish

	b_1	b_2	b_3	…	b_n
a_1					…
a_2			1		…
a_3					…
…		-1			…
a_m	…	…	…	…	…

Analyzing the game: binary case



$\Downarrow$

for all i: $a_i^1, \ldots, a_i^Q \sim_{iid} p_i$

for all j: $b_j^1, \ldots, b_j^Q \sim_{iid} q_j$

where $Q = \frac{VC(G)}{\epsilon^2}$

then $u(\bar{a}_i, \bar{b}_j) \geq \theta + \epsilon/2, \forall i \leq j$

$u(\bar{a}_i, \bar{b}_j) \leq \theta + \epsilon/4, \forall i > j$

$\Rightarrow$

$i \leq j$:

$$u(\bar{a}_i, \bar{b}_j) = \frac{1}{Q^2} \cdot \sum_{k=1}^{Q} \sum_{\ell=1}^{Q}$$

	b_j^1	b_j^2	b_j^3 …	b_j^Q
a_i^1	-1	1	1	-1
a_i^2	1	-1	1	1
a_i^3	1	1	-1	-1
⋮				
a_i^Q	-1	1	1	1

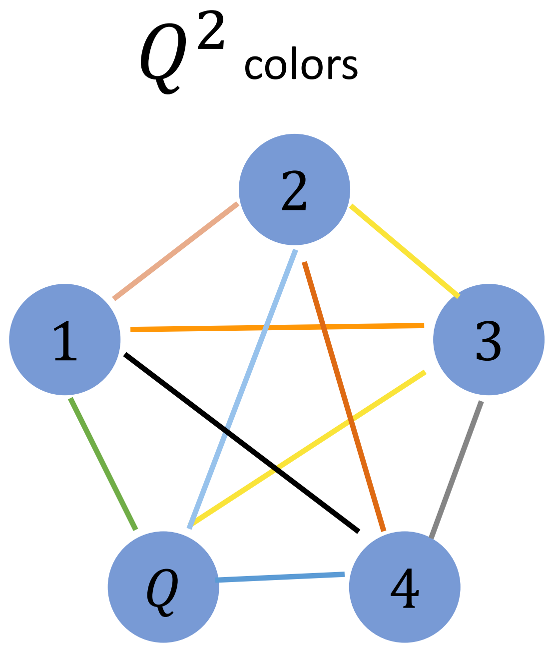
$i > j$:

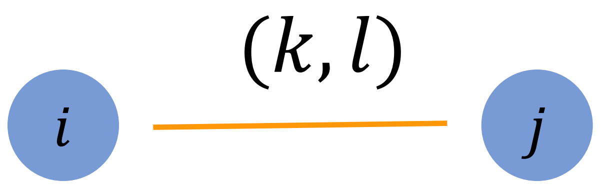
$$u(\bar{a}_i, \bar{b}_j) = \frac{1}{Q^2} \cdot \sum_{k=1}^{Q} \sum_{\ell=1}^{Q}$$

	b_j^1	b_j^2	b_j^3 …	b_j^Q
a_i^1	1	1	-1	-1
a_i^2	1	-1	-1	1
a_i^3	1	1	-1	-1
⋮				
a_i^Q	-1	1	-1	1

Analyzing the game: binary case

For all $i > j$, there exists $k, \ell \in [Q]$ such that $u\left(a_i^k, b_j^\ell\right) = -1$ and $u\left(a_j^k, b_i^\ell\right) = 1$

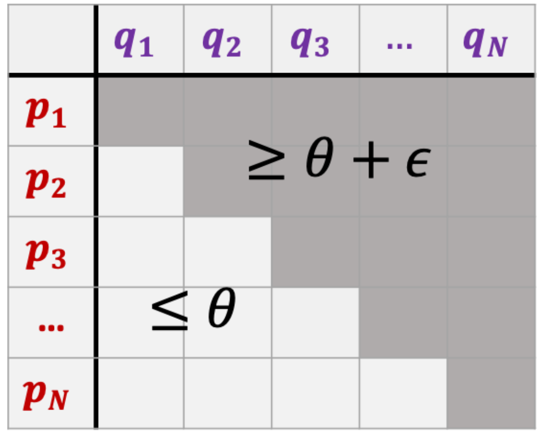




Ramsey: Original graph size $N \Longrightarrow$ monochromatic $n \approx \frac{\log N}{Q^2}$ -size clique must exist!

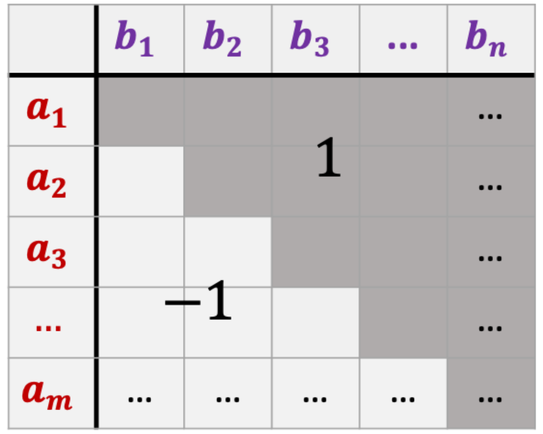
There exist $t_1, t_2, \cdots, t_n$ such that: $u\left(a_{t_i}^k, b_{t_j}^\ell\right) = -1$ for all $i > j$ and $u\left(a_{t_i}^k, b_{t_j}^\ell\right) = 1$ for all $i \leq j$

Large



$\Longrightarrow$

Large-ish



Assume bounded $\mathbf{Tr}(G)$, $\mathbf{VC}(G)$: number of algorithm iterations $\approx \epsilon^{O(TDim \cdot Q^2)} = \epsilon^{O(TDim \cdot VCDim^2/\epsilon^4)}$

Can also get bound in terms of $\mathbf{Ldim}(G)$: $e^{O(LDim/\epsilon^2)}$

General Results

Setting	Time per iter.	BR calls/it.	#iterations
Minmax, 0-1 valued	t/ϵ^4	$\log t/\epsilon^2$	$C^{\mathrm{Lit}(G)/\epsilon^2} \wedge \epsilon^{-C\,\mathrm{VC}(G)^2\,\mathrm{tr}(G)/\epsilon^4}$
Minmax, real valued	t/ϵ^4	$\log t/\epsilon^2$	$C^{\mathrm{sfat}(G,\epsilon)/\epsilon^2} \wedge \epsilon^{-CI(G)^2\,\mathrm{fattr}(G,\epsilon)/\epsilon^5}$
CCE, 0-1 valued	kt/ϵ^2	$k\log t/\epsilon^2$	$C^{(k/\epsilon^3)\,\mathrm{Lit}(G)} \wedge \epsilon^{-C(k^3/\epsilon^6)\,\mathrm{VC}(G)^2\,\mathrm{tr}(G)}$
CCE, real valued	kt/ϵ^2	$k\log t/\epsilon^2$	$C^{(k/\epsilon^3)\,\mathrm{sfat}(G,\epsilon)} \wedge \epsilon^{-C(k^3/\epsilon^6)I(G)^2\,\mathrm{fattr}(G,\epsilon)}$

Table 2: The table describes the time per iteration, the number of best-response calls per iteration and the number of iterations of our algorithms, up to polylogarithmic factors for finding an $O(\epsilon)$-approximate Nash in a zero-sum two player game (minmax equilibrium) and Coarse Correlated Equilibrium (CCE) in general games G. Here, $C > 0$ is a universal constant, and $\mathrm{Lit}, \mathrm{VC}, \mathrm{tr}, \mathrm{sfat}, \mathrm{fat}, \mathrm{fattr}$ denote Littlestone, VC, threshold, sequential fat, fat and fat-threshold dimensions of G, $I(G) = \int_0^1 \left(\sqrt{\mathrm{fat}(G,\delta)d\delta}\right)^2$ and $\wedge$ denotes a minimum of two terms.

Conclusions

- ML developments motivate deeper study of high-dimensional/non-parametric/non-concave games
- In these games, pure Nash equilibria may fail to exist, while mixed Nash equilibria, correlated equilibria and other game-theoretic solution concepts may fail to exist or, if they do exist, they can be infinitely supported
- This motivates studying:
  - **local** notions of stability, e.g. *local pure Nash equilibria [c.f. lecture 18]*
  - games w/ **special structure**, e.g. *stochastic games, extensive-form games [c.f. lectures 9-17]*
  - or **arbitrary** games *[lectures 19 + 20 (today!)]*
    - characterize existence of finitely supported equilibria
    - develop algorithms for learning equilibria
    - in particular, we showed characterization results for the existence of finitely supported Nash and Coarse Correlated Equilibria, and identified algorithms whose iterations can be executed efficiently and are guaranteed to converge to equilibrium.
    - correlated?
      - **[Dagan-Daskalakis-Golowich-Fishelson'23]**: no-regret learning possible $\Rightarrow$ correlated equilibria exist!
- Broad topic that is widely unexplored!
- **Let us call it a class!**