Algorithmic Game Theory
Algorithms for 0-sum games

Vangelis Markakis
markakis@gmail.com

Nash equilibria: Computation

- Nash's theorem only guarantees the existence of Nash equilibria
  - Proof via Brouwer's fixed point theorem
- The proof does not imply an efficient algorithm for computing equilibria
  - Because we do not have efficient algorithms for finding fixed points of continuous functions
- Can we design polynomial time algorithms for 2-player games?
  - For games with more players?

Zero-sum Games

A special case: 0-sum games

- Games where for every profile (s_i, t_j) we have

  $$u_1(s_i, t_j) + u_2(s_i, t_j) = 0$$

  – The payoff of one player is the payment made by the other

4	2
1	3

- Also referred to as strictly competitive
- It suffices to use only the matrix of player 1 to represent such a game

- How should we play in such a game?

A special case: 0-sum games

- Idea: Pessimistic play
- Assume that no matter what you choose the other player will pick the worst outcome for you
- Reasoning of player 1:
  - If I pick row 1, in worst case I get 2
  - If I pick row 2, in worst case I get 1
  - I will pick the row that has the best worst case
  - Payoff = $\max_i \min_j A_{ij} = 2$
- Reasoning of player 2:
  - If I pick column 1, in worst case I pay 4
  - If I pick column 2, in worst case I pay 3
  - I will pick the column that has the smallest worst case payment
  - Payment = $\min_j \max_i A_{ij} = 3$

4	2
1	3

0-sum games

Definitions

- For pl. 1:
  - The best of the worst-case scenarios:

    $v_1 = \max_i \min_j A_{ij}$
  - We take the minimum of each row and select the best minimum
- For pl. 2:
  - Again the best of the worst-case scenarios

    $v_2 = \min_j \max_i A_{ij}$
  - We take the max in each column and then select the best maximum
- In the example:
  - $v_1 = 2$, $v_2 = 3$
- The game also does not have pure Nash equilibria

Example 2

- Computing v_1 for pl. 1:
  - Row 1, min = 4
  - Row 2, min = 1
  - Row 3, min = 0
  - Row 4, min = 4
  - v_1 = max {4, 1, 0, 4} = 4
- Computing v_2 for pl. 2:
  - Column 1, max = 4
  - Column 2, max = 6
  - Column 3, max = 7
  - Column 4, max = 4
  - v_2 = min {4, 6, 7, 4} = 4

	t_1	t_2	t_3	t_4
s_1	4	5	6	4
s_2	2	6	1	3
s_3	1	0	0	2
s_4	4	4	7	4

Example 2

- In contrast to the first example, here we have $v_1 = v_2$
- Recommended strategies:
  - s_1 or s_4 for pl. 1
  - t_1 or t_4 for pl. 2
- Pessimistic play can lead to 4 different profiles
- Observations:
  i. Same utility in all 4 profiles
  ii. All 4 profiles are Nash equilibria!
  iii. There is no other Nash equilibrium

	t_1	t_2	t_3	t_4
s_1	4	5	6	4
s_2	2	6	1	3
s_3	1	0	0	2
s_4	4	4	7	4

Nash equilibria in 0-sum games

Theorem: For every finite 2-player 0-sum game:

- $v_1 \leq v_2$
- There exists a Nash equilibrium with pure strategies if and only if $v_1 = v_2$
- If (s, t) and (s', t') are pure equilibria, then the profiles (s, t'), (s', t) are also equilibria
- When we have multiple Nash equilibria, the utility is the same for both players in all equilibria (v_1 for pl. 1 and $-v_1$ for pl. 2)

Corollary: In games where $v_1 < v_2$, there is no Nash equilibrium with pure strategies

Nash equilibria in 0-sum games

- In general $v_1 \neq v_2$
- Pessimistic play with pure strategies does not always lead to a Nash equilibrium
- Idea (von Neumann): Use pessimistic play with mixed strategies!
- Definitions:
  - $w_1 = \max_{\mathbf{p}} \min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q})$
  - $w_2 = \min_{\mathbf{q}} \max_{\mathbf{p}} u_1(\mathbf{p}, \mathbf{q})$
- We can easily show that: $v_1 \leq w_1 \leq w_2 \leq v_2$
  - Because we are optimizing over a larger strategy space
- How can we compute w_1 and w_2?

Back to Example 1

- We will find first $w_1 = \max_{\mathbf{p}} \min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q})$
- We need to look for a strategy $\mathbf{p} = (p_1, p_2) = (p_1, 1 - p_1)$ of pl. 1
- We need to look better at the 2 consecutive optimization steps
- Lemma: Given a strategy **p** of pl. 1, the term $\min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q})$ is minimized at a pure strategy of pl. 2
  - Hence, no need to have both optimization steps over mixed strategies

4	2
1	3

Analysis of Example 1

- The lemma simplifies the process as follows:

$w_1 = \max_{\mathbf{p}} \min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q})$

$= \max_{\mathbf{p}} \min\{ u_1(\mathbf{p}, e^1), u_1(\mathbf{p}, e^2) \}$

$= \max_{p1} \min\{ 4p_1 + 1\text{-}p_1, 2p_1 + 3(1\text{-}p_1) \}$

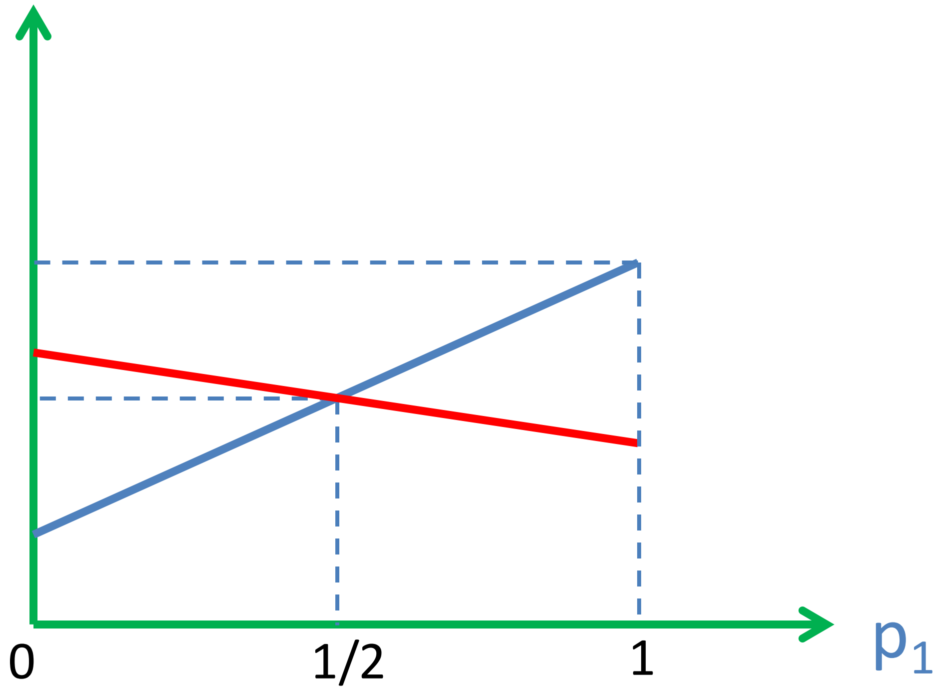
$= \max_{p1} \min\{ 3p_1 + 1, 3 – p_1 \}$

4	2
1	3

Analysis of Example 1

- $w_1 = \max_{p1} \min \{ 3p_1 + 1, 3 - p_1 \}$
- We need to maximize the minimum of 2 lines

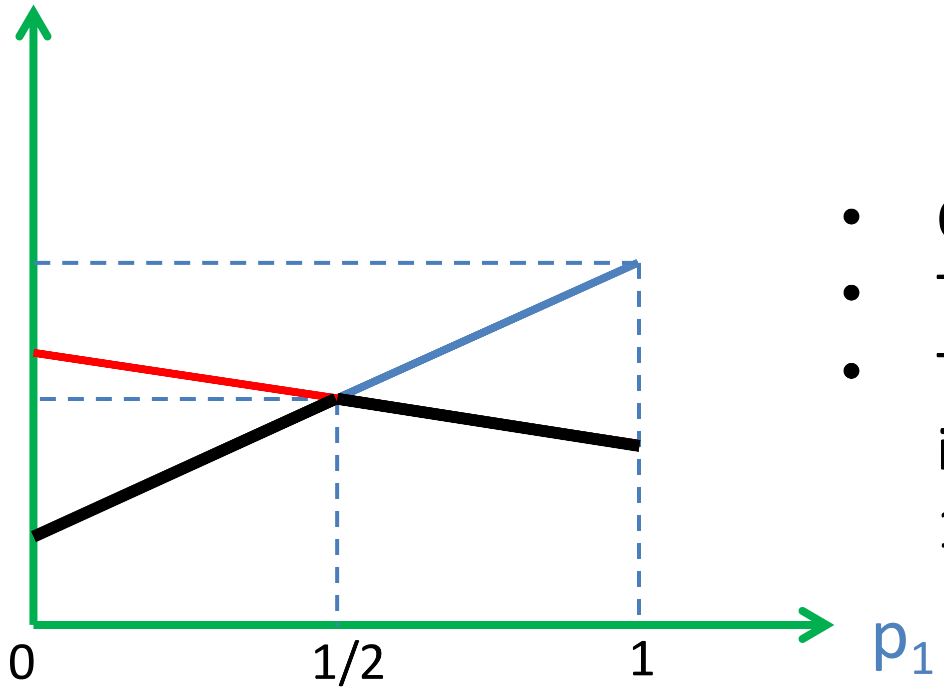
4	2
1	3



Analysis of Example 1

- $w_1 = \max_{p1} \min \{ 3p_1 + 1, 3 - p_1 \}$
- We need to maximize the minimum of 2 lines

4	2
1	3



- One line is increasing
- The other is decreasing
- The min. is achieved at the intersection point ➔ p_1 = 1/2

Analysis of Example 1

Summing up:

- $w_1 = \max_{\mathbf{p}} \min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q}) = \max_{p1} \min \{ 3p_1 + 1, 3 - p_1 \} = 3*1/2 + 1 = 5/2$
- If pl. 1 plays strategy **p** = (1/2, 1/2), he can guarantee on average 5/2, independent of the choice of pl. 2
- Thus, with mixed strategies, pessimistic play provides a better guarantee than with pure ($v_1 = 2 < 2.5$)

4	2
1	3

Analysis of Example 1

With a similar analysis for pl. 2:

$w_2 = \min_{\mathbf{q}} \max_{\mathbf{p}} u_1(\mathbf{p}, \mathbf{q})$

$= \min_{\mathbf{q}} \max\{ u_1(e^1, \mathbf{q}), u_1(e^2, \mathbf{q}) \}$

$= \min_{q1} \max\{ 4q_1 + 2(1\text{-}q_1), q_1 + 3(1\text{-}q_1) \}$

$= \min_{q1} \max\{ 2q_1 + 2, 3 – 2q_1 \}$

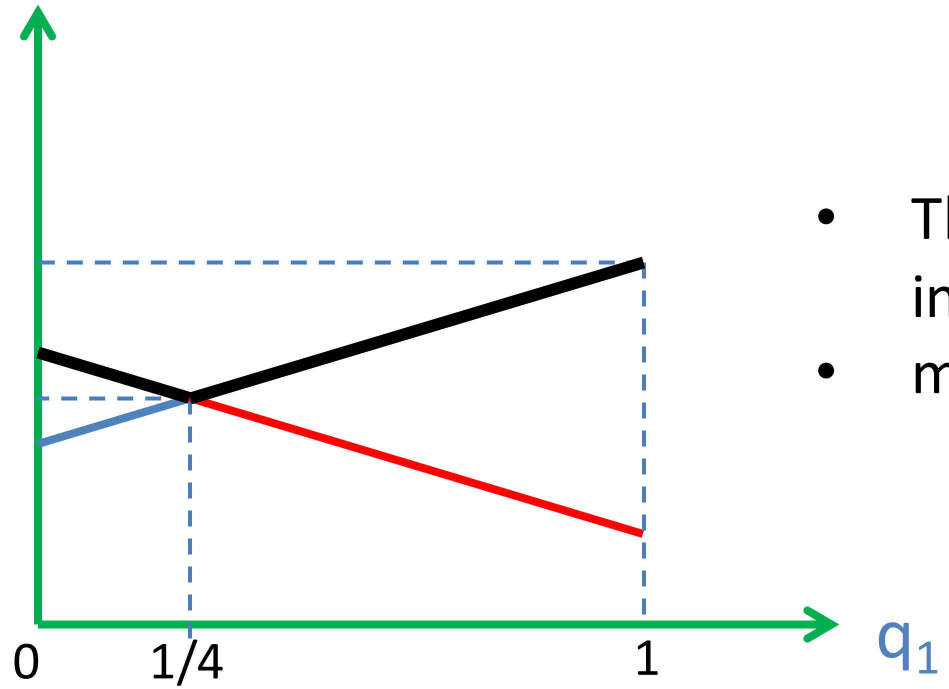
4	2
1	3

- We now want to minimize the max among 2 lines

Analysis of Example 1

- $w_2 = \min_{q1} \max\{ 2q_1 + 2, 3 - 2q_1 \}$
- Again, one is increasing, the other is decreasing

4	2
1	3



- The max. is achieved at the intersection point ➔ $q_1 = 1/4$
- min-max strategy: (1/4, 3/4)

Analysis of Example 1

Final conclusions:

- We found the profile
  - $\mathbf{p} = (1/2, 1/2)$, $\mathbf{q} = (1/4, 3/4)$
- $w_1 = w_2 = 5/2$
- Both players guarantee something better to themselves by using mixed strategies
- With pure strategies:

$$\max_i \min_j A_{ij} \neq \min_j \max_i A_{ij}$$

- With mixed strategies, we have equality

$$\max_{\mathbf{p}} \min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{\mathbf{p}} u_1(\mathbf{p}, \mathbf{q})$$

- Also, $(\mathbf{p}, \mathbf{q})$ is a Nash equilibrium! (check)

4	2
1	3

Nash equilibria in 0-sum games

<u>Theorem</u> (von Neumann, 1928): For every finite 2-player 0-sum game:

1. $w_1 = w_2$ (referred to as the value of the game)
2. The profile (**p**, **q**), where w_1 and w_2 are achieved forms a Nash equilibrium
3. If (**p**, **q**) and (**p**', **q**') are equilibria, then the profiles (**p**, **q'**), (**p'**, **q**) are also equilibria
4. In every Nash equilibrium, the utility to each player is the same (w_1 for pl. 1 and $-w_1$ for pl. 2)

Nash equilibria in 0-sum games

Conclusions from von Neumann's theorem

- For the family of 2-player 0-sum games, all the problematic issues we had identified for normal form games are resolved
  - Existence: guaranteed
  - Non-uniqueness: not a problem, because all equilibria yield the same utility to each player
  - If there are multiple equilibria, all of them are equally acceptable

Nash equilibria in 0-sum games

Computation of Nash equilibria

- Till now we saw how to find Nash equilibria in 2x2 0-sum games
- The previous reasoning cannot be generalized
  - We get problems with more variables, cannot visualize as before
- Can we find an equilibrium for arbitrary nxm 0-sum games?

Nash equilibria in 0-sum games

- We need a different approach
- Initial proof of von Neumann's theorem (1928) is not constructive
  - Based on fixed point theorems
- Fortunately: there is an alternative algorithmic proof of existence
- Finding w_1 and the strategy of pl. 1 can be modeled as a linear programming problem
- Finding the equilibrium strategy of pl. 2 can be modeled as the **dual** problem to that of pl. 1

Linear Programming

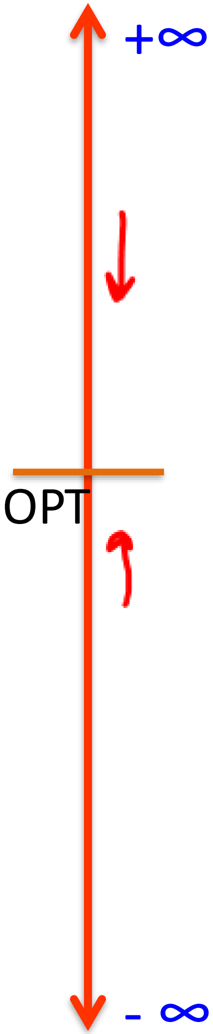
- What is a linear program?
- Any optimization problem where
  - The objective function is linear
  - The constraints are also linear

$$
\begin{aligned}
&\text{maximize } Z(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
&\text{subject to:} \\
&\qquad a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \\
&\qquad a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \\
&\qquad \vdots \\
&\qquad a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m \\
&\qquad x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0
\end{aligned}
$$

- We can also have inequalities with ≥ or equalities in the constraints
- We can solve linear programs very fast, even with hunderds of variables and constraints (Matlab, AMPL,...)

Linear Programming

- Basic component for the alternative proof of von Neumann's theorem:
- Duality theorem: For every maximization LP, there is a corresponding dual minimization LP such that
  - The primal LP has an optimal solution iff the dual LP has an optimal solution
  - The optimal value (when it exists) for both the primal and the dual LP is the same



Nash equilibria in 0-sum games

- Consider a 0-sum game with an nxm matrix A for pl. 1
- Recall: $w_1 = \max_{\mathbf{p}} \min_{\mathbf{q}} u_1(\mathbf{p}, \mathbf{q}) = \max_{\mathbf{p}} \min \{u_1(\mathbf{p}, e^k)\}_{k=1,\ldots,m}$
- LP-based proof of von Neumann's theorem: The max-min and the min-max strategies of pl. 1 and pl. 2 are obtained by solving the linear programs:

$u_1(p, e^k)$

$$
\begin{aligned}
\max \quad & w \\
\text{s. t.:} \quad & \\
& w \leq \sum_{i=1}^{n} A_{ik}\, p_i, \forall k = 1, \ldots, m \\
& \sum_{i=1}^{n} p_i = 1 \\
& p_i \geq 0, \qquad \forall i = 1, \ldots, n
\end{aligned}
$$

Primal LP

$$
\begin{aligned}
\min \quad & w \\
\text{s. t.:} \quad & \\
& w \geq \sum_{j=1}^{m} A_{ij}\, q_j, \forall i = 1, \ldots, n \\
& \sum_{j=1}^{m} q_j = 1 \\
& q_j \geq 0, \qquad \forall j = 1, \ldots, m
\end{aligned}
$$

Dual LP

Example

- $v_1 = 3$, $v_2 = 5$, no pure Nash equilibrium
- We have to use linear programming to find the equilibrium profile

	t_1	t_2	t_3	t_4
s_1	6	5	3	5
s_2	1	2	6	4
s_3	3	8	3	2

Primal LP

max w

s.t.

$w \leq 6p_1 + p_2 + 3p_3$

$w \leq 5p_1 + 2p_2 + 8p_3$

$w \leq 3p_1 + 6p_2 + 3p_3$

$w \leq 5p_1 + 4p_2 + 2p_3$

$p_1 + p_2 + p_3 = 1$

$p_1, p_2, p_3 \geq 0$

Dual LP

min w

s.t.

$w \geq 6q_1 + 5q_2 + 3q_3 + 5q_4$

$w \geq q_1 + 2q_2 + 6q_3 + 4q_4$

$w \geq 3q_1 + 8q_2 + 3q_3 + 2q_4$

$q_1 + q_2 + q_3 + q_4 = 1$

$q_1, q_2, q_3, q_4 \geq 0$

Summary on 0-sum games

- There always exists a Nash equilibrium in finite 0-sum games, when we allow mixed strategies
- $w_1 = w_2$ = value of the game
- If there are multiple equilibria, they all have the same utility for each player (w_1 for pl. 1, $-w_1$ for pl. 2)
- The value of the game as well as the equilibrium profile can be computed in polynomial time by solving a pair of primal and dual linear programs

0-sum games and optimization

Further connections with Computer Science and Algorithms:

1. Every linear program is "equivalent" to solving a 0-sum game
   - Finding the optimal solution to any linear program can be reduced to finding an equilibrium in some 0-sum game
   - Initially stated in [Dantzig '51], complete proof in [Adler '13]
2. Every problem solvable in polynomial time (class **P**), can be reduced to linear programming, and hence to finding a Nash equilibrium in some appropriately constructed 0-sum game!

0-sum games and complexity classes

Class **P** ⇔ 0-sum games

Shortest paths, minimum spanning trees, sorting, …

Matching Pennies, Rock-Paper-Scissors, …

And some more observations

- Anything we have seen so far also hold for constant-sum games
- In a constant-sum game, for every profile (s, t) with $s \in S^1$, $t \in S^2$

  $u_1(s, t) + u_2(s, t) = c$, for some parameter c

- WHY?
  - We can subtract c from the payoff matrix of pl. 1 (or pl. 2 but not both), so as to convert it to a 0-sum game
  - Adding/subtracting the same parameter from every cell of a payoff matrix do not change the set of Nash equilibria

Learning Algorithms for Zero-sum Games

0-sum games and learning

Machine learning applications: deep learning models, training GANs (Generative Adversarial Networks), boosting, etc

- [Goodfellow et al ’14]: training 2 antagonistic models (the Generator and the Discriminator) can be seen as a 0-sum game
- [Schuurmans, Zinkevich ’16]: deep learning games, reducing supervised learning to game playing
- [Freund, Schapire ’96]: boosting via no-regret dynamics for solving 0-sum games

0-sum games and learning

- Especially for GANs, it is infeasible to use a linear program to do the training
- What are we after then?
  - Iterative learning algorithms that converge to an (approximate) equilibrium
- What do people use in practice?
  - Some versions of Stochastic Gradient Descent
- Any hope for better methods?
  - YES! Better performance and theoretical guarantees for some variations of Gradient Descent
  - Extra gradient, Optimistic gradient and their analogues for Multiplicative Weight Update methods

Min-max optimization

- The problem we are interested in:

$$\min_y \max_x f(x, y)$$

Subject to:

- $x = (x_1, x_2, \ldots, x_n)$, is a probability distribution, $\Sigma_i x_i = 1$, $x_i \geq 0$
- $y = (y_1, y_2, \ldots, y_n)$ is a probability distribution
- $f(x, y)$ is bilinear: $f(x, y) = \Sigma_i \Sigma_j R_{ij} x_i y_j$

Further variations/generalizations

- The domain of x and y may be some different convex set
- Or they can be unconstrained (domain = $R^n \times R^n$)
- $f(x, y)$ can range from bilinear to convex-concave or to more arbitrary smooth, non-convex, non-concave functions

Gradient Descent and Multiplicative Weights Update methods

Descent methods

- First thoughts for solving the problem: use gradient descent/ascent (GDA)
- Caution: need to project to the simplex

$$x^t = \mathbf{\Pi}_\Delta[x^{t-1} + \eta\nabla f(x^{t-1})] = \text{argmin}_{x\in\Delta} \|x - [x^{t-1} + \eta\nabla f(x^{t-1})]\|$$

$$y^t = \mathbf{\Pi}_\Delta [y^{t-1} - \eta\nabla f(y^{t-1})] = \text{argmin}_{y\in\Delta} \|y - [y^{t-1} - \eta\nabla f(y^{t-1})]\|$$

- For optimization over the simplex, we can opt for better algorithms
- Can we adapt gradient descent to the "geometry" of our problem?

Multiplicative Weights Update Method

- One of the most known learning algorithms [Littlestone, Warmuth '94, Fudenberg, Levine '95, Freund, Schapire '99]
- It can be interpreted as the Mirror-Descent method with entropic regularization [Nemirovski, Yudin '83]
- Main intuition of MWU:
  - In each iteration, reward the pure strategies that perform better against the opponent's strategy in the previous iteration
- Several other variations in the literature (e.g. linear instead of exponential updates)
- Also known by different names: FTRL, Hedge,…

Multiplicative Weights Update Method

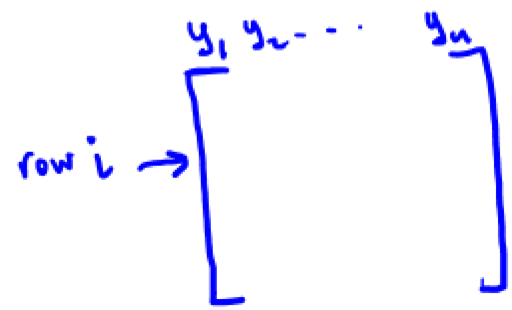
Dynamics for MWU:

$$x_i^t = x_i^{t-1} \cdot \frac{e^{\eta \cdot f(e_i, y^{t-1})}}{\sum_{j=1}^{n} x_j^{t-1} e^{\eta \cdot f(e_j, y^{t-1})}}, \quad y_j^t = y_j^{t-1} \cdot \frac{e^{-\eta \cdot f(x^{t-1}, e_j)}}{\sum_{i=1}^{n} y_i^{t-1} e^{-\eta \cdot f(x^{t-1}, e_i)}}$$

Some notation:

- η = learning rate parameter (step size)
- Let e_i = (0, 0,…,1, 0,…,0) be the *i*-th pure strategy
- $f(e_i, y^{t-1})$ = payoff of row player against y^{t-1}, when selecting the i-th row

$$f(e_i, y^{t-1}) = \sum_{j=1}^{n} R_{ij} \cdot y_j^{t-1}$$



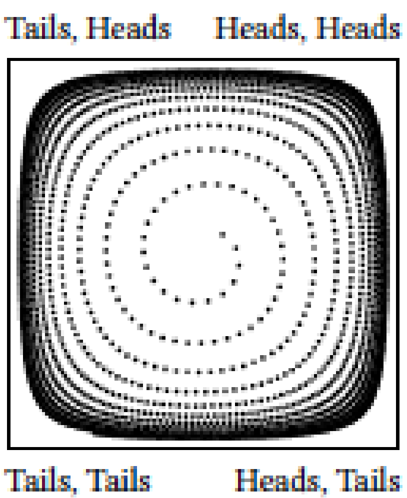
Multiplicative Weights Update Method

Some known properties:

- MWU are no-regret algorithms
- They converge in an “average sense”:
  - $(1/T)\ \Sigma_{t \leq T}\ (x^t, y^t)$ converges to the equilibrium as $T \to \infty$
- How about last-iterate convergence?
- $\lim_{t \to \infty} (x^t, y^t) = ?$

Multiplicative Weights Update Method

- $\lim_{t \to \infty} (x^t, y^t) = ?$
- [Bailey, Piliouras '18]: MWU (and many of its variants) do not converge in the last-iterate sense, and enter limit cycles even for 2x2 0-sum games



Spiraling away from the equilibrium in the Matching Pennies game

How can we "correct" MWU?

- Let's first ask: how do we "correct" the dynamics of gradient descent?
- Two well known tricks:
  - Optimistic gradient descent (OG) [Popov '80]
  - Extra gradient (EG) [Korpelevich '76]

Optimistic Gradient:

$$x^t = x^{t-1} + 2\eta\nabla f(x^{t-1}) - \eta\nabla f(x^{t-2})$$

(resp. for y^t)

Extra Gradient:

Intermediate step:

$$x^{t-1/2} = x^{t-1} - \eta\nabla f(x^{t-1})$$

Update step:

$$x^t = x^{t-1} - \eta\nabla f(x^{t-1/2}) \quad \text{(resp. for } y^t\text{)}$$

Can we define analogous versions for MWU?

Optimistic MWU

[Daskalakis, Panageas '19]: study of OMWU

- OMWU adds a negative momentum term to "correct" the MWU dynamics

Dynamics for OMWU:

$$x_i^t = x_i^{t-1} \cdot \frac{e^{2\eta \cdot f(e_i, y^{t-1}) - \eta \cdot f(e_i, y^{t-2})}}{\sum_{j=1}^n x_j^{t-1} e^{2\eta \cdot f(e_j, y^{t-1}) - \eta \cdot f(e_j, y^{t-2})}}, \quad y_j^t = y_j^{t-1} \cdot \frac{e^{-2\eta \cdot f(x^{t-1}, e_j) + \eta \cdot f(x^{t-2}, e_j)}}{\sum_{i=1}^n y_i^{t-1} e^{-2\eta \cdot f(x^{t-1}, e_i) + \eta \cdot f(x^{t-2}, e_i)}}$$

Theorem [from DP '19]: OMWU attains asymptotic last-iterate convergence, when the game has a unique equilibrium

[Wei et al. '21]: convergence rate analysis of OMWU

Conclusions

- Very active research agenda (both experimentally and theoretically)
- Open questions:
  - ➢Q1: Are there other dynamics with last-iterate convergence?
  - ➢Q2: Can we attain faster convergence rates? Especially when f is not bilinear (or not convex-concave but under other restrictions)
- Improved variants have potential to be deployed in practice
  - ➢ An example with Optimistic Gradient: [Daskalakis, Ilyas, Syrgkanis, Zeng '18], Training GANs with Optimism
- Beyond 0-sum?
  - Recent progress for rank-1 games [Patris, Panageas '24]