Learning against No-Regret Learners

Long Tran-Thanh

University of Warwick, UK

long.tran-thanh@warwick.ac.uk

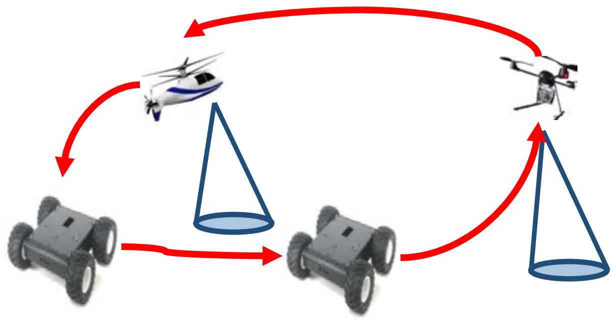
Joint work with **Le Cong Dinh**, Alain Zemkoho, Tri-Dung Nguyen (Southampton)

Shivakumar Mahesh (Warwick), Nick Bishop (Oxford)

Workshop on Learning in Games, Toulouse, France

1-3 July 2024

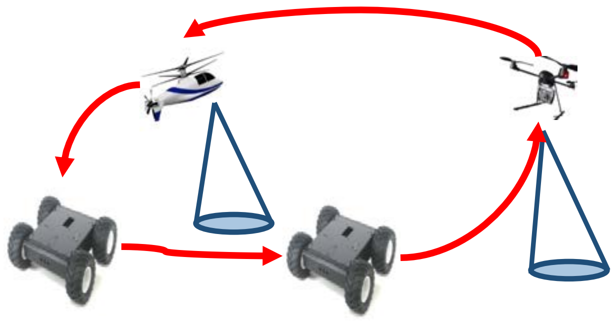
Multi-Agent Systems



System of **no-regret learners**:

- Selfish behaviour + learning ability
- Different agents may follow different learning algorithms

Multi-Agent Systems

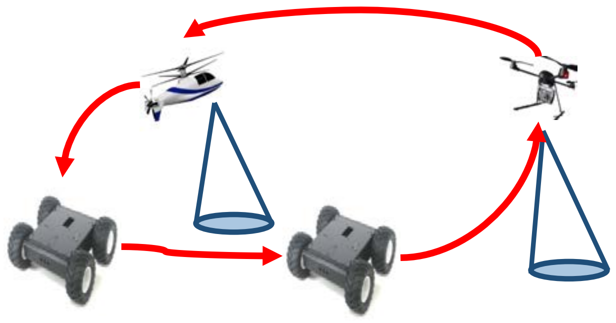


System of **no-regret learners**:

- Selfish behaviour + learning ability
- Different agents may follow different learning algorithms

Question: Can we gain any benefits from playing against these no-regret learners?

Multi-Agent Systems



System of **no-regret learners**:

- Selfish behaviour + learning ability

- Different agents may follow different learning algorithms

Question: Can we gain any benefits from playing against these no-regret learners?

Part 1:
- Topic 1: **Better regret bounds** against no-regret learners
- Topic 2: Last-iterate convergence with **generic no-regret learners**

Part 2:
- Topic 3: Exploiting no-regret learners with (minimal) **payoff manipulation** (setting: **coopetitive games**)

1. Better regret bounds in repeated games

Problem setting

Repeated 2-player zero-sum game: agent, adversary

- At each t = {1,..T}: agent chooses strategy (action) $f_t \in \mathcal{F} \subseteq [0,1]^n$
- Adversary simultaneously chooses strategy $x_t \in \mathcal{X} \subseteq [0,1]^n$
- Agent observes loss $\langle f_t, x_t \rangle$ and x_t (full information feedback)
- Adversary is a no-(external)-regret learner:

$$\frac{1}{T} \max_{x \in \mathcal{X}} \sum_{t=1}^{T} (\langle f_t, x \rangle - \langle f_t, x_t \rangle) \to 0, \quad T \to \infty$$

Regret minimization notions

The agent's objective: $\min_{f_1, f_2, \ldots, f_T} \sum_{t=1}^{T} (\langle f_t, x_t \rangle)$

Regret minimization notions

The agent's objective: $\min_{f_1, f_2, \ldots, f_T} \sum_{t=1}^{T} (\langle f_t, x_t \rangle)$

Dynamic regret: $\mathrm{DR}_T = \sum_{t=1}^{T} (\langle f_t, x_t \rangle - \min_{g_t \in \mathcal{F}} \langle g_t, x_t \rangle)$

(sub-linear dynamic regret: only if $\{x_t\}_{t=1}^{T}$ can be estimated efficiently)

Regret minimization notions

The agent's objective: $\min_{f_1, f_2, \ldots, f_T} \sum_{t=1}^{T} (\langle f_t, x_t \rangle)$

Dynamic regret: $\mathrm{DR}_T = \sum_{t=1}^{T} (\langle f_t, x_t \rangle - \min_{g_t \in \mathcal{F}} \langle g_t, x_t \rangle)$

(sub-linear dynamic regret: only if $\{x_t\}_{t=1}^{T}$ can be estimated efficiently)

External regret: $\mathrm{R}_T = \min_{f \in \mathcal{F}} \sum_{t=1}^{T} (\langle f_t, x_t \rangle - \langle f, x_t \rangle)$

Regret minimization notions

The agent's objective: $\min_{f_1,f_2,\ldots,f_T} \sum_{t=1}^{T}(\langle f_t, x_t\rangle)$

Dynamic regret: $\mathrm{DR}_T = \sum_{t=1}^{T}(\langle f_t, x_t\rangle - \min_{g_t\in\mathcal{F}}\langle g_t, x_t\rangle)$

(sub-linear dynamic regret: only if $\{x_t\}_{t=1}^{T}$ can be estimated efficiently)

External regret: $\mathrm{R}_T = \min_{f\in\mathcal{F}} \sum_{t=1}^{T}(\langle f_t, x_t\rangle - \langle f, x_t\rangle)$

Forward regret (Saha *et al.*, 2012):

$$FR_T := \sum_{t=1}^{T} \left(\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle\right), \text{where } \boldsymbol{g}_{t+1} = \arg\min_{\boldsymbol{g}\in\mathcal{F}} G_{t+1}(\boldsymbol{g}) = \langle \boldsymbol{g}, \sum_{s=1}^{t} \boldsymbol{x}_s + \boldsymbol{x}_{t+1}\rangle + \frac{R(\boldsymbol{g})}{\eta}$$

Regret minimization notions

The agent's objective: $\min_{f_1, f_2, \ldots, f_T} \sum_{t=1}^{T} (\langle f_t, x_t \rangle)$

Dynamic regret: $\mathrm{DR}_T = \sum_{t=1}^{T} (\langle f_t, x_t \rangle - \min_{g_t \in \mathcal{F}} \langle g_t, x_t \rangle)$

(sub-linear dynamic regret: only if $\{x_t\}_{t=1}^{T}$ can be estimated efficiently)

External regret: $\mathrm{R}_T = \min_{f \in \mathcal{F}} \sum_{t=1}^{T} (\langle f_t, x_t \rangle - \langle f, x_t \rangle)$

Forward regret (Saha *et al.*, 2012):

$$FR_T := \sum_{t=1}^{T} \left(\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right), \text{where } \boldsymbol{g}_{t+1} = \arg\min_{\boldsymbol{g} \in \mathcal{F}} G_{t+1}(\boldsymbol{g}) = \langle \boldsymbol{g}, \sum_{s=1}^{t} \boldsymbol{x}_s + \boldsymbol{x}_{t+1} \rangle + \frac{R(\boldsymbol{g})}{\eta}$$

Claim: $\mathrm{R}_T \leq \mathrm{FR}_T$

Achieving sub-linear forward regret

Algorithm 1: Accurate Follow the Regularized Leader (AFTRL)

Input: learning rate $\eta > 0$, exploiting rate $\alpha \geq 1$,

$f_1 = \arg\min_{f \in \mathcal{F}} R(f)$.

Output: next strategy update

$$f_{t+1} = \arg\min_{f \in \mathcal{F}} F_{t+1}(f) = \langle f, \sum_{s=1}^{t} \boldsymbol{x}_s + \alpha x_t \rangle + \frac{R(f)}{\eta}.$$

Achieving sub-linear forward regret

Algorithm 1: Accurate Follow the Regularized Leader (AFTRL)

Input: learning rate $\eta > 0$, exploiting rate $\alpha \geq 1$,
$f_1 = \arg\min_{f \in \mathcal{F}} R(f)$.
Output: next strategy update

$$f_{t+1} = \arg\min_{f \in \mathcal{F}} F_{t+1}(f) = \langle f, \sum_{s=1}^{t} x_s + \alpha x_t \rangle + \frac{R(f)}{\eta}.$$

Theorem: If $\mathcal{F}, \mathcal{X}$ are compact convex sets, $p, q > 0 : \frac{1}{p} + \frac{1}{q} = 1$, R is strongly convex in p-norm $\min_{f \in \mathcal{F}} R(f) = 0$ and adversary is a no-(external)-regret learner:

$$\mathrm{RT}_T(\mathrm{AFTRL}) \in O(1)$$

Achieving sub-linear forward regret

Algorithm 1: Accurate Follow the Regularized Leader (AFTRL)

Input: learning rate $\eta > 0$, exploiting rate $\alpha \geq 1$,
$f_1 = \arg\min_{f \in \mathcal{F}} R(f)$.
Output: next strategy update

$$f_{t+1} = \arg\min_{f \in \mathcal{F}} F_{t+1}(f) = \langle f, \sum_{s=1}^{t} x_s + \alpha x_t \rangle + \frac{R(f)}{\eta}.$$

Theorem: If $\mathcal{F}, \mathcal{X}$ are compact convex sets, $p, q > 0 : \frac{1}{p} + \frac{1}{q} = 1,$ R is strongly convex in p-norm $\min_{f \in \mathcal{F}} R(f) = 0$ and adversary is a no-(external)-regret learner:

$$\mathrm{RT}_T(\mathrm{AFTRL}) \in O(1)$$

Key steps: We show that $||x_{t+1} - x_t||_q \in O\left(\frac{1}{\sqrt{T}}\right)$ -> use x_t to predict x_{t+1}

Achieving sub-linear forward regret

Algorithm 1: Accurate Follow the Regularized Leader (AFTRL)

Input: learning rate $\eta > 0$, exploiting rate $\alpha \geq 1$,
$f_1 = \arg\min_{f \in \mathcal{F}} R(f)$.
Output: next strategy update

$$f_{t+1} = \arg\min_{f \in \mathcal{F}} F_{t+1}(f) = \langle f, \sum_{s=1}^{t} x_s + \alpha x_t \rangle + \frac{R(f)}{\eta}.$$

Theorem: If $\mathcal{F}, \mathcal{X}$ are compact convex sets, $p, q > 0 : \frac{1}{p} + \frac{1}{q} = 1$, R is strongly convex in p-norm $\min_{f \in \mathcal{F}} R(f) = 0$ and adversary is a no-(external)-regret learner:

$$\mathrm{RT}_T(\mathrm{AFTRL}) \in O(1)$$

Key steps: We show that $||x_{t+1} - x_t||_q \in O\left(\frac{1}{\sqrt{T}}\right)$ -> use x_t to predict x_{t+1}

Next: extend results for predictable sequences, eg., Rakhlin & Shridharan (2013)

Achieving sub-linear dynamic regret

Algorithm 2: Prod-Best Response algorithm (Prod-BR) – based on (A,B)-Prod (Sani *et al.*, 2014)

Input: learning rate $\eta > 0$, $\eta_1 \in (0, 1]$, initial weight $w_{1,R}, w_{1,BR}$, regularizer function $R(.)$.

$f_{t+1} = \arg\min_{f \in \mathcal{F}} F_{t+1}(f) = \langle f, \sum_{s=1}^{t} \boldsymbol{x}_s \rangle + \frac{R(f)}{\eta}$; $BR_{t+1} = \arg\min_{f \in \mathcal{F}} \langle f, \boldsymbol{x}_t \rangle$

Output: next strategy update $\boldsymbol{g}_{t+1}$ and next weight $w_{t+1,R}$:

$$\boldsymbol{g}_{t+1} = \frac{w_{t,R}}{w_{t,R} + w_{1,BR}} f_{t+1} + \frac{w_{1,BR}}{w_{t,FTRL} + w_{1,BR}} BR_{t+1}; \quad w_{t+1,R} = w_{t,R}(1 + \eta_1 \langle BR_{t+1} - f_{t+1}, \boldsymbol{x}_{t+1} \rangle).$$

Achieving sub-linear dynamic regret

Algorithm 2: Prod-Best Response algorithm (Prod-BR) – based on (A,B)-Prod (Sani *et al.*, 2014)

Input: learning rate $\eta > 0$, $\eta_1 \in (0, 1]$, initial weight $w_{1,R}, w_{1,BR}$, regularizer function $R(.)$.
$f_{t+1} = \arg\min_{f \in \mathcal{F}} F_{t+1}(f) = \langle f, \sum_{s=1}^{t} x_s \rangle + \frac{R(f)}{\eta}$; $BR_{t+1} = \arg\min_{f \in \mathcal{F}} \langle f, x_t \rangle$
Output: next strategy update g_{t+1} and next weight $w_{t+1,R}$:

$$g_{t+1} = \frac{w_{t,R}}{w_{t,R} + w_{1,BR}} f_{t+1} + \frac{w_{1,BR}}{w_{t,FTRL} + w_{1,BR}} BR_{t+1}; \quad w_{t+1,R} = w_{t,R}(1 + \eta_1 \langle BR_{t+1} - f_{t+1}, x_{t+1} \rangle).$$

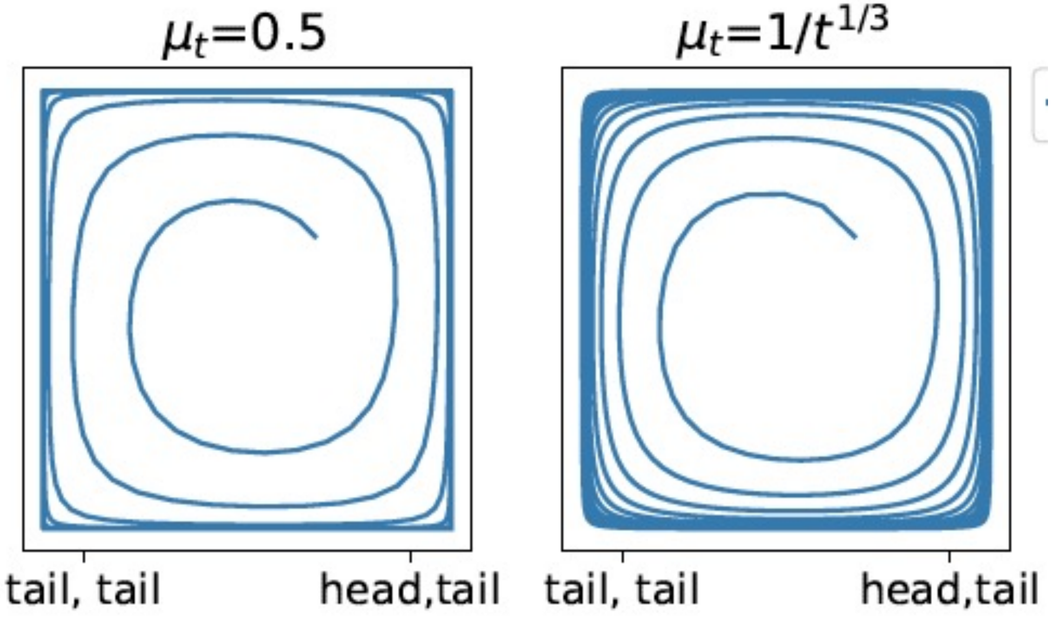
THEOREM 5.2. *Let the agent follows Prod-BR Algorithm 2 with* $\eta = n/\sqrt{2T}$, $\eta_1 = 1/2.\sqrt{\log(T)/T}$ *and* $w_{1,BR} = 1 - w_{1,R} = 1 - \eta_1$. *Then it achieves* $O(\sqrt{T \log(T)})$ *external regret against general adversary while maintaining* $O(\sqrt{T})$ *dynamic regret against no-external regret adversary.*

2. Last-iterate convergence in repeated games

Current state of the art

Repeated Matching Pennies after 2500 iterations:

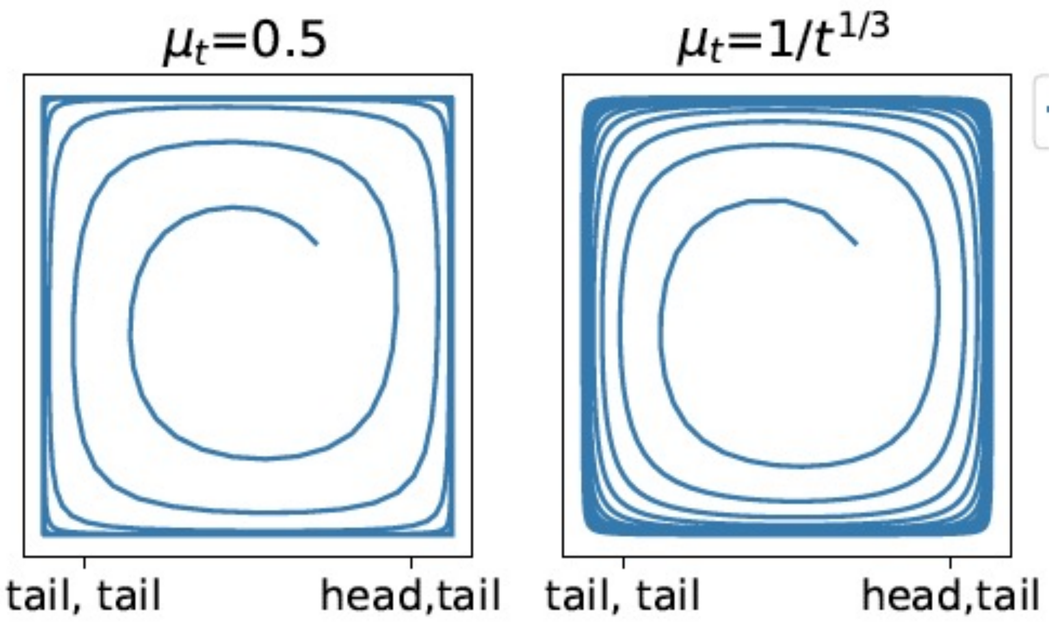
- No-regret learning alg: Multiplicative Weight Update
- Blue line: MWU vs MWU
- System dynamics: outward spiral -> no convergence



Current state of the art

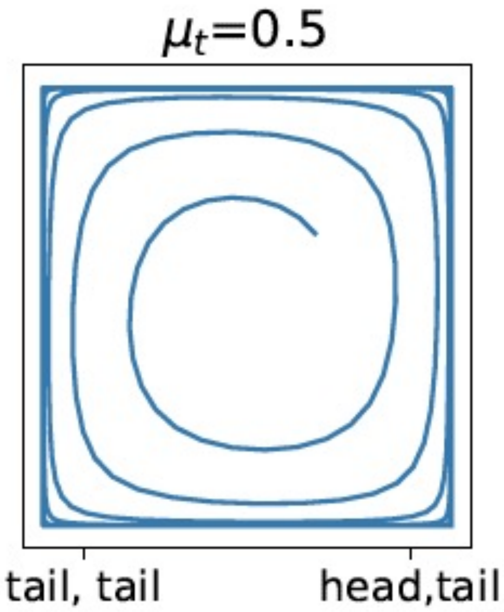
Repeated Matching Pennies after 2500 iterations:

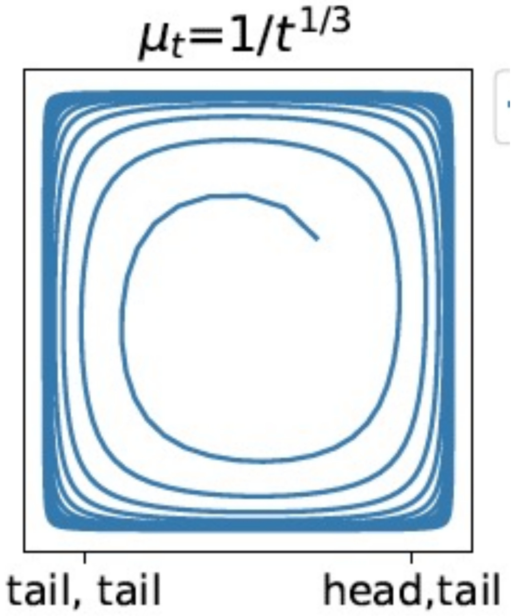
- No-regret learning alg: Multiplicative Weight Update
- Blue line: MWU vs MWU
- System dynamics: outward spiral -> no convergence



Known since Mertikopoulos, Papadimitriou & Piliouras (2018): **no last-iterate convergence** in general case. Other notable work: Bailey and Piliouras (2018), Cheung and Piliouras (2019)

Current state of the art





Repeated Matching Pennies after 2500 iterations:

- No-regret learning alg: Multiplicative Weight Update
- Blue line: MWU vs MWU
- System dynamics: outward spiral -> no convergence

Known since Mertikopoulos, Papadimitriou & Piliouras (2018): **no last-iterate convergence** in general case. Other notable work: Bailey and Piliouras (2018), Cheung and Piliouras (2019)

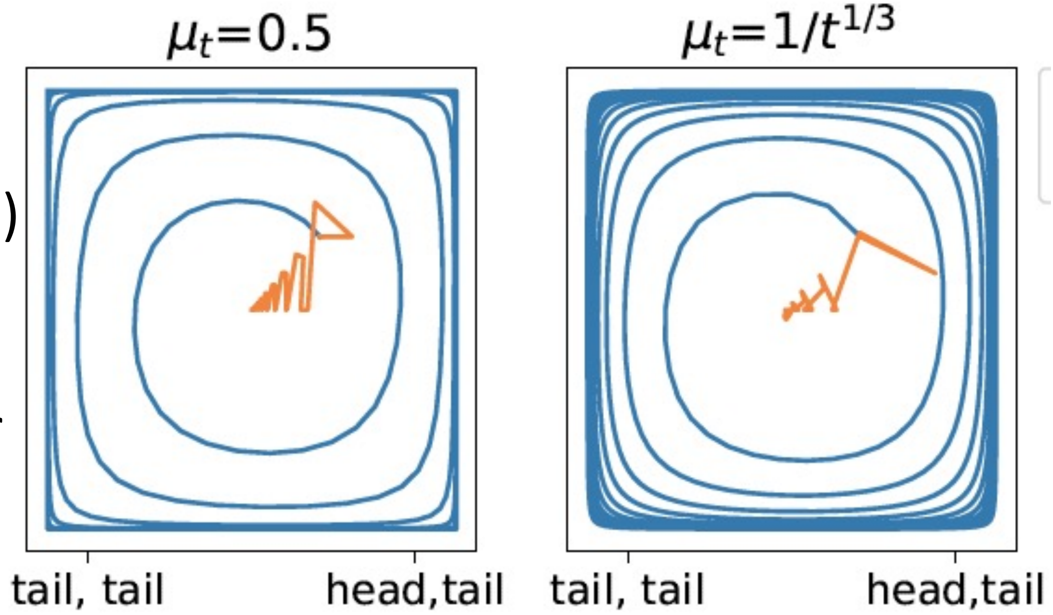
Existence of last-iterate convergence – some special cases:

- Daskalakis and Panageas (2018): Optimistic MWU + unique minimax equilibrium
- Bu, Ratliff & Mesbahi (2019): Differential games (linear-quadratic) + gradient ascent/descent
- Goktas & Greenwald (2022): Exploitability-minimising strategy profiles

Last-iterate convergence with asymmetric knowledge

2-player zero-sum + asymmetric information:

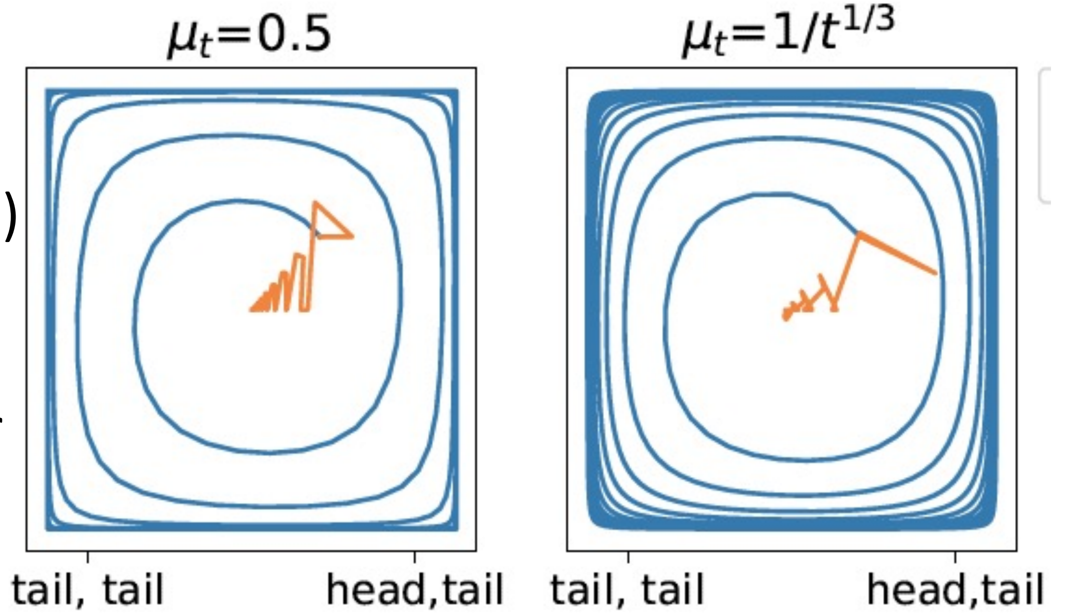
- Column player (agent) can estimate her (approximate) minimax strategy
- Row player (adversary) is a no-external-regret learner



Last-iterate convergence with asymmetric knowledge

2-player zero-sum + asymmetric information:

- Column player (agent) can estimate her (approximate) minimax strategy
- Row player (adversary) is a no-external-regret learner



At each time step t:

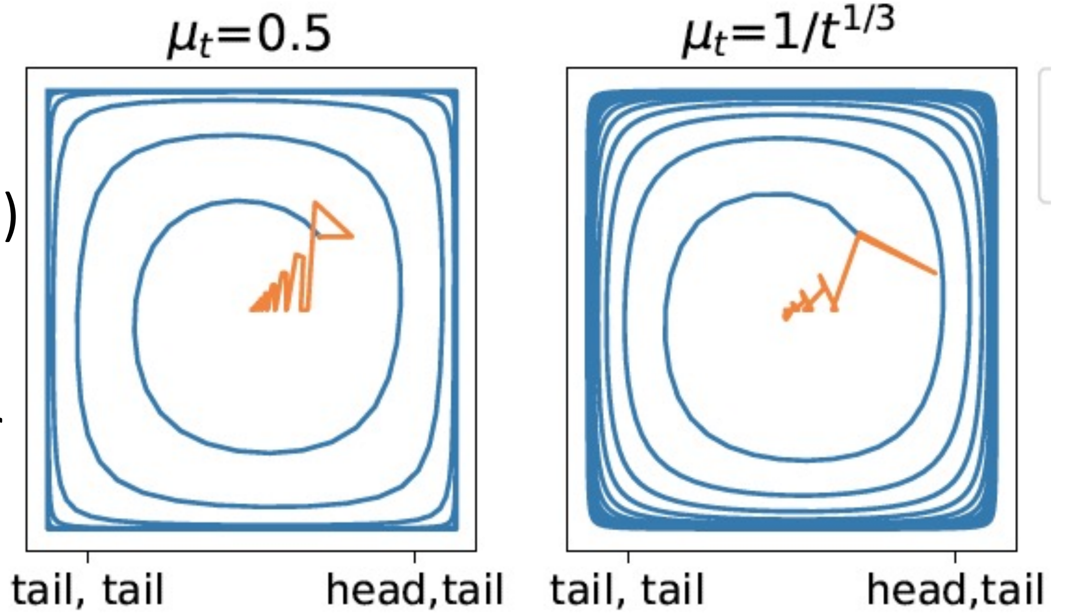
- Agent chooses mix strategy $y_t \in [0,1]^m$ and adversary chooses $x_t \in [0,1]^n$
- Payoff matrix $A \in [0,1]^{n \times m}$: $\max_{\boldsymbol{y} \in \Delta_m} \min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} = \min_{\boldsymbol{x} \in \Delta_n} \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} = v$
- Epsilon-Nash (x^*, y^*) : $|(x^*)^T A y^* - v| \leq \varepsilon$

Last-iterate convergence with asymmetric knowledge

2-player zero-sum + asymmetric information:

- Column player (agent) can estimate her (approximate) minimax strategy

- Row player (adversary) is a no-external-regret learner



At each time step t:

- Agent chooses mix strategy $y_t \in [0,1]^m$ and adversary chooses $x_t \in [0,1]^n$

- Payoff matrix $A \in [0,1]^{n \times m}$: $\max_{\boldsymbol{y} \in \Delta_m} \min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} = \min_{\boldsymbol{x} \in \Delta_n} \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} = v$

- Epsilon-Nash (x^*, y^*) : $|(x^*)^T A y^* - v| \leq \varepsilon$

Goal of the agent: achieve **last-iterate convergence** to (x^*, y^*) **AND no-external-regret**

The LRCA algorithm

Algorithm 1: *Last Round Convergence in Asymmetric algorithm* (LRCA)

Input: Current iteration t, past feedback $x_{t-1}^\top A$ of the row player

Output: Strategy y_t for the column player

if $t = 2k - 1,\ k \in \mathbb{N}$ **then**

$y_t = y^*$

end

if $t = 2k,\ k \in \mathbb{N}$ **then**

$e_t := \text{argmax}_{e \in \{e_1, e_2, \ldots e_m\}} x_{t-1}^\top A e; \quad f(x_{t-1}) := \max_{y \in \Delta_m} x_{t-1}^\top A y$

$\alpha_t := \frac{f(x_{t-1}) - v}{\max\left(\frac{n}{4}, 2\right)}$

$y_t := (1 - \alpha_t) y^* + \alpha_t e_t$

end

Odd time step: play the (approx.) minimax strategy

Even time step: play an adaptive strategy

The LRCA algorithm

Algorithm 1: *Last Round Convergence in Asymmetric algorithm* (LRCA)

Input: Current iteration t, past feedback $x_{t-1}^{\top}A$ of the row player

Output: Strategy y_t for the column player

if $t = 2k - 1,\ k \in \mathbb{N}$ **then**

  $y_t = y^*$

end

if $t = 2k,\ k \in \mathbb{N}$ **then**

  $e_t := \text{argmax}_{e \in \{e_1, e_2, \dots e_m\}} x_{t-1}^{\top}Ae; \quad f(x_{t-1}) := \max_{y \in \Delta_m} x_{t-1}^{\top}Ay$

  $\alpha_t := \frac{f(x_{t-1}) - v}{\max\ (\frac{n}{4}, 2)}$

  $y_t := (1 - \alpha_t)y^* + \alpha_t e_t$

end

Odd time step: play the (approx.) minimax strategy

Even time step: play an adaptive strategy

Notes:

- Playing the approximate Nash repeatedly doesn't achieve no-external-regret
- Playing it up to a constant number of times doesn't help last-iterate convergence

Main result

Theorem:

- If the adversary is a no-external-regret learner, then LRCA achieves $O\left(\sqrt{\log(n)}T^{3/4}\right)$ **dynamic regret + convergence to** (x^*, y^*)
- If adversary uses a constant learning rate μ , the dynamic regret is $O\left(\frac{n}{\sqrt{n}}T^{1/2}\right)$

Main result

Theorem:

- If the adversary is a no-external-regret learner, then LRCA achieves $O\left(\sqrt{\log(n)}T^{3/4}\right)$ **dynamic regret + convergence to** (x^*, y^*)
- If adversary uses a constant learning rate μ , the dynamic regret is $O\left(\frac{n}{\sqrt{n}}T^{1/2}\right)$

Key step:

- Similarly to Topic 1, we want to show that the adversary's behaviour is predictable
- This is more difficult due to the alternating behaviour of the agent

Definition 2 (Kullback and Leibler (1951)) *The relative entropy or K-L divergence between two vectors* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *in* Δ_n *is defined as* $RE(\boldsymbol{x}_1 \| \boldsymbol{x}_2) = \sum_{i=1}^{n} \boldsymbol{x}_1(i) \log\left(\frac{\boldsymbol{x}_1(i)}{\boldsymbol{x}_2(i)}\right)$.

Main result

Theorem:

- If the adversary is a no-external-regret learner, then LRCA achieves $O\left(\sqrt{\log(n)}T^{3/4}\right)$ **dynamic regret + convergence to** (x^*, y^*)
- If adversary uses a constant learning rate μ , the dynamic regret is $O\left(\frac{n}{\sqrt{n}}T^{1/2}\right)$

Key step:

- Similarly to Topic 1, we want to show that the adversary's behaviour is predictable
- This is more difficult due to the alternating behaviour of the agent

Definition 2 (Kullback and Leibler (1951)) *The relative entropy or K-L divergence between two vectors* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *in* Δ_n *is defined as* $RE(\boldsymbol{x}_1 \| \boldsymbol{x}_2) = \sum_{i=1}^{n} \boldsymbol{x}_1(i) \log\left(\frac{\boldsymbol{x}_1(i)}{\boldsymbol{x}_2(i)}\right)$.

Claim: $RE\left(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1}\right) - RE\left(\boldsymbol{x}^* \| \boldsymbol{x}_{2k+1}\right) \geq \frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) \;\; \forall k \in \mathbb{N}: \;\; 2k \geq t'$

3. Exploiting no-external-regret learners via (minimal) payoff manipulation

Payoff manipulation

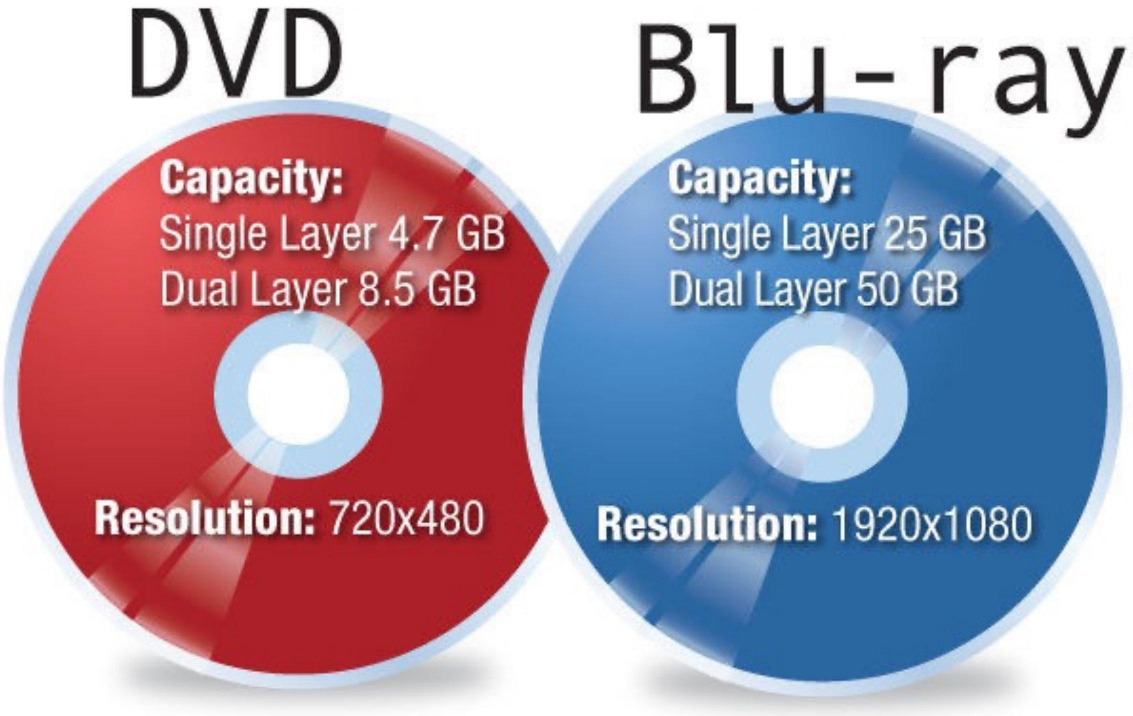
- Data poisoning attacks against bandit and RL agents
- Last-iterate convergence to a given mix strategy profile
- **Learning to win coopetitive games**

What’s coopetitive game?

- In order to win/perform well, one must cooperate with their opponents
- But they also need to know when to stop cooperating to become the winner/achieve their goal
- That is, they need to cooperate and compete at the same time (Nalebuff & Brandenburger, 1996)

https://cruciformstuff.com/2023/07/30/betrayal/

Example 1: Blue-Ray vs. DVD



https://fr.tipard.com/resource/blu-ray-vs-dvd.html

Example 2: Tour de France

https://www.ef.fr/blog/language/les-principaux-termes-de-cyclisme-connaitre-pour-regarder-le-tour-de-france/

Recent interests from the AI Community

Google Deepmind + Cooperative AI Foundation's Melting Pot Challenge (hosted at NeurIPS 2023)

https://www.aicrowd.com/challenges/meltingpot-challenge-2023

Research questions

In AI, we consider a multi-agent sequential decision-making version of coopetitive games:

- Who to cooperate with?

- How to signal/incentivise others to collaborate

- When to switch side?

Our focus

- Aim: Proof of Concept
- Simplified setting
- 3 players
- Repeated games
- Polymatrix games
- Signaling: payoff manipulation

Payoff manipulation explained

- In our setting no explicit communication between agents is allowed

- Instead, we allow one agent to modify another agent's payoff by:

    - Sacrificing from their own payoffs (e.g., gift, bribery, etc) -> increasing the other's payoff

    - Enforce some penalties -> decreasing opponent's payoff

    - Examples: multiplayer video games, nature, etc.

Problem formulation

- 3 players: P1, P2, P3 (we are P1) – repeated game (each round they play the same game)
- Polymatrix game:
  - Game can be decomposed to sum or pairwise 2-player games
  - Payoff = sum of pairwise payoffs defined by pairwise payoff matrices $A^{(i,j)}$
- Payoff manipulation: P1 can modify $A^{(2,1)}$ and $A^{(3,1)}$
- Payoff of P1:

$$x^T A^{(1,2)} y + x^T A^{(1,3)} z - ||M^{(2,1)} - A^{(2,1)}||_\infty - ||M^{(3,1)} - A^{(3,1)}||_\infty$$

- Payoff of P2 & P3:

$$y^T M^{(2,1)} x + y^T A^{(2,3)} z$$

$$z^T M^{(3,1)} x + z^T A^{(3,2)} y$$

Winning policies

Objective: P1 will have higher total/average payoff than P2 and P3

Idea: We are interested in a certain type of behaviour (policy) that can lead to winning the game

- Suppose P1 plays i* action for all the rounds
- Suppose P2 has a **strictly dominant** strategy j* against i*, similarly P3 has a **strictly dominant** strategy k* against i*
- Also, suppose $u_1(i^*, j^*, k^*) > \max\left\{u_2(i^*, j^*, k^*), u_3(i^*, j^*, k^*)\right\}$
- Then by consistently playing i*, P1 would eventually *win the game*

Winning policies

Objective: P1 will have higher total/average payoff than P2 and P3

Idea: We are interested in a certain type of behaviour (policy) that can lead to winning the game

- Suppose P1 plays i* action for all the rounds
- Suppose P2 has a **strictly dominant** strategy j* against i*, similarly P3 has a **strictly dominant** strategy k* against i*
- Also, suppose $u_1(i^*, j^*, k^*) > \max\left\{u_2(i^*, j^*, k^*), u_3(i^*, j^*, k^*)\right\}$
- Then by consistently playing i*, P1 would eventually *win the game*

Issue: such situation does not always exist ☹

Winning policies

Objective: P1 will have higher total/average payoff than P2 and P3

Idea: We are interested in a certain type of behaviour (policy) that can lead to winning the game

- Suppose P1 plays i* action for all the rounds
- Suppose P2 has a **strictly dominant** strategy j* against i*, similarly P3 has a **strictly dominant** strategy k* against i*
- Also, suppose $u_1(i^*, j^*, k^*) > \max\left\{u_2(i^*, j^*, k^*), u_3(i^*, j^*, k^*)\right\}$
- Then by consistently playing i*, P1 would eventually *win the game*

Issue: such situation does not always exist ☹

Solution: create such solution via (minimal) payoff matrix manipulation!!! ☺

Existence of dominant solvable games

Goal: Design a game via (optimally) manipulating $M^{(2,1)}$ and $M^{(3,1)}$ such that P2 has a **strictly dominant** strategy j* against i*, similarly P3 has a **strictly dominant** strategy k* against i* (for some i* action of P1)

Existence of dominant solvable games

Goal: Design a game via (optimally) manipulating $M^{(2,1)}$ and $M^{(3,1)}$ such that P2 has a **strictly dominant** strategy j* against i*, similarly P3 has a **strictly dominant** strategy k* against i* (for some i* action of P1)

Result 1: such dominant solvable game exists for any original 3-player polymatrix games

Even more, if we fix i*, j*, and k* in advance -> there exists a dominant solvable game for (i*,j*,k*)

Existence of dominant solvable games

Goal: Design a game via (optimally) manipulating $M^{(2,1)}$ and $M^{(3,1)}$ such that P2 has a **strictly dominant** strategy j* against i*, similarly P3 has a **strictly dominant** strategy k* against i* (for some i* action of P1)

Result 1: such dominant solvable game exists for any original 3-player polymatrix games

Even more, if we fix i*, j*, and k* in advance -> there exists a dominant solvable game for (i*,j*,k*)

Issue 1: How to achieve $u_1(i^*, j^*, k^*) > \max\left\{u_2(i^*, j^*, k^*), u_3(i^*, j^*, k^*)\right\}$

Issue 2: What happens if P2 and P3 are learning agents?

Consistent agents

Definition 1. *(Consistent Agent) Suppose that for an agent there exists an action a^* that is the unique best response for her for every round of the game. Suppose that within T rounds of the game, the number of rounds the agent plays action a^* is T^*. If $\mathbb{P}\left(\lim_{T\to\infty}\frac{T^*}{T}=1\right)=1$ then the agent is 'consistent'.*

Consistent agent:

- There is a same fixed best action for that agent in **every round**
- Event: the fraction of number of times the agent plays this best action tends to 1
- Probability of this event = 1

Persistent agents

Definition 4. *(Persistent Agent) Suppose that the action k^* is the best action in hindsight for player 3 eventually, with probability 1. That is,*

$$\mathbb{P}\Big(\boldsymbol{e}_{k^*} = \arg\max_{\boldsymbol{z}\in\Delta_l} U_3(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z})_{t=1}^{T} \ \textit{eventually}\Big) = 1$$

Let T^* denote the number of rounds within T rounds, that player 3 plays action k^*. If $\mathbb{P}\Big(\lim_{T\to\infty} \frac{T^*}{T} = 1\Big) = 1$ then player 3 is 'persistent'.

Persistent agent:

- There is a same fixed best action for that agent **from some round** (i.e., eventually)
- Event: the fraction of number of times the agent plays this best action tends to 1
- Probability of this event = 1

Persistent agents

Definition 4. *(Persistent Agent) Suppose that the action k^* is the best action in hindsight for player 3 eventually, with probability 1. That is,*

$$\mathbb{P}\Big(\boldsymbol{e}_{k^*} = \arg\max_{\boldsymbol{z}\in\Delta_l} U_3(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z})_{t=1}^T \ \textit{eventually}\Big) = 1$$

Let T^* denote the number of rounds within T rounds, that player 3 plays action k^*. If $\mathbb{P}\Big(\lim_{T\to\infty} \frac{T^*}{T} = 1\Big) = 1$ then player 3 is 'persistent'.

Persistent agent:

- There is a same fixed best action for that agent **from some round** (i.e., eventually)
- Event: the fraction of number of times the agent plays this best action tends to 1
- Probability of this event = 1

Proposition 2. *All persistent players are consistent. Further, all no-regret players are persistent.*

Main results

Winning dominance solvable policies:

- Each action of P1= $(a_t^1, M_t^{(2,1)}, M_t^{(3,1)})$
- Makes P1 is the winner of the resulting dominant solvable game

Main results

Winning dominance solvable policies:

- Each action of P1= $(a_t^1, M_t^{(2,1)}, M_t^{(3,1)})$
- Makes P1 is the winner of the resulting dominant solvable game

Theorem 1: If P2 and P3 are consistent agents then there exists a winning dominance solvable policy for P1

Theorem 2: If P2 is consistent and P3 is persistent, then there exists a winning dominance solvable policy for P1

Theorem 3: These winning dominance solvable policies, if exist, can be calculated in polynomial running time

Additional objectives

- Winning by largest margin
- Winning by lowest inefficiency ratio
- Maximising the egalitarian social welfare

Winning by largest margin

Margin of P1:

$$\min\left\{\mathbb{E}\left[U_1(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty} - U_2(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty}\right], \mathbb{E}\left[U_1(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty} - U_3(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty}\right]\right.$$

- How much better the (expected) average payoff of P1 is compared to the others’

Winning by largest margin

Margin of P1:

$$\min \left\{ \mathbb{E}\left[U_1(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty} - U_2(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty} \right], \mathbb{E}\left[U_1(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty} - U_3(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)_{t=1}^{\infty} \right] \right.$$

- How much better the (expected) average payoff of P1 is compared to the others'

Theorem 6: *If winning dominance solvable policies exist*, then there exists an algorithm that can find the **largest margin dominance solvable policy**, with running time that is polynomial in the number of actions of the players.

Winning by lowest inefficiency ratio

Inefficiency ratio: the ratio between the **cost for modifying the payoff matrices** and the **expected increase in long run payoffs** from the worst-case payoff.

$$\frac{\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{(i,j)\in P} \|A_t^{(i,j)} - A_0^{(i,j)}\|_\infty}{\mathbb{E}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\mathbf{x}_t^T A_t^{(1,2)} \mathbf{y}_t + \mathbf{x}_t^T A_t^{(1,3)} \mathbf{z}_t\right)\right] - K}$$

where $K = \min_{i,j,k} \left(A^{(1,2)}(i,j) + A^{(1,3)}(j,k)\right)$ *is the minimum revenue for player 1.*

Winning by lowest inefficiency ratio

Inefficiency ratio: the ratio between the **cost for modifying the payoff matrices** and the **expected increase in long run payoffs** from the worst-case payoff.

$$\frac{\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{(i,j)\in P}\|A_t^{(i,j)} - A_0^{(i,j)}\|_\infty}{\mathbb{E}\Big[\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\Big(\mathbf{x}_t^T A_t^{(1,2)}\mathbf{y}_t + \mathbf{x}_t^T A_t^{(1,3)}\mathbf{z}_t\Big)\Big] - K}$$

where $K = \min_{i,j,k}\left(A^{(1,2)}(i,j) + A^{(1,3)}(j,k)\right)$ *is the minimum revenue for player 1.*

Theorem: *If winning dominance solvable policies exist*, then there exists an algorithm that can find the **winning dominance solvable policy with the lowest inefficiency ratio**, with running time that is polynomial in the number of actions of the players.

Maximising egalitarian social welfare

Egalitarian social welfare: The lowest payoff among the players'

Definition 9. *The Egalitarian Social Welfare of a strategy profile* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ *is defined to be*

$$\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) := \min \left\{ U_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), U_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), U_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \right\}$$

Maximising egalitarian social welfare

Egalitarian social welfare: The lowest payoff among the players'

Definition 9. *The Egalitarian Social Welfare of a strategy profile* $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ *is defined to be*

$$\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) := \min \left\{ U_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), U_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), U_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \right\}$$

Theorem: There exists an algorithm that can find the dominance solvable policy that **maximizes egalitarian social welfare** with running time that is polynomial in the number of actions of the players.

Application 1: 3-Player iterated prisoner's dilemma

Action space = {C, D}

$$A_0^{(i,j)} = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix} \text{ if } i < j \text{ and } A_0^{(i,j)} = \begin{bmatrix} 3 & 5 \\ 0 & 1 \end{bmatrix} \text{ if } i > j$$

Application 1: 3-Player iterated prisoner's dilemma

Action space = {C, D}

$$A_0^{(i,j)} = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix} \text{ if } i < j \text{ and } A_0^{(i,j)} = \begin{bmatrix} 3 & 5 \\ 0 & 1 \end{bmatrix} \text{ if } i > j$$

For P1, a winning strategy would be always playing D (and both P2 and P3 also defect all the time)

- But this one has 0 margin as well
- Can we design a better policy with positive margin, and **incentivises cooperation**?

Application 1: 3-Player iterated prisoner's dilemma

Action space = {C, D}

$$A_0^{(i,j)} = \begin{bmatrix} 3 & 0 \\ 5 & 1 \end{bmatrix} \text{ if } i < j \text{ and } A_0^{(i,j)} = \begin{bmatrix} 3 & 5 \\ 0 & 1 \end{bmatrix} \text{ if } i > j$$

For P1, a winning strategy would be always playing D (and both P2 and P3 also defect all the time)

- But this one has 0 margin as well
- Can we design a better policy with positive margin, and **incentivises cooperation**?

We show that for $0 < \varepsilon \leq \frac{7}{6}$ we set $\hat{A} = \begin{bmatrix} 3 & 5 \\ 3/2 + \epsilon & -1/2 \end{bmatrix}$

P1 plays D and manipulates opponents' payoff matrices to $\hat{A}$

Theorem: system will converge to (D,C,C) and P1 wins with large (positive) margin

Application 2: social distancing game

Inspired by Zinkevic's Lemonade Stand Game

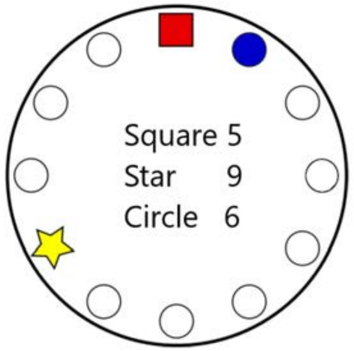


Fig. 1. Example Social Distancing Game

Application 2: social distancing game

Inspired by Zinkevic's Lemonade Stand Game

Winning the game:

Theorem 1: P1 can win the game with **negligible manipulation cost**

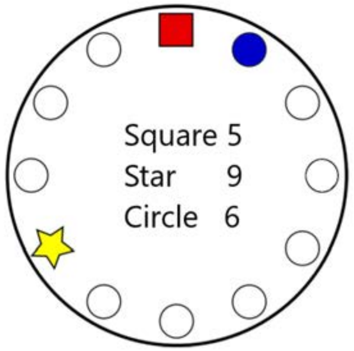


Fig. 1. Example Social Distancing Game

Application 2: social distancing game

Inspired by Zinkevic’s Lemonade Stand Game

Winning the game:

Theorem 1: P1 can win the game with **negligible manipulation cost**

Egalitarian social welfare:

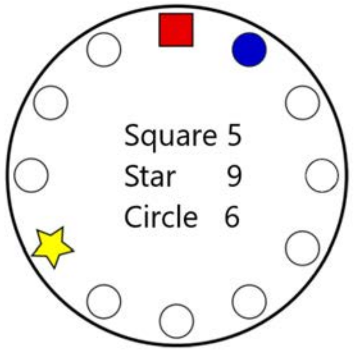


Fig. 1. Example Social Distancing Game

Application 2: social distancing game

Inspired by Zinkevic's Lemonade Stand Game

Winning the game:

Theorem 1: P1 can win the game with **negligible manipulation cost**

Egalitarian social welfare:

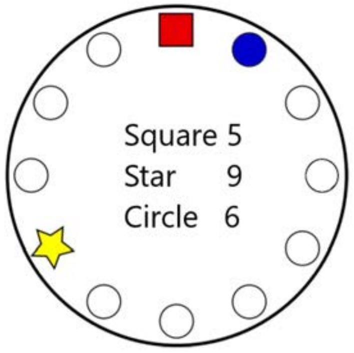


Fig. 1. Example Social Distancing Game

$$\hat{A}(k,l) = \begin{cases} d(k,l) & \text{if } k \neq 12 \\ d(k,l) - 1 - 2\epsilon & \text{if } k = 12 \text{ and } l \neq 5 \\ d(k,l) + 1 - \epsilon & \text{if } k = 12 \text{ and } l = 5 \end{cases}$$

$$\tilde{A}(k,l) = \begin{cases} d(k,l) & \text{if } k \neq 12 \\ d(k,l) - 1 + \epsilon & \text{if } k = 12 \text{ and } l \neq 7 \\ d(k,l) + 1 - \epsilon & \text{if } k = 12 \text{ and } l = 7 \end{cases}$$

Application 2: social distancing game

Inspired by Zinkevic's Lemonade Stand Game

Winning the game:

Theorem 1: P1 can win the game with **negligible manipulation cost**

Egalitarian social welfare:

Theorem 2: P1 plays position 12 and use $\hat{A}$ and $\tilde{A}$ to manipulate the payoff of P2 and P3, then the **egalitarian social welfare is maximised**

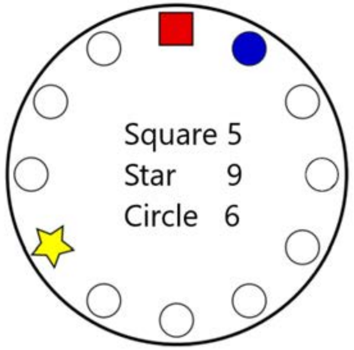


Fig. 1. Example Social Distancing Game

$$\hat{A}(k,l) = \begin{cases} d(k,l) & \text{if } k \neq 12 \\ d(k,l) - 1 - 2\epsilon & \text{if } k = 12 \text{ and } l \neq 5 \\ d(k,l) + 1 - \epsilon & \text{if } k = 12 \text{ and } l = 5 \end{cases}$$

$$\tilde{A}(k,l) = \begin{cases} d(k,l) & \text{if } k \neq 12 \\ d(k,l) - 1 + \epsilon & \text{if } k = 12 \text{ and } l \neq 7 \\ d(k,l) + 1 - \epsilon & \text{if } k = 12 \text{ and } l = 7 \end{cases}$$

Summary

- No-regret learner’s behaviour is predictable
- Better regret bounds against no-regret learners (Topic 1)
- Last-iterate convergence under information asymmetry (Topic 2)
- Easy to manipulate their behaviour with minimal manipulation cost (Topic 3)

Open questions

- Topic 1 (better regret bounds):
  - extend to (episodic) RL, online MDPs, stochastic games

- Topic 2 (last-iterate convergence):
  - Relax the information asymmetry assumption;
  - How frequently we need to play the approximate Nash

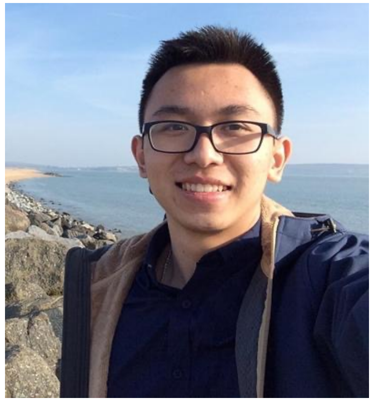
- Topic 3 (minimal manipulation cost):
  - Optimal manipulation schemes?
  - N-player games ($N > 3$)
  - General games (not polymatrix)?

Online version of our papers

- Topic 1: https://arxiv.org/abs/2302.06652
- Topic 2: https://proceedings.mlr.press/v132/dinh21a.html
- Topic 3: https://arxiv.org/abs/2110.13532

Many thanks for your attention

Nick Bishop



Le Cong Dinh

Shiva Mahesh