# Regret Matching$^+$:

## Instability, average- and last-iterate convergence in games

Julien Grand-Clément, Assistant Professor, ISOM Department, HEC Paris

Joint with Yang Cai[1], Gabriele Farina[2],Christian Kroer[3], Chung-Wei Lee[4], Haipeng Luo[4], Weiqiang Zheng[1]

[1] Yale, [2] MIT, [3] Columbia, [4] USC

# TL;DR

What is this talk about?

- Regret minimization: prevalent for solving games
- Regret Matching$^+$ (RM$^+$): regret minimizer used in all poker AI breakthroughs, widely outperform other methods in practice...
- ... despite "weak" theoretical guarantees:
  - RM$^+$:$O(1/\sqrt{T})$ convergence to Nash equilibrium
  - State-of-the-art: $O(1/T)$ convergence to NE

# TL;DR

What is this talk about?

- Regret minimization: prevalent for solving games
- Regret Matching$^+$ (RM$^+$): regret minimizer used in all poker AI breakthroughs, widely outperform other methods in practice...
- ... despite "weak" theoretical guarantees:
  - RM$^+$:$O(1/\sqrt{T})$ convergence to Nash equilibrium
  - State-of-the-art: $O(1/T)$ convergence to NE

What is missing in the literature?

1. Gap between empirical vs. theoretical performances of RM$^+$
2. Can RM$^+$-based algorithms achieve $O(1/T)$ average convergence?

# TL;DR

Our contributions:

1. We show a surprising "failure mode" of $\text{RM}^+$, due to its *instability*.
2. We provide two fixes: *restarting* and *smoothing*.
   $\Rightarrow$ New algorithms for game solving:
   $\cdot O(1/T)$ average convergence
   $\cdot O(1/\sqrt{T})$ best-iterate convergence, last-iterate convergence

# TL;DR

Our contributions:

1. We show a surprising "failure mode" of $\mathrm{RM}^+$, due to its *instability*.
2. We provide two fixes: *restarting* and *smoothing*.
   $\Rightarrow$ New algorithms for game solving:
   $\cdot O(1/T)$ average convergence
   $\cdot O(1/\sqrt{T})$ best-iterate convergence, last-iterate convergence

Why is this interesting?

1. Reconcile $\mathrm{RM}^+$-based methods with state-of-the-art th. guarantees
2. Several questions remain open: advantages of alternation, linear averaging, the case of extensive-form games, etc.

# TL;DR

Presentation based on:

- *Regret Matching$^+$: Instability and Fast Convergence in Games*, Farina, G.-C., Kroer, Lee and Luo, NeurIPS 2023.
- *Last-iterate convergence of regret matching-based algorithms in games*, Cai, Farina, G.-C., Kroer, Lee, Luo, Zheng, under review.

Outline for today:

1. Game solving via regret minimization
2. Regret Matching$^+$ (RM$^+$) and instability
3. Improved average convergence after stabilizing RM$^+$
4. Last-iterate convergence after stabilizing RM$^+$

Regret minimization: for $t = 1, ..., T$,

1. Choose a strategy $\boldsymbol{x}_t \in \Delta_n$ based on past observations
2. Observe the *loss vector* $\boldsymbol{\ell}_t \in \mathbb{R}^n$
3. Suffer an instantaneous loss $\langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \in \mathbb{R}$

Regret minimization: for $t = 1, ..., T$,

1. Choose a strategy $\boldsymbol{x}_t \in \Delta_n$ based on past observations
2. Observe the *loss vector* $\boldsymbol{\ell}_t \in \mathbb{R}^n$
3. Suffer an instantaneous loss $\langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \in \mathbb{R}$

The *regret* $\mathrm{Reg}^T$ at period $T$ is

$$\mathrm{Reg}^T := \max_{a \in \{1,\dots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$

Regret minimization: for $t = 1, \ldots, T$,

1. Choose a strategy $\boldsymbol{x}_t \in \Delta_n$ based on past observations
2. Observe the *loss vector* $\boldsymbol{\ell}_t \in \mathbb{R}^n$
3. Suffer an instantaneous loss $\langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \in \mathbb{R}$

The *regret* $\mathrm{Reg}^T$ at period $T$ is

$$\mathrm{Reg}^T := \max_{a \in \{1,\ldots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$

A *regret minimizer* constructs a sequence of decisions $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ in $\Delta_n$ such that for any sequence of losses $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots$, we have

$$\lim_{T \to +\infty} \frac{\mathrm{Reg}^T}{T} = 0.$$

Regret minimization: for $t = 1, ..., T$,

1. Choose a strategy $\boldsymbol{x}_t \in \Delta_n$ based on past observations
2. Observe the *loss vector* $\boldsymbol{\ell}_t \in \mathbb{R}^n$
3. Suffer an instantaneous loss $\langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \in \mathbb{R}$

The *regret* $\mathrm{Reg}^T$ at period $T$ is

$$\mathrm{Reg}^T := \max_{a \in \{1,\dots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$

A *regret minimizer* constructs a sequence of decisions $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots$ in $\Delta_n$ such that for any sequence of losses $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \dots$, we have

$$\lim_{T \to +\infty} \frac{\mathrm{Reg}^T}{T} = 0.$$

Why do we care?

Regret minimization: for $t = 1, \ldots, T$,

1. Choose a strategy $\boldsymbol{x}_t \in \Delta_n$ based on past observations
2. Observe the *loss vector* $\boldsymbol{\ell}_t \in \mathbb{R}^n$
3. Suffer an instantaneous loss $\langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \in \mathbb{R}$

The *regret* $\mathrm{Reg}^T$ at period $T$ is

$$\mathrm{Reg}^T := \max_{a \in \{1,\ldots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$

A *regret minimizer* constructs a sequence of decisions $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ in $\Delta_n$ such that for any sequence of losses $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots$, we have

$$\lim_{T \to +\infty} \frac{\mathrm{Reg}^T}{T} = 0.$$

Why do we care? Online resource allocation [BLM22], auctions [BG19], game solving: poker [BBJT15], Go [SHM$^+$16]...

Regret minimization can be used to solve matrix games:

$$\min_{\boldsymbol{x}\in\Delta_n} \max_{\boldsymbol{y}\in\Delta_m} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle.$$

Duality gap of a pair $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$:

$$\mathrm{DualityGap}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \max_{\boldsymbol{y}\in\Delta_m} \langle \hat{\boldsymbol{x}}, \boldsymbol{A}\boldsymbol{y} \rangle - \min_{\boldsymbol{x}\in\Delta_n} \langle \boldsymbol{x}, \boldsymbol{A}\hat{\boldsymbol{y}} \rangle.$$

$\mathrm{DualityGap}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \leq \epsilon \Rightarrow (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is $\epsilon$-Nash equilibrium

Regret minimization can be used to solve matrix games:

$$\min_{\boldsymbol{x}\in\Delta_n}\max_{\boldsymbol{y}\in\Delta_m}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle.$$

Duality gap of a pair $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$:

$$\text{DualityGap}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \max_{\boldsymbol{y}\in\Delta_m}\langle \hat{\boldsymbol{x}}, \boldsymbol{A}\boldsymbol{y}\rangle - \min_{\boldsymbol{x}\in\Delta_n}\langle \boldsymbol{x}, \boldsymbol{A}\hat{\boldsymbol{y}}\rangle.$$

$\text{DualityGap}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \le \epsilon \Rightarrow (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is $\epsilon$-Nash equilibrium

**Folk Theorem [FS99]**

Assume that each player of a matrix game runs a regret minimizer with loss $\ell_t$ equal to their own *expected cost*.

Then the average of the iterates is an approximate *Nash equilibrium* of the game, with a duality gap equal to

$$\frac{\mathrm{Reg}_1^T + \mathrm{Reg}_2^T}{T}.$$

Rock Paper Scissors:

$$\min_{\boldsymbol{x}\in\Delta_3}\max_{\boldsymbol{y}\in\Delta_3}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle, \boldsymbol{A} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

$x_0 = \mathbb{P}(\text{play rock}), x_1 = \mathbb{P}(\text{play paper}), x_3 = \mathbb{P}(\text{play scissors})$, etc.

Unique Nash Eq.: $\boldsymbol{x}^\star = \boldsymbol{y}^\star = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$.

Losses for x-player: $\boldsymbol{A}\boldsymbol{y}$, loss for $y$-player: $-\boldsymbol{A}^\top \boldsymbol{x}$.

Rock Paper Scissors:

$$\min_{\boldsymbol{x}\in\Delta_3}\max_{\boldsymbol{y}\in\Delta_3}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle, \boldsymbol{A} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

Run Regret Matching$^+$ (TBD) to generate $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ and $\boldsymbol{y}_1, ..., \boldsymbol{y}_T$.

Average iterates:

$$\bar{\boldsymbol{x}}_T = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t, \bar{\boldsymbol{y}}_t = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}_t$$

Rock Paper Scissors:

$$\min_{\boldsymbol{x}\in\Delta_3}\max_{\boldsymbol{y}\in\Delta_3}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle, \boldsymbol{A} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

Run Regret Matching$^+$ (TBD) to generate $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$.

Average iterates:

$$\bar{\boldsymbol{x}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t, \bar{\boldsymbol{y}}_t = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}_t$$

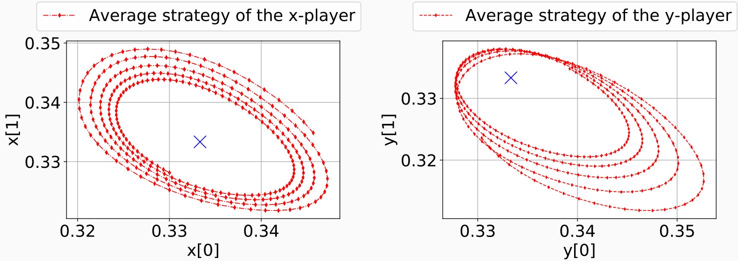



**Figure 1:** Running Regret Matching$^+$ for 500 iterations.

### Folk Theorem [FS99]

Assume that each player of a matrix game runs a regret minimizer with loss $\ell_t$ equal to their own *expected cost*.

Then the average of the iterates is an approximate *Nash equilibrium* of the game, with a duality gap equal to

$$\frac{\mathrm{Reg}_1^T + \mathrm{Reg}_2^T}{T}.$$

**Folk Theorem [FS99]**

Assume that each player of a matrix game runs a regret minimizer with loss $\ell_t$ equal to their own *expected cost*.

Then the average of the iterates is an approximate *Nash equilibrium* of the game, with a duality gap equal to

$$\frac{\mathrm{Reg}_1^T + \mathrm{Reg}_2^T}{T}.$$

1. If $\forall$ player $i, \mathrm{Reg}_i^T = O\left(\sqrt{T}\right)$ then convergence in $O\left(1/\sqrt{T}\right)$.

**Folk Theorem [FS99]**

Assume that each player of a matrix game runs a regret minimizer with loss $\ell_t$ equal to their own *expected cost*.

Then the average of the iterates is an approximate *Nash equilibrium* of the game, with a duality gap equal to

$$\frac{\mathrm{Reg}_1^T + \mathrm{Reg}_2^T}{T}.$$

1. If $\forall$ player $i, \mathrm{Reg}_i^T = O\left(\sqrt{T}\right)$ then convergence in $O\left(1/\sqrt{T}\right)$.
2. If $\mathrm{Reg}_1^T + \mathrm{Reg}_2^T = \tilde{O}\left(1\right)$ then convergence in $\tilde{O}\left(1/T\right)$.

   This is the <u>theoretical</u> state-of-the-art [RS13, SALS15, DFG21]...

**Folk Theorem [FS99]**

Assume that each player of a matrix game runs a regret minimizer with loss $\ell_t$ equal to their own *expected cost*.

Then the average of the iterates is an approximate *Nash equilibrium* of the game, with a duality gap equal to

$$\frac{\mathrm{Reg}_1^T + \mathrm{Reg}_2^T}{T}.$$

1. If $\forall$ player $i, \mathrm{Reg}_i^T = O\left(\sqrt{T}\right)$ then convergence in $O\left(1/\sqrt{T}\right)$.
2. If $\mathrm{Reg}_1^T + \mathrm{Reg}_2^T = \tilde{O}\left(1\right)$ then convergence in $\tilde{O}\left(1/T\right)$.

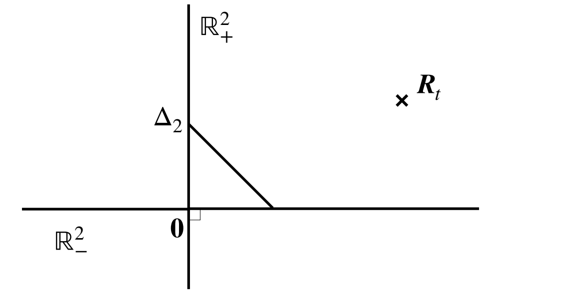
   This is the theoretical state-of-the-art [RS13, SALS15, DFG21]...

   ... but the empirical state-of-the-art (for poker AI) is a regret minimizer with "only" $O(1/\sqrt{T})$ convergence guarantees.

# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \mathbf{0} \in \mathbb{R}^n_+$, then

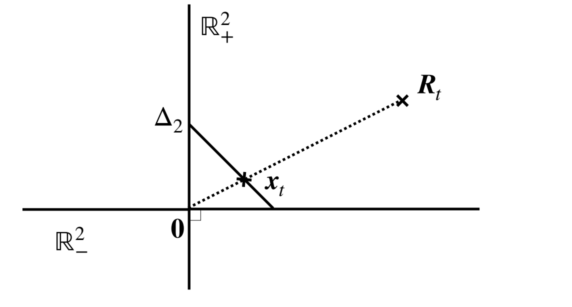
$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+$$



# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \mathbf{0} \in \mathbb{R}^n_+$, then

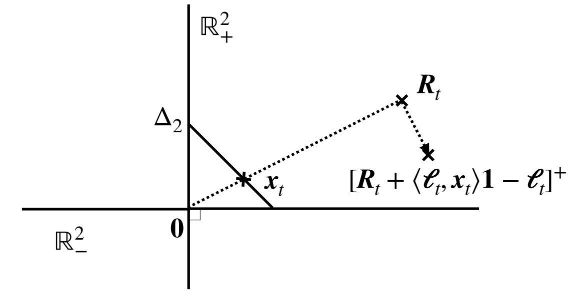
$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+$$



# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \boldsymbol{0} \in \mathbb{R}^n_+$, then

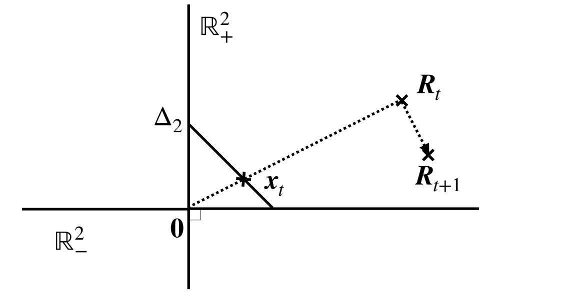
$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1 \\ \boldsymbol{R}_{t+1} &= [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \boldsymbol{1} - \boldsymbol{\ell}_t]^+ \end{aligned}$$



# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \mathbf{0} \in \mathbb{R}^n_+$, then

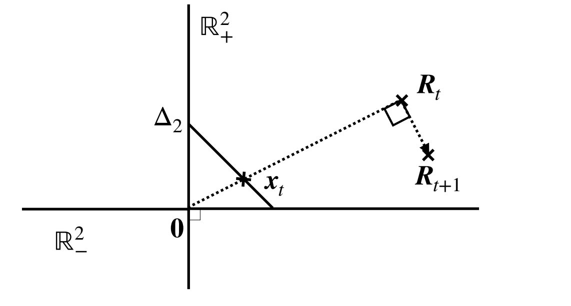
$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+$$



# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \boldsymbol{0} \in \mathbb{R}^n_+$, then

$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \boldsymbol{1} - \boldsymbol{\ell}_t]^+$$



# Why is this called Regret Matching$^+$?

The update for $\boldsymbol{R}_t$ is

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

# Why is this called Regret Matching$^+$?

The update for $\boldsymbol{R}_t$ is

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

Suppose we forget about the operator $[\cdot]^+$, then

$$\boldsymbol{R}_{T+1} = \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \sum_{t=1}^{T} \boldsymbol{\ell}_t.$$

# Why is this called Regret Matching$^+$?

The update for $\boldsymbol{R}_t$ is

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

Suppose we forget about the operator $[\cdot]^+$, then

$$\boldsymbol{R}_{T+1} = \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \sum_{t=1}^{T} \boldsymbol{\ell}_t.$$

Recall the definition of the regret:

$$\begin{aligned} \mathrm{Reg}^T &:= \max_{a \in \{1,\dots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta} \\ &= \max_{a \in \{1,\dots,n\}} R_{T+1,a} \end{aligned}$$

# Why is this called Regret Matching$^+$?

The update for $\boldsymbol{R}_t$ is

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

Suppose we forget about the operator $[\cdot]^+$, then

$$\boldsymbol{R}_{T+1} = \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \sum_{t=1}^{T} \boldsymbol{\ell}_t.$$

Recall the definition of the regret:

$$\begin{aligned} \mathrm{Reg}^T &:= \max_{a \in \{1,\ldots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta} \\ &= \max_{a \in \{1,\ldots,n\}} R_{T+1,a} \end{aligned}$$

$\Rightarrow$ $\boldsymbol{R}_t$ is called the *lifted regret* and $\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty$.

# Why is this called Regret Matching$^+$?

The update for $\boldsymbol{R}_t$ is

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

Suppose we forget about the operator $[\cdot]^+$, then

$$\boldsymbol{R}_{T+1} = \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \sum_{t=1}^{T} \boldsymbol{\ell}_t.$$

Recall the definition of the regret:

$$\begin{aligned} \mathrm{Reg}^T &:= \max_{a \in \{1,\ldots,n\}} \sum_{t=1}^{T} \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \ell_{ta} \\ &= \max_{a \in \{1,\ldots,n\}} R_{T+1,a} \end{aligned}$$

$\Rightarrow$ $\boldsymbol{R}_t$ is called the *lifted regret* and $\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty$.
$\Rightarrow$ $\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$: we play actions with large regrets

# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \mathbf{0} \in \mathbb{R}^n_+$, then

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1, \\ \boldsymbol{R}_{t+1} &= [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+. \end{aligned}$$

Why do we like this algorithm?

1. RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.

# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \boldsymbol{0} \in \mathbb{R}^n_+$, then

$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1,$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

Why do we like this algorithm?

1. RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Geometric intuition: $\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty$.

# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \boldsymbol{0} \in \mathbb{R}_+^n$, then

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1, \\ \boldsymbol{R}_{t+1} &= [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \boldsymbol{1} - \boldsymbol{\ell}_t]^+. \end{aligned}$$

Why do we like this algorithm?

1. RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Geometric intuition: $\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty$.
3. Parameter-free: no step size to learn/choose

# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \boldsymbol{0} \in \mathbb{R}^n_+$, then

$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1,$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \boldsymbol{1} - \boldsymbol{\ell}_t]^+.$$

Why do we like this algorithm?

1. RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Geometric intuition: $\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty$.
3. Parameter-free: no step size to learn/choose
4. Strong empirical performances, $\approx$ 10x faster than $O(1/T)$ algos [BBJT15, MSB$^+$17, BS18, BS19, FKS21]...

# Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $\boldsymbol{R}_1 = \boldsymbol{0} \in \mathbb{R}_+^n$, then

$$\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1,$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \boldsymbol{1} - \boldsymbol{\ell}_t]^+.$$

Why do we like this algorithm?

1. RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Geometric intuition: $\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty$.
3. Parameter-free: no step size to learn/choose
4. Strong empirical performances, $\approx$ 10x faster than $O(1/T)$ algos [BBJT15, MSB$^+$17, BS18, BS19, FKS21]...
5. ... and RM$^+$ is still not very well understood!

# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\boldsymbol{\ell}_t$ when computing $\boldsymbol{x}_t$.

$$\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \boldsymbol{\ell}_t$$

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\boldsymbol{\ell}_t$ when computing $\boldsymbol{x}_t$.

$$\begin{aligned} \hat{\boldsymbol{R}}_t &= [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_{t-1}, \boldsymbol{x}_{t-1} \rangle \mathbf{1} - \boldsymbol{\ell}_{t-1}]^+ \\ \boldsymbol{x}_t &= \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1, \\ \boldsymbol{R}_{t+1} &= [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+. \end{aligned}$$

# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\boldsymbol{\ell}_t$ when computing $\boldsymbol{x}_t$.

$$\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \boldsymbol{\ell}_t$$

$$\begin{aligned} \boldsymbol{x}_t &= \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1, \\ \boldsymbol{R}_{t+1} &= [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+. \end{aligned}$$

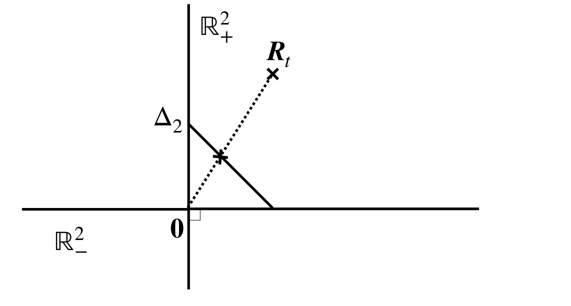
# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

$$\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \ell_t$$

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

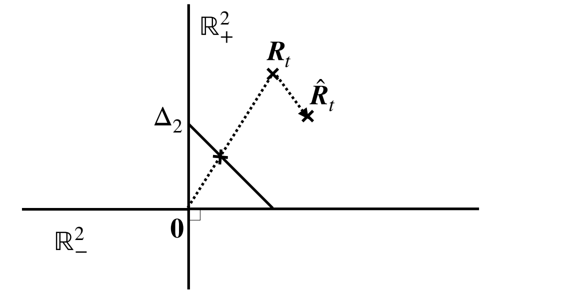
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \ell_t, \boldsymbol{x}_t \rangle \mathbf{1} - \ell_t]^+.$$



# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

$$\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \ell_t$$
$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$



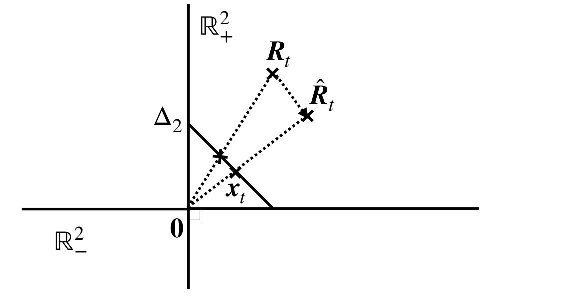
# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

*Build $\hat{\boldsymbol{R}}_t$ by predicting $\ell_t$*

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

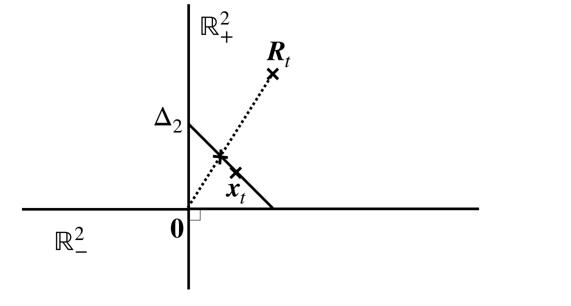
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$



# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

$$\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \ell_t$$
$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$



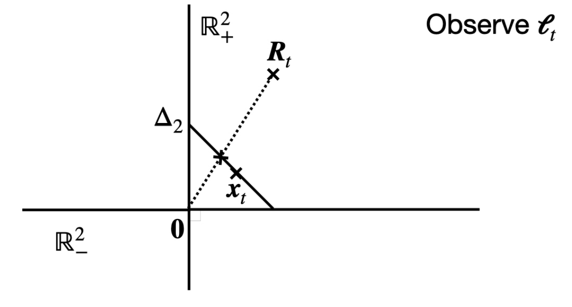
# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

*Build $\hat{\boldsymbol{R}}_t$ by predicting $\ell_t$*

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$



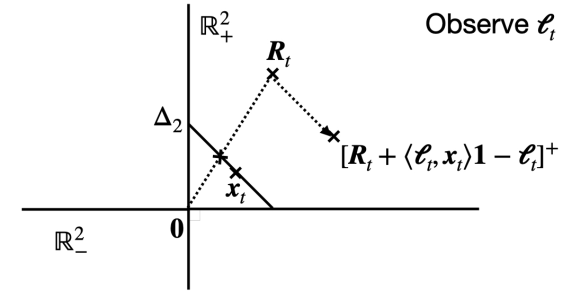
# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

$$\color{blue}{\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \ell_t}$$

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

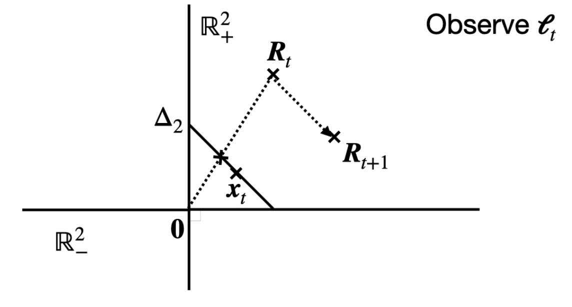
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \ell_t, \boldsymbol{x}_t \rangle \mathbf{1} - \ell_t]^+.$$



# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

$$\textcolor{blue}{\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \ell_t}$$
$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \ell_t, \boldsymbol{x}_t \rangle \mathbf{1} - \ell_t]^+.$$



# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\ell_t$ when computing $\boldsymbol{x}_t$.

*Build $\hat{\boldsymbol{R}}_t$ by predicting $\ell_t$ as $\ell_{t-1}$*

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

1. Predictive RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.

# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\boldsymbol{\ell}_t$ when computing $\boldsymbol{x}_t$.

*Build $\hat{\boldsymbol{R}}_t$ by predicting $\boldsymbol{\ell}_t$ as $\boldsymbol{\ell}_{t-1}$*

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

1. Predictive RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Parameter-free: no step size to learn/choose

# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\boldsymbol{\ell}_t$ when computing $\boldsymbol{x}_t$.

$$\textit{Build } \hat{\boldsymbol{R}}_t \textit{ by predicting } \boldsymbol{\ell}_t \textit{ as } \boldsymbol{\ell}_{t-1}$$

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

1. Predictive RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Parameter-free: no step size to learn/choose
3. Strong empirical performances, vastly outperforms $O(1/T)$ algos [BBJT15, MSB$^+$17, BS18, BS19, FKS21].

# Predictive Regret Matching$^+$ [FKS21]

Main idea: use a prediction of $\boldsymbol{\ell}_t$ when computing $\boldsymbol{x}_t$.

*Build $\hat{\boldsymbol{R}}_t$ by predicting $\boldsymbol{\ell}_t$ as $\boldsymbol{\ell}_{t-1}$*

$$\boldsymbol{x}_t = \hat{\boldsymbol{R}}_t / \|\hat{\boldsymbol{R}}_t\|_1,$$

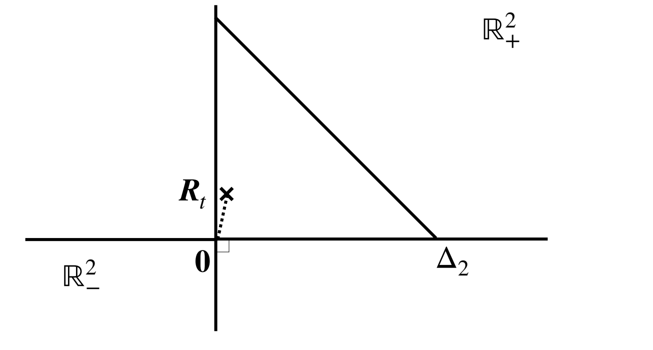
$$\boldsymbol{R}_{t+1} = [\boldsymbol{R}_t + \langle \boldsymbol{\ell}_t, \boldsymbol{x}_t \rangle \mathbf{1} - \boldsymbol{\ell}_t]^+.$$

1. Predictive RM$^+$ is a regret minimizer: $\mathrm{Reg}^T = O\left(\sqrt{T}\right)$.
2. Parameter-free: no step size to learn/choose
3. Strong empirical performances, vastly outperforms $O(1/T)$ algos [BBJT15, MSB$^+$17, BS18, BS19, FKS21].
4. But not known to ensure $O(1/T)$ convergence, despite optimism!

# Instability in (predictive) RM$^+$

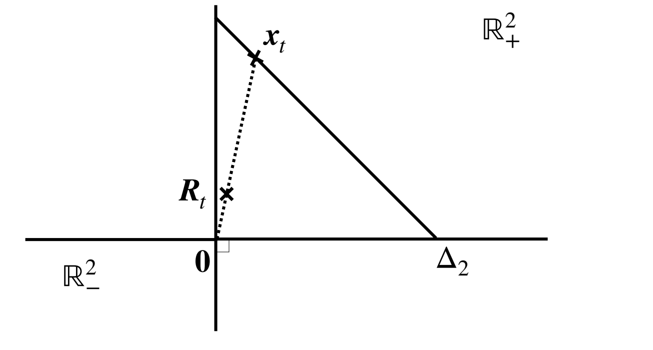
Recall that $\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$.

Instability: $\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_2$ may be large... despite small $\|\boldsymbol{R}_t - \boldsymbol{R}_{t+1}\|_2$.



Recall that $\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$.

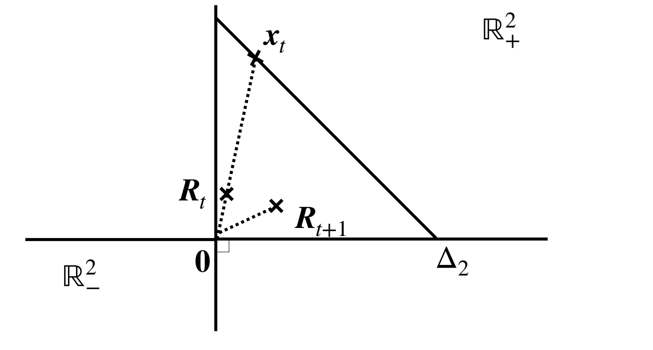
Instability: $\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_2$ may be large... despite small $\|\boldsymbol{R}_t - \boldsymbol{R}_{t+1}\|_2$.



# Instability in (predictive) RM$^+$

Recall that $\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$.

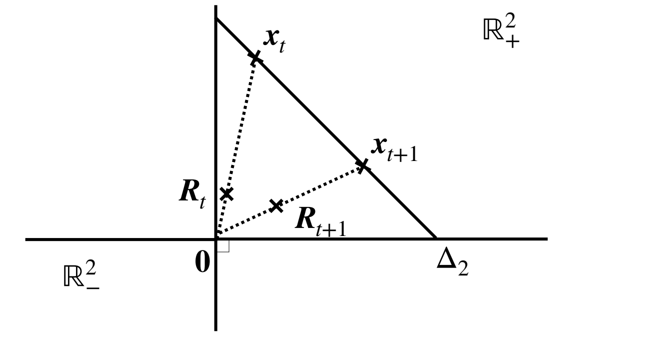
Instability: $\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_2$ may be large... despite small $\|\boldsymbol{R}_t - \boldsymbol{R}_{t+1}\|_2$.



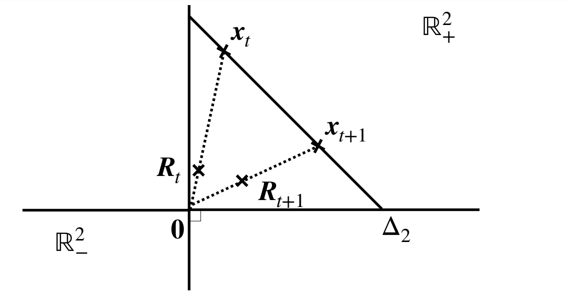
# Instability in (predictive) RM$^+$

Recall that $\boldsymbol{x}_t = \boldsymbol{R}_t / \|\boldsymbol{R}_t\|_1$.

Instability: $\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_2$ may be large... despite small $\|\boldsymbol{R}_t - \boldsymbol{R}_{t+1}\|_2$.

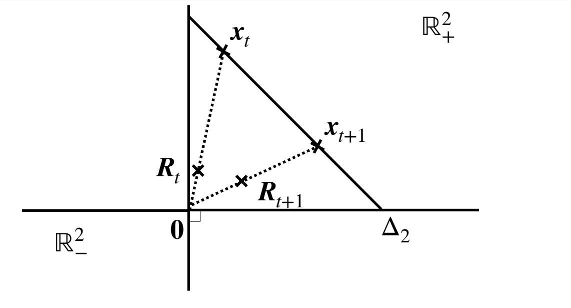


# Instability in (predictive) RM$^+$




Instability happens *because* $\|\boldsymbol{R}_t\|_1$ is small.

# Instability in (predictive) RM$^+$



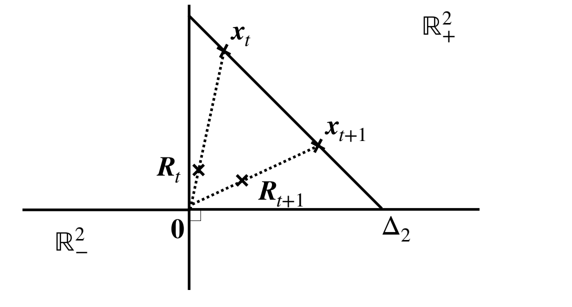

Instability happens *because* $\|\boldsymbol{R}_t\|_1$ is small.

**Proposition**

Let $\boldsymbol{R}_1, \boldsymbol{R}_2 \in \mathbb{R}^n_+$ and $\boldsymbol{x}_1 = \boldsymbol{R}_1/\|\boldsymbol{R}_1\|_1, \boldsymbol{x}_2 = \boldsymbol{R}_2/\|\boldsymbol{R}_2\|_1$. Then

$$\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \leq \frac{\sqrt{n}}{\max\{\|\boldsymbol{R}_1\|_1, \|\boldsymbol{R}_2\|_1\}} \cdot \|\boldsymbol{R}_1 - \boldsymbol{R}_2\|_2 \tag{1}$$

# Instability in (predictive) RM$^+$



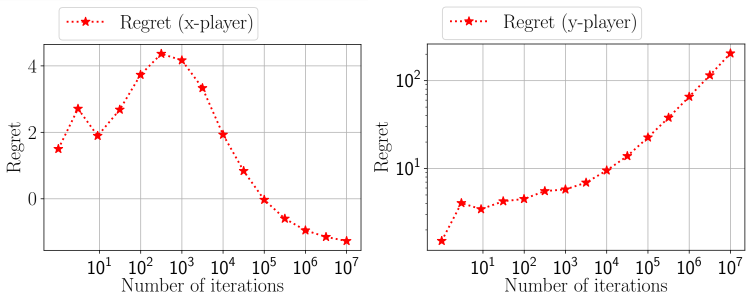

- Instability makes it hard to minimize regret for the other players...
- But recall that small $\|\boldsymbol{R}_T\|_\infty$ is good news for the player:

$$\mathrm{Reg}^T \leq \|\boldsymbol{R}_{T+1}\|_\infty.$$

# Example on a pathological example

Solving a small matrix game: $\min_{\boldsymbol{x} \in \Delta_3} \max_{\boldsymbol{y} \in \Delta_3} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$.

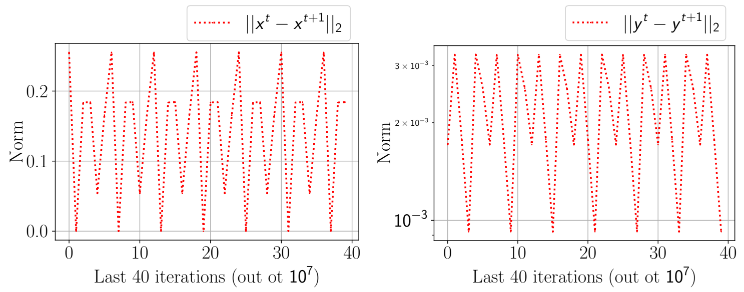
Running (vanilla) Predictive RM$^+$:



# Example on a pathological example

Solving a small matrix game: $\min_{\boldsymbol{x}\in\Delta_3} \max_{\boldsymbol{y}\in\Delta_3} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle$.

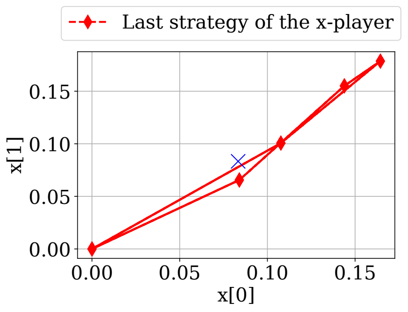
Running (vanilla) Predictive RM$^+$:



# Example on a pathological example

Solving a small matrix game: $\min_{\boldsymbol{x} \in \Delta_3} \max_{\boldsymbol{y} \in \Delta_3} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$.

Running (vanilla) Predictive RM$^+$:



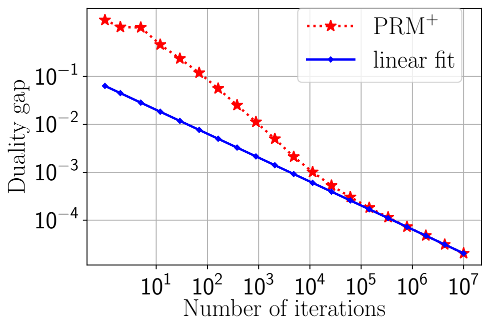

After $10^7$ iterations, $\boldsymbol{x}_t$ cycles between 5 strategies.

Recall that the loss for the y-player is $-\boldsymbol{A}^\top \boldsymbol{x}_t$!

# Example on a pathological example

Solving a small matrix game: $\min_{\boldsymbol{x}\in\Delta_3}\max_{\boldsymbol{y}\in\Delta_3}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle$.

Running (vanilla) Predictive RM$^+$:




Slope of the linear fit: $-0.496 \Rightarrow$ duality gap decreases as $O(1/\sqrt{T})$.

# Example on a pathological example

Diagnostic:

1. Instability of one player harms the convergence to an equilibrium.
2. Instability happens because $\|\boldsymbol{R}_t\|_1$ is small.

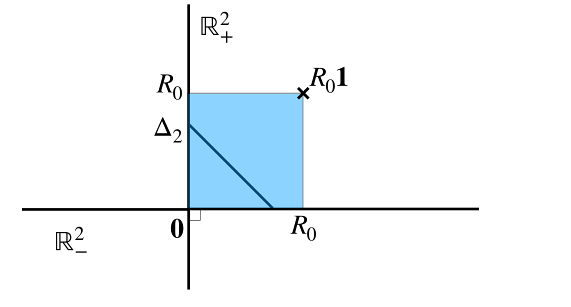
Question:

How to ensure that $\boldsymbol{R}_t$ is not too close to the origin $\mathbf{0}$?

# Toward stable Predictive RM$^+$: first idea

Restarting: run Predictive RM$^+$, and at the end of every iteration:

$$\text{If } \boldsymbol{R}_{t+1} \leq R_0\mathbf{1} \text{ then } \boldsymbol{R}_{t+1} = R_0\mathbf{1}.$$



# Toward stable Predictive RM$^+$: first idea

Restarting: run Predictive RM$^+$, and at the end of every iteration:

$$\text{If } \boldsymbol{R}_{t+1} \leq R_0\mathbf{1} \text{ then } \boldsymbol{R}_{t+1} = R_0\mathbf{1}.$$

This can be done in linear time.

**Theorem**

Assume that each player runs Predictive RM$^+$ with restarting with $R_0 = XXX$.

Then $\max \left\{ \mathrm{Reg}_1^T, \mathrm{Reg}_2^T \right\} = O(T^{1/4})$.

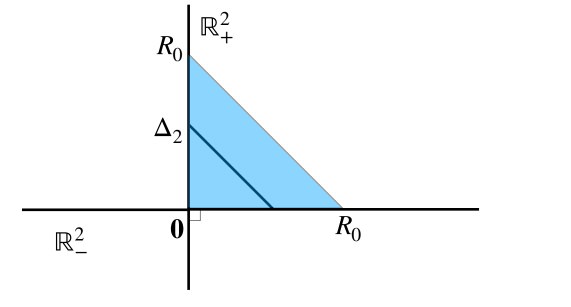
$\Rightarrow$ Convergence to a Nash Equilibrium at a rate of $O\left(1/T^{3/4}\right)$.

# Toward stable RM$^+$: second idea

Smoothing: run Predictive RM$^+$, and at the end of every iteration:

If $\langle \boldsymbol{R}_{t+1}, \mathbf{1} \rangle \leq R_0$ then replace $\boldsymbol{R}_{t+1}$ by its projection on $R_0 \Delta_n$.

This ensures $\boldsymbol{R}_t \in \{\boldsymbol{R} \in \mathbb{R}^n \mid \boldsymbol{R} \geq \mathbf{0}, \langle \boldsymbol{R}, \mathbf{1} \rangle \geq R_0\}$.

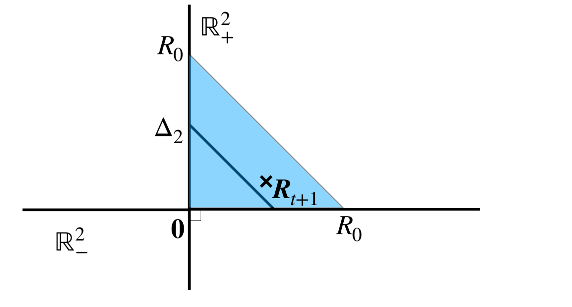


# Toward stable RM$^+$: second idea

Smoothing: run Predictive RM$^+$, and at the end of every iteration:

If $\langle \boldsymbol{R}_{t+1}, \mathbf{1} \rangle \leq R_0$ then replace $\boldsymbol{R}_{t+1}$ by its projection on $R_0 \Delta_n$.

This ensures $\boldsymbol{R}_t \in \{\boldsymbol{R} \in \mathbb{R}^n \mid \boldsymbol{R} \geq \mathbf{0}, \langle \boldsymbol{R}, \mathbf{1} \rangle \geq R_0\}$.

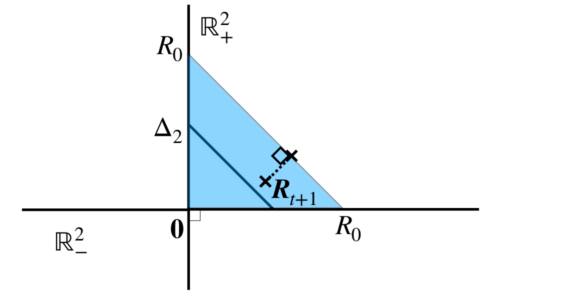


# Toward stable RM$^+$: second idea

Smoothing: run Predictive RM$^+$, and at the end of every iteration:

If $\langle \boldsymbol{R}_{t+1}, \mathbf{1} \rangle \leq R_0$ then replace $\boldsymbol{R}_{t+1}$ by its projection on $R_0\Delta_n$.

This ensures $\boldsymbol{R}_t \in \{\boldsymbol{R} \in \mathbb{R}^n \mid \boldsymbol{R} \geq \mathbf{0}, \langle \boldsymbol{R}, \mathbf{1} \rangle \geq R_0\}$.

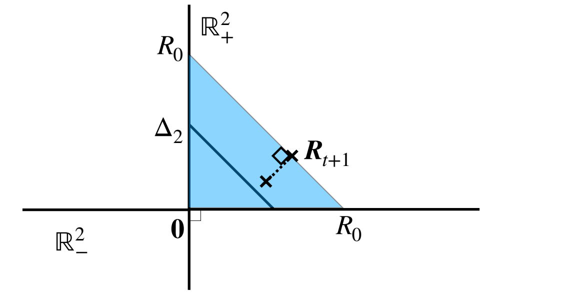


# Toward stable $\mathrm{RM}^+$: second idea

Smoothing: run Predictive $\mathrm{RM}^+$, and at the end of every iteration:

If $\langle \boldsymbol{R}_{t+1}, \mathbf{1} \rangle \leq R_0$ then replace $\boldsymbol{R}_{t+1}$ by its projection on $R_0\Delta_n$.

This ensures $\boldsymbol{R}_t \in \{\boldsymbol{R} \in \mathbb{R}^n \mid \boldsymbol{R} \geq \mathbf{0}, \langle \boldsymbol{R}, \mathbf{1} \rangle \geq R_0\}$.



# Toward stable RM$^+$: second idea

Smoothing: run Predictive RM$^+$, and at the end of every iteration:

If $\langle \boldsymbol{R}_{t+1}, \mathbf{1} \rangle \leq R_0$ then replace $\boldsymbol{R}_{t+1}$ by its projection on $R_0 \Delta_n$.

This ensures $\boldsymbol{R}_t \in \{\boldsymbol{R} \in \mathbb{R}^n \mid \boldsymbol{R} \geq \mathbf{0}, \langle \boldsymbol{R}, \mathbf{1} \rangle \geq R_0\}$.

This can be done in $O(n \log(n))$.

$\boldsymbol{R} \mapsto \boldsymbol{R}/\|\boldsymbol{R}\|_1$ is smooth on $\{\boldsymbol{R} \in \mathbb{R}^n \mid \boldsymbol{R} \geq \mathbf{0}, \langle \boldsymbol{R}, \mathbf{1} \rangle \geq R_0\}$:

$$\left\| \frac{\boldsymbol{R}_1}{\|\boldsymbol{R}_1\|_1} - \frac{\boldsymbol{R}_2}{\|\boldsymbol{R}_2\|_1} \right\|_2 \leq \frac{\sqrt{n}}{R_0} \cdot \|\boldsymbol{R}_1 - \boldsymbol{R}_2\|_2 \tag{2}$$

# Toward stable RM$^+$: second idea

Smoothing: run Predictive RM$^+$, and at the end of every iteration:

If $\langle \boldsymbol{R}_{t+1}, \mathbf{1} \rangle \leq R_0$ then replace $\boldsymbol{R}_{t+1}$ by its projection on $R_0 \Delta_n$.

**Theorem**

Assume that each player runs Predictive RM$^+$ with Smoothing with $R_0 = XXX$. Then:

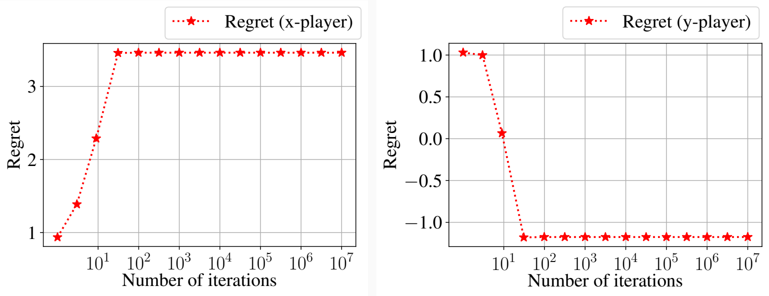
- $\max \left\{ \mathrm{Reg}_1^T, \mathrm{Reg}_2^T \right\} = O\left(T^{1/4}\right)$.
- $\mathrm{Reg}_1^T + \mathrm{Reg}_2^T = O(1)$.

$\Rightarrow$ Convergence to a Nash Equilibrium at a rate of $O\left(1/T\right)$.

# Example on a pathological example (continued)

Solving a small matrix game: $\min_{\boldsymbol{x} \in \Delta_3} \max_{\boldsymbol{y} \in \Delta_3} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle$.

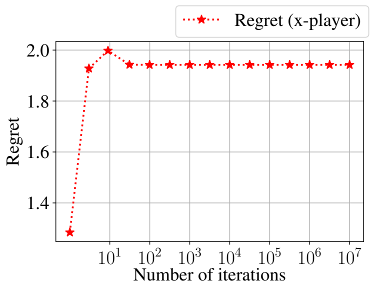
Running Predictive RM$^+$ with restarting:

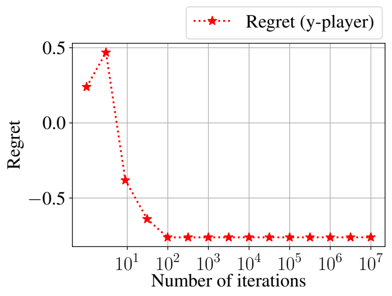


# Example on a pathological example (continued)

Solving a small matrix game: $\min_{\boldsymbol{x}\in\Delta_3}\max_{\boldsymbol{y}\in\Delta_3}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle$.

Running Predictive RM$^+$ with Smoothing:

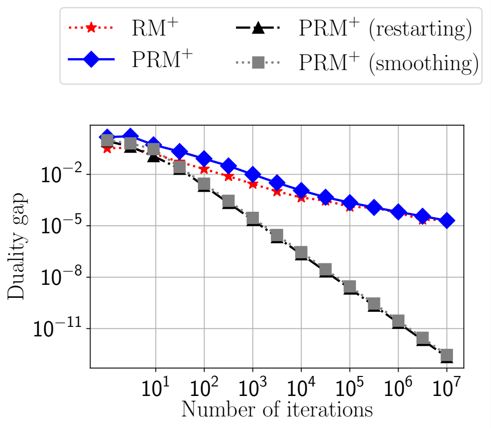


Solving a small matrix game:

$$\min_{\boldsymbol{x}\in\Delta_3} \max_{\boldsymbol{y}\in\Delta_3} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} \rangle.$$

Comparing the average convergence to a Nash Equilibrium:



# Last-iterate convergence

All the guarantees presented so far are for the average iterates:

$$\bar{\boldsymbol{x}}_T = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t, \bar{\boldsymbol{y}}_t = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}_t$$

How about convergence in $\boldsymbol{x}_T, \boldsymbol{y}_T$, i.e., last-iterate convergence?

# Last-iterate convergence

All the guarantees presented so far are for the average iterates:

$$\bar{\boldsymbol{x}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t, \bar{\boldsymbol{y}}_t = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}_t$$

How about convergence in $\boldsymbol{x}_T, \boldsymbol{y}_T$, i.e., last-iterate convergence?

Why do we care?

- Quite simpler than average iterates
- Averaging may be cumbersome/expensive computationally
- No last-iterate convergence $\Rightarrow$ cycling/diverging behaviors

# Last-iterate convergence

Convergence on average vs. last-iterate convergence:

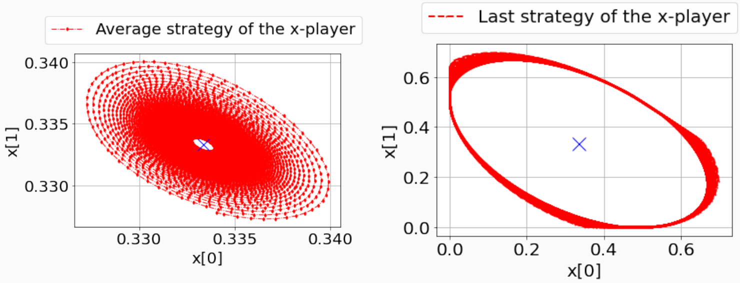



**Figure 4:** Running Regret Matching$^{+}$ for $10^5$ iterations for *Rock-Paper-Scissors*.

$\Rightarrow$ RM$^+$ and Predictive RM$^+$ may diverge on a simple $3 \times 3$ matrix game.

$\Rightarrow$ Poor performance of the last iterates of RM$^+$/ PRM$^+$:

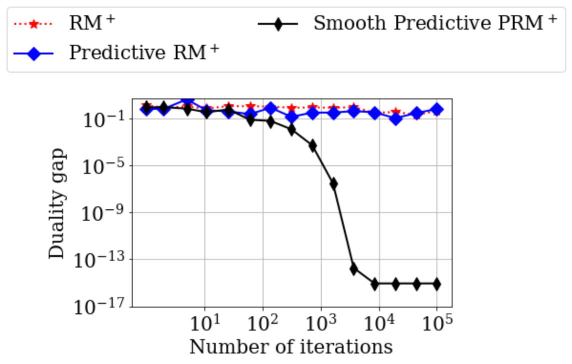



**Figure 5:** Last iterate performance of RM$^+$, PRM$^+$ and Smooth PRM$^+$.

# Our contributions 1/3

We could only prove convergence of $\mathrm{RM}^+$ under very strong assumptions.

**Theorem**

Assume that the matrix game has a *strict Nash Eq.* $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$:

- $\boldsymbol{x}^\star$ is the unique best-response to $\boldsymbol{y}^\star$
- $\boldsymbol{y}^\star$ is the unique best-response to $\boldsymbol{x}^\star$

Then $\mathrm{RM}^+$ converges: the sequence $(\boldsymbol{x}_t, \boldsymbol{y}_t)_{t\in\mathbb{N}}$ has a limit.

Note: strict N.E. implies N.E. is unique and $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ are deterministic.

# Our contributions 2/3

Let $\mathcal{Z}^\star \subset \Delta_n \times \Delta_m$ be the set of Nash equilibria.

**Theorem**

For Smooth Predictive RM$^+$, we show

1. Last-iterate convergence: the sequence $(\boldsymbol{x}_t, \boldsymbol{y}_t)_{t \in \mathbb{N}}$ has a limit.
2. Best-iterate convergence:
   For some $\alpha > 0$ and starting at $(\boldsymbol{x}_0, \boldsymbol{y}_0)$,

$$\min_{t \in \{1,\dots,T\}} \mathrm{DualityGap}\,(\boldsymbol{x}_t, \boldsymbol{y}_t) = \frac{\alpha \cdot \mathrm{dist}\,((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star)}{\sqrt{T}}$$

# Our contributions 2/3

Let $\mathcal{Z}^\star \subset \Delta_n \times \Delta_m$ be the set of Nash equilibria.

**Theorem**

For Smooth Predictive RM$^+$, we show

1. Last-iterate convergence: the sequence $(\boldsymbol{x}_t, \boldsymbol{y}_t)_{t \in \mathbb{N}}$ has a limit.
2. Best-iterate convergence:
   For some $\alpha > 0$ and starting at $(\boldsymbol{x}_0, \boldsymbol{y}_0)$,

$$\min_{t \in \{1,\ldots,T\}} \text{DualityGap}\,(\boldsymbol{x}_t, \boldsymbol{y}_t) = \frac{\alpha \cdot \text{dist}\,((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star)}{\sqrt{T}}$$

**Metric subregularity** [WLZL20] $\exists\ c > 0$ such that, for any $t \in \mathbb{N}$,

$$c \cdot \text{dist}\,((\boldsymbol{x}_t, \boldsymbol{y}_t), \mathcal{Z}^\star) \leq \text{DualityGap}(\boldsymbol{x}_t, \boldsymbol{y}_t).$$

# From best-iterate to linear last-iterate convergence

There exists a time $\tilde{t} \in \{1, ..., T\}$ such that

$$\mathrm{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{\alpha}{c\sqrt{T}} \cdot \mathrm{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

# From best-iterate to linear last-iterate convergence

There exists a time $\tilde{t} \in \{1, ..., T\}$ such that

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{\alpha}{c\sqrt{T}} \cdot \operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$T$ such that $\frac{\alpha}{c\sqrt{T}} = \frac{1}{2}$:

$\Rightarrow$ in a *constant* number of steps, we halve the distance to $\mathcal{Z}^\star$:

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{1}{2}\operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

# From best-iterate to linear last-iterate convergence

There exists a time $\tilde{t} \in \{1, ..., T\}$ such that

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{\alpha}{c\sqrt{T}} \cdot \operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$T$ such that $\frac{\alpha}{c\sqrt{T}} = \frac{1}{2}$:

$\Rightarrow$ in a *constant* number of steps, we halve the distance to $\mathcal{Z}^\star$:

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{1}{2}\operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$\Rightarrow$ Why not reinitializing the algorithm at time $\tilde{t}$: $(\boldsymbol{x}_0, \boldsymbol{y}_0) \leftarrow (\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}})$?

# From best-iterate to linear last-iterate convergence

There exists a time $\tilde{t} \in \{1, ..., T\}$ such that

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{\alpha}{c\sqrt{T}} \cdot \operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$T$ such that $\frac{\alpha}{c\sqrt{T}} = \frac{1}{2}$:

$\Rightarrow$ in a *constant* number of steps, we halve the distance to $\mathcal{Z}^\star$:

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{1}{2}\operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$\Rightarrow$ Why not reinitializing the algorithm at time $\tilde{t}$: $(\boldsymbol{x}_0, \boldsymbol{y}_0) \leftarrow (\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}})$?

Problem: of course we can't identify the time $\tilde{t}$...

# From best-iterate to linear last-iterate convergence

There exists a time $\tilde{t} \in \{1, ..., T\}$ such that

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{\alpha}{c\sqrt{T}} \cdot \operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$T$ such that $\frac{\alpha}{c\sqrt{T}} = \frac{1}{2}$:

$\Rightarrow$ in a *constant* number of steps, we halve the distance to $\mathcal{Z}^\star$:

$$\operatorname{dist}\left((\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}}), \mathcal{Z}^\star\right) \leq \frac{1}{2}\operatorname{dist}\left((\boldsymbol{x}_0, \boldsymbol{y}_0), \mathcal{Z}^\star\right).$$

$\Rightarrow$ Why not reinitializing the algorithm at time $\tilde{t}$: $(\boldsymbol{x}_0, \boldsymbol{y}_0) \leftarrow (\boldsymbol{x}_{\tilde{t}}, \boldsymbol{y}_{\tilde{t}})$?

Problem: of course we can't identify the time $\tilde{t}$...

Solution: bound the distance to $\mathcal{Z}^\star$ by distances between $\hat{\boldsymbol{R}}^t, \boldsymbol{R}^{t+1}, \boldsymbol{R}^t$.

# Our contributions 3/3

**Theorem**

Consider running Smooth Predictive RM$^+$, with the following trick:

At iteration $t$,

"Reinitialize the algorithm if the current duality gap has been halved since last reinitialization"

Then we have linear last-iterate convergence:

$$\text{DualityGap}\left(\boldsymbol{x}_t, \boldsymbol{y}_t\right) = O\left(\beta^t\right) \text{ for some } \beta \in (0,1)$$

# Conclusion

- Better understanding of Regret Matching$^+$ and predictive variants
- New algorithms with strong theoretical guarantees

# Conclusion

- Better understanding of Regret Matching$^+$ and predictive variants
- New algorithms with strong theoretical guarantees
- Limitations:
    1. We loose the step-size free property (choice of $R_0$)
    2. Convergence rates don't apply for extensive-form games (CFR)/multiplayer normal-form games
    3. Other unexplained aspects of RM$^+$: alternation, linear averaging, etc.
- More in the papers + code available online

# Conclusion

- Better understanding of Regret Matching$^+$ and predictive variants
- New algorithms with strong theoretical guarantees
- Limitations:
  1. We loose the step-size free property (choice of $R_0$)
  2. Convergence rates don't apply for extensive-form games (CFR)/multiplayer normal-form games
  3. Other unexplained aspects of RM$^+$: alternation, linear averaging, etc.
- More in the papers + code available online

Thank you!

- Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin.
  **Heads-up limit hold'em poker is solved.**
  *Science*, 347(6218):145–149, 2015.
- Santiago R Balseiro and Yonatan Gur.
  **Learning in repeated auctions with budgets: Regret minimization and equilibrium.**
  *Management Science*, 65(9):3952–3968, 2019.
- Santiago R Balseiro, Haihao Lu, and Vahab Mirrokni.
  **The best of many worlds: Dual mirror descent for online allocation problems.**
  *Operations Research*, 2022.

Noam Brown and Tuomas Sandholm.
**Superhuman AI for heads-up no-limit poker: Libratus beats top professionals.**
*Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm.
**Superhuman AI for multiplayer poker.**
*Science*, 365(6456):885–890, 2019.

Yang Cai, Argyris Oikonomou, and Weiqiang Zheng.
**Finite-time last-iterate convergence for learning in multi-player games.**
*Advances in Neural Information Processing Systems*, 35:33904–33919, 2022.

# References iii


Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich.
**Near-optimal no-regret learning in general games.**
*Advances in Neural Information Processing Systems*, 34:27604–27616, 2021.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm.
**Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent.**
In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2021.

Yoav Freund and Robert E Schapire.
**Adaptive game playing using multiplicative weights.**
*Games and Economic Behavior*, 29(1-2):79–103, 1999.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling.
**Deepstack: Expert-level artificial intelligence in heads-up no-limit poker.**
*Science*, 356(6337):508–513, 2017.

Alexander Rakhlin and Karthik Sridharan.
**Online learning with predictable sequences.**
In *Conference on Learning Theory*, pages 993–1019. PMLR, 2013.

Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire.
**Fast convergence of regularized learning in games.**
28, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al.
**Mastering the game of go with deep neural networks and tree search.**
*nature*, 529(7587):484–489, 2016.

Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling.
**Solving heads-up limit Texas hold'em.**
In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo.
**Linear last-iterate convergence in constrained saddle-point optimization.**
In *International Conference on Learning Representations*, 2020.

# Our contributions 3/3

**Theorem**

Consider running Smooth Predictive RM$^+$, with the following trick:

At iteration $t$,

$$\text{if } \|\hat{\boldsymbol{R}}^{t+1} - \boldsymbol{R}^t\|_2 + \|\hat{\boldsymbol{R}}^t - \boldsymbol{R}^t\|_2 \leq 2^{-k} \text{ then } \boldsymbol{R}^{t+1} \leftarrow \boldsymbol{x}_{t+1}, k \leftarrow k+1$$

and similarly for the $y$-player.

Then we have linear last-iterate convergence:

$$\text{DualityGap}\left(\boldsymbol{x}_t, \boldsymbol{y}_t\right) = O\left(\beta^t\right) \text{ for some } \beta \in (0,1)$$

# Monotone operator 1/2

Zero-sum game $G$: $\min_{\boldsymbol{x}\in\Delta_{d_1}} \max_{\boldsymbol{y}\in\Delta_m} \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{y}\rangle$.

Gradient operator $F_G(\boldsymbol{z}) := \begin{pmatrix} \boldsymbol{A}\boldsymbol{y} \\ -\boldsymbol{A}^\top \boldsymbol{x} \end{pmatrix}$ for $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \Delta_n \times \Delta_m$.

This is a *monotone* operator:

$$\langle F_G(\boldsymbol{z}) - F_G(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}'\rangle \geq 0, \forall\, \boldsymbol{z}, \boldsymbol{z}' \in \Delta_n \times \Delta_m.$$

OGD has last-iterate convergence for monotone operators [COZ22].

# Monotone operator 2/2

Smooth PRM$^+$ $\iff$ running OGD with operator $F$ defined as

$$F(\boldsymbol{z}) := \begin{pmatrix} \boldsymbol{A}\frac{\boldsymbol{z}_2}{\|\boldsymbol{z}_2\|_1} - \frac{\boldsymbol{z}_1^\top}{\|\boldsymbol{z}_1\|_1}\boldsymbol{A}\frac{\boldsymbol{z}_2}{\|\boldsymbol{z}_2\|_1} \cdot 1_n \\ -\boldsymbol{A}^\top\frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_1} + \frac{\boldsymbol{z}_2^\top}{\|\boldsymbol{z}_2\|_1}\boldsymbol{A}^\top\frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_1} \cdot 1_m \end{pmatrix}$$

for all $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$.

# Monotone operator 2/2

Smooth PRM$^+$ $\iff$ running OGD with operator $F$ defined as

$$F(\boldsymbol{z}) := \begin{pmatrix} \boldsymbol{A}\frac{\boldsymbol{z}_2}{\|\boldsymbol{z}_2\|_1} - \frac{\boldsymbol{z}_1^\top}{\|\boldsymbol{z}_1\|_1}\boldsymbol{A}\frac{\boldsymbol{z}_2}{\|\boldsymbol{z}_2\|_1} \cdot 1_n \\ -\boldsymbol{A}^\top\frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_1} + \frac{\boldsymbol{z}_2^\top}{\|\boldsymbol{z}_2\|_1}\boldsymbol{A}^\top\frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_1} \cdot 1_m \end{pmatrix}$$

for all $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$.

A simpler form:

$$F(\boldsymbol{z}) := \begin{pmatrix} \boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{y} \cdot 1_n \\ -\boldsymbol{A}^\top\boldsymbol{x} + \boldsymbol{y}^\top\boldsymbol{A}^\top\boldsymbol{x} \cdot 1_m \end{pmatrix}$$

for $\boldsymbol{x} = \frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_1}, \boldsymbol{y} = \frac{\boldsymbol{z}_2}{\|\boldsymbol{z}_2\|_1}$ for $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$.

The operator $F$ is **not** monotone.