Regret Matching+: Instability, average- and last-iterate convergence in games

Julien Grand-Clément, Assistant Professor, ISOM Department, HEC Paris
Joint with Yang Cai¹, Gabriele Farina², Christian Kroer³, Chung-Wei Lee⁴, Haipeng Luo⁴, Weiqiang Zheng¹

¹ Yale, ² MIT, ³ Columbia, ⁴ USC
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 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?
Our contributions:

1. We show a surprising “failure mode” of 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
Our contributions:

1. We show a surprising “failure mode” of RM$^+$, due to its instability.

2. We provide two fixes: restarting and smoothing.
   ⇒ New algorithms for game solving:
   \[ O(1/T) \] average convergence
   \[ O(1/\sqrt{T}) \] best-iterate convergence, last-iterate convergence

Why is this interesting?

1. Reconcile 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.
Presentation based on:

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 $x_t \in \Delta_n$ based on past observations
2. Observe the loss vector $\ell_t \in \mathbb{R}^n$
3. Suffer an instantaneous loss $\langle \ell_t, x_t \rangle \in \mathbb{R}$
Regret minimization: for $t = 1, \ldots, T$,

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

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

$$
\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \ell_{ta}.
$$
Regret minimization: for $t = 1, \ldots, T$,

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

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

$$\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$ 

A regret minimizer constructs a sequence of decisions $x_1, x_2, \ldots$ in $\Delta_n$ such that for any sequence of losses $\ell_1, \ell_2, \ldots$, we have

$$\lim_{T \to +\infty} \frac{\text{Reg}^T}{T} = 0.$$
Regret minimization: for $t = 1, \ldots, T$,

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

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

$$\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$

A regret minimizer constructs a sequence of decisions $x_1, x_2, \ldots$ in $\Delta_n$ such that for any sequence of losses $\ell_1, \ell_2, \ldots$, we have

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

Why do we care?
Regret minimization: for $t = 1, \ldots, T$,

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

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

$$\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \ell_{ta}.$$ 

A regret minimizer constructs a sequence of decisions $x_1, x_2, \ldots$ in $\Delta_n$ such that for any sequence of losses $\ell_1, \ell_2, \ldots$, we have

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

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:

\[
\min_{x \in \Delta_n} \max_{y \in \Delta_m} \langle x, Ay \rangle.
\]

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

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

\[
\text{DualityGap}(\hat{x}, \hat{y}) \leq \epsilon \Rightarrow (\hat{x}, \hat{y}) \text{ is } \epsilon\text{-Nash equilibrium}
\]
Regret minimization can be used to solve matrix games:

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

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

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

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

\textbf{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

$$\frac{\text{Reg}_1^T + \text{Reg}_2^T}{T}.$$
Rock Paper Scissors:

\[
\min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle, \quad A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}
\]

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

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

Losses for x-player: \(Ay\), loss for y-player: \(-A^T x\).
Rock Paper Scissors:

\[
\min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle, \quad A = \begin{pmatrix}
0 & -1 & 1 \\
1 & 0 & -1 \\
-1 & 1 & 0 \\
\end{pmatrix}
\]

Run Regret Matching\[^{+}\] (TBD) to generate \(x_1, \ldots, x_T\) and \(y_1, \ldots, y_T\).

Average iterates:

\[
\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t, \quad \bar{y}_t = \frac{1}{T} \sum_{t=1}^{T} y_t
\]
Rock Paper Scissors:

\[
\min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle, \quad A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}
\]

Run Regret Matching\(^+\) (TBD) to generate \(x_1, \ldots, x_T\) and \(y_1, \ldots, y_T\).

Average iterates:

\[
\bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t, \quad \bar{y}_t = \frac{1}{T} \sum_{t=1}^{T} y_t
\]

**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

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

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

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

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" $O\left(1/\sqrt{T}\right)$ convergence guarantees.
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

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

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

2. If $\text{Reg}_1^T + \text{Reg}_2^T = \tilde{O}\left(1\right)$ then convergence in $\tilde{O}\left(1/T\right)$.

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

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

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

2. If $\text{Reg}_1^T + \text{Reg}_2^T = \tilde{O}\left(1\right)$ 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$^+$ ($\text{RM}^+$) [TBJB15]

Start at $R_1 = 0 \in \mathbb{R}^n_+$, then

\[ x_t = \frac{R_t}{\|R_t\|_1} \]

\[ R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+ \]
Start at $R_1 = 0 \in \mathbb{R}^n$, then

$$x_t = R_t/\|R_t\|_1$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+$$
Start at $R_1 = 0 \in \mathbb{R}^n$, then

$$x_t = \frac{R_t}{\|R_t\|_1}$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+$$
Regret Matching\(^+\) (RM\(^+\)) [TBJB15]

Start at \(R_1 = 0 \in \mathbb{R}_+^n\), then

\[
x_t = \frac{R_t}{\|R_t\|_1}
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+
\]
Regret Matching\(^+\) (RM\(^+\)) [TBJB15]

Start at \( R_1 = 0 \in \mathbb{R}_+^n \), then

\[
x_t = \frac{R_t}{\|R_t\|_1}
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+
\]
Why is this called Regret Matching$^+$?

The update for $R_t$ is

$$R_{t+1} = [R_t + \langle l_t, x_t \rangle 1 - l_t]^+.$$
Why is this called Regret Matching$^+$?

The update for $R_t$ is

$$R_{t+1} = [R_t + \langle l_t, x_t \rangle 1 - l_t]^+.$$  

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

$$R_{T+1} = \sum_{t=1}^T \langle l_t, x_t \rangle 1 - \sum_{t=1}^T l_t.$$
Why is this called Regret Matching+$^+$?

The update for $R_t$ is

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$ 

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

$$R_{T+1} = \sum_{t=1}^{T} \langle \ell_t, x_t \rangle 1 - \sum_{t=1}^{T} \ell_t.$$ 

Recall the definition of the regret:

$$\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \ell_{ta}$$

$$= \max_{a \in \{1, \ldots, n\}} R_{T+1,a}.$$
Why is this called Regret Matching\(^+\)?

The update for \( R_t \) is

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]

Suppose we forget about the operator \([.]^+\), then

\[
R_{T+1} = \sum_{t=1}^{T} \langle \ell_t, x_t \rangle 1 - \sum_{t=1}^{T} \ell_t.
\]

Recall the definition of the regret:

\[
\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^{T} \langle \ell_t, x_t \rangle - \sum_{t=1}^{T} \ell_{ta}
\]

\[
= \max_{a \in \{1, \ldots, n\}} R_{T+1, a}
\]

\( \Rightarrow \) \( R_t \) is called the lifted regret and \( \text{Reg}^T \leq \|R_{T+1}\|_{\infty} \).
Why is this called Regret Matching$^+$?

The update for $R_t$ is

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$  

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

$$R_{T+1} = \sum_{t=1}^T \langle \ell_t, x_t \rangle 1 - \sum_{t=1}^T \ell_t.$$  

Recall the definition of the regret:

$$\text{Reg}^T := \max_{a \in \{1, \ldots, n\}} \sum_{t=1}^T \langle \ell_t, x_t \rangle - \sum_{t=1}^T \ell_{ta}$$

$$= \max_{a \in \{1, \ldots, n\}} R_{T+1,a}$$

$\Rightarrow R_t$ is called the *lifted regret* and $\text{Reg}^T \leq \|R_{T+1}\|_\infty$.  

$\Rightarrow x_t = R_t/\|R_t\|_1$: we play actions with large regrets.
Regret Matching\(^+\) (\(RM^+\)) [TBJB15]

Start at \(R_1 = 0 \in \mathbb{R}^n\), then

\[
x_t = \frac{R_t}{\|R_t\|_1},
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]

Why do we like this algorithm?

1. \(RM^+\) is a regret minimizer: \(\text{Reg}^T = O\left(\sqrt{T}\right)\).
Regret Matching$^+$ (RM$^+$) [TBJB15]

Start at $R_1 = 0 \in \mathbb{R}^n$, then

$$x_t = R_t / \|R_t\|_1,$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$

Why do we like this algorithm?

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

2. Geometric intuition: $\text{Reg}^T \leq \|R_{T+1}\|_\infty$. 


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

Start at $R_1 = 0 \in \mathbb{R}^n_+$, then

$$x_t = R_t / \| R_t \|_1,$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$ 

Why do we like this algorithm?

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

2. Geometric intuition: $\text{Reg}^T \leq \|R_{T+1}\|_\infty$.

3. Parameter-free: no step size to learn/choose.
Regret Matching\(^+\) (RM\(^+\)) [TBJB15]

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

\[
\begin{align*}
x_t &= \mathbf{R}_t / \| \mathbf{R}_t \|_1, \\
\mathbf{R}_{t+1} &= [\mathbf{R}_t + \langle \ell_t, x_t \rangle \mathbf{1} - \ell_t]^+. \end{align*}
\]

Why do we like this algorithm?

1. RM\(^+\) is a regret minimizer: \( \text{Reg}_T = O\left( \sqrt{T} \right) \).
2. Geometric intuition: \( \text{Reg}_T \leq \| \mathbf{R}_{T+1} \|_\infty \).
3. Parameter-free: no step size to learn/choose
4. Strong empirical performances, \( \approx 10x \) faster than \( O(1/T) \) algo\( s \) [BBJT15, MSB\(^+\)17, BS18, BS19, FKS21]...
Regret Matching\(^+\) (RM\(^+\)) [TBJB15]

Start at \( R_1 = 0 \in \mathbb{R}^n_+ \), then

\[
x_t = R_t / \| R_t \|_1,
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]

Why do we like this algorithm?

1. RM\(^+\) is a regret minimizer: \( \text{Reg}^T = O\left( \sqrt{T} \right) \).
2. Geometric intuition: \( \text{Reg}^T \leq \| R_{T+1} \|_\infty \).
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!
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

$$x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

$$
\hat{R}_t = [R_t + \langle \ell_{t-1}, x_{t-1} \rangle 1 - \ell_{t-1}]^+
$$

$$
x_t = \frac{\hat{R}_t}{\| \hat{R}_t \|_1},
$$

$$
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
$$
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

\[
x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

\[
x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

$$x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+. $$

\[ \mathbb{R}_+^2 \]
\[ \mathbb{R}_-^2 \]
\[ \Delta_2 \]
\[ R_t \]
\[ \hat{R}_t \]
\[ 0 \]
Main idea: use a prediction of \( \ell_t \) when computing \( x_t \).

*Build \( \hat{R}_t \) by predicting \( \ell_t \)*

\[
x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

\begin{align*}
\text{Build } \hat{R}_t \text{ by predicting } \ell_t \\
x_t &= \frac{\hat{R}_t}{\|\hat{R}_t\|_1}, \\
R_{t+1} &= [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\end{align*}
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

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

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$
Main idea: use a prediction of \( \ell_t \) when computing \( x_t \).

*Build \( \hat{R}_t \) by predicting \( \ell_t \)*

\[
x_t = \frac{\hat{R}_t}{\| \hat{R}_t \|_1},
\]

\[
R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.
\]
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

$$x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+. $$
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

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

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+. $$

1. Predictive RM$^+$ is a regret minimizer: $\text{Reg}^T = O\left(\sqrt{T}\right)$. 
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

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

$$x_t = \hat{R}_t / \|\hat{R}_t\|_1,$$
$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$

1. Predictive RM$^+$ is a regret minimizer: $\text{Reg}^T = O\left(\sqrt{T}\right)$.
2. Parameter-free: no step size to learn/choose
Main idea: use a prediction of $\ell_t$ when computing $x_t$.

BUILD $\hat{R}_t$ BY PREDICTING $\ell_t$ AS $\ell_{t-1}$

$$x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+. $$

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

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

$$x_t = \frac{\hat{R}_t}{\|\hat{R}_t\|_1},$$

$$R_{t+1} = [R_t + \langle \ell_t, x_t \rangle 1 - \ell_t]^+.$$

1. Predictive RM$^+$ is a regret minimizer: $\text{Reg}^T = O\left(\sqrt{T}\right)$.
2. Parameter-free: no step size to learn/choose
3. Strong empirical performances, vastly outperforms $O(1/T)$ algs [BBJT15, MSB$^+$17, BS18, BS19, FKS21].
4. But not known to ensure $O(1/T)$ convergence, despite optimism!
Recall that $x_t = R_t / \| R_t \|_1$.

Instability: $\| x_t - x_{t+1} \|_2$ may be large... despite small $\| R_t - R_{t+1} \|_2$. 

\[ \mathbb{R}_+^2 \]

\[ \mathbb{R}_-^2 \]

\[ 0 \]

\[ \Delta_2 \]
Recall that \( x_t = \frac{R_t}{\|R_t\|_1} \).

Instability: \( \|x_t - x_{t+1}\|_2 \) may be large... despite small \( \|R_t - R_{t+1}\|_2 \).
Recall that $x_t = \frac{R_t}{\|R_t\|_1}$.

Instability: $\|x_t - x_{t+1}\|_2$ may be large... despite small $\|R_t - R_{t+1}\|_2$. 
Recall that $x_t = R_t / \| R_t \|_1$.

Instability: $\| x_t - x_{t+1} \|_2$ may be large... despite small $\| R_t - R_{t+1} \|_2$. 
Instability in (predictive) $\text{RM}^+$

Instability happens because $\|R_t\|_1$ is small.
Instability in (predictive) $RM^+$

Instability happens because $\|R_t\|_1$ is small.

**Proposition**

Let $R_1, R_2 \in \mathbb{R}_+^n$ and $x_1 = R_1/\|R_1\|_1, x_2 = R_2/\|R_2\|_1$. Then

$$\|x_1 - x_2\|_2 \leq \frac{\sqrt{n}}{\max\{\|R_1\|_1, \|R_2\|_1\}} \cdot \|R_1 - R_2\|_2$$

(1)
Instability in (predictive) RM$^+$

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

$$\text{Reg}^T \leq \|R_{T+1}\|_\infty.$$
Example on a pathological example

Solving a small matrix game: \( \min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle \).

Running (vanilla) Predictive RM\(^+\):
Example on a pathological example

Solving a small matrix game: \( \min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle \).

Running (vanilla) Predictive RM\(^+\):

![Graphs showing the behavior of norms over iterations](image)
Example on a pathological example

Solving a small matrix game: \( \min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle \).

Running (vanilla) Predictive RM\(^+\):

![Graph showing strategy cycles](image)

After \(10^7\) iterations, \(x_t\) cycles between 5 strategies.

Recall that the loss for the y-player is \(-A^T x_t\)!
Example on a pathological example

Solving a small matrix game: \( \min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle \).

Running (vanilla) Predictive RM\(^+\):

![Graph showing duality gap vs. number of iterations](image)

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 $\|R_t\|_1$ is small.

Question:

How to ensure that $R_t$ is not too close to the origin $0$?
Toward stable Predictive RM$^+$: first idea

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

If $R_{t+1} \leq R_01$ then $R_{t+1} = R_01$. 

\[
\begin{align*}
  \mathbb{R}_2^+ - \mathbb{R}_2^- + \Delta_2 R_0 & R_0 1 \\
  0 & R_0 \\
\end{align*}
\]
Toward stable Predictive RM\(^+\): first idea

Restarting: run Predictive RM\(^+\), and at the end of every iteration:

\[ \text{If } R_{t+1} \leq R_01 \text{ then } R_{t+1} = R_01. \]

This can be done in linear time.

**Theorem**

Assume that each player runs Predictive RM\(^+\) with restarting with \( R_0 = XXX \).

Then max \( \{ \text{Reg}_1^T, \text{Reg}_2^T \} \) = \( O(T^{1/4}) \).

\( \Rightarrow \) Convergence to a Nash Equilibrium at a rate of \( O \left( \frac{1}{T^{3/4}} \right) \).
Toward stable $\text{RM}^+$: second idea

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

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

This ensures $R_t \in \{ R \in \mathbb{R}^n \mid R \geq 0, \langle R, 1 \rangle \geq R_0 \}$. 
Toward stable $\text{RM}^+$: second idea

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

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

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

\[
\mathbb{R}_+^2 - \mathbb{R}_-^2 \ni R_0 \ni \Delta_2 \ni \times R_{t+1}
\]
Toward stable $\text{RM}^+$: second idea

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

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

This ensures $R_t \in \{ R \in \mathbb{R}^n \mid R \geq 0, \langle R, 1 \rangle \geq R_0 \}$. 
Toward stable $\text{RM}^+:$ second idea

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

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

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

\[
\begin{align*}
\mathbb{R}^2_+ & \quad \mathbb{R}^2_- \\
\mathbb{R}^2_+ & \quad \Delta_2 \\
\mathbb{R}^2_- & \quad 0 \\
\mathbb{R}^2_+ & \quad R_0 \\
\mathbb{R}^2_- & \quad R_0 \\
\mathbb{R}^2_+ & \quad \mathbb{R}^2_- \\
\mathbb{R}^2_+ & \quad \Delta_2 \\
\mathbb{R}^2_- & \quad 0 \\
\mathbb{R}^2_+ & \quad R_0 \\
\mathbb{R}^2_- & \quad R_0 \\
\end{align*}
\]
Toward stable $\text{RM}^+$: second idea

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

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

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

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

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

$$\| \frac{R_1}{\|R_1\|_1} - \frac{R_2}{\|R_2\|_1} \|_2 \leq \frac{\sqrt{n}}{R_0} \cdot \|R_1 - R_2\|_2$$ (2)
Toward stable RM$^+$: second idea

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

\[ \text{If } \langle R_{t+1}, 1 \rangle \leq R_0 \text{ then replace } R_{t+1} \text{ by its projection on } R_0 \Delta_n. \]

**Theorem**

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

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

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


Example on a pathological example (continued)

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

Running Predictive RM$^+$ with restarting:
Solving a small matrix game: $\min_{x \in \Delta_3} \max_{y \in \Delta_3} \langle x, Ay \rangle$.

Running Predictive RM$^+$ with Smoothing:
Solving a small matrix game:

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

Comparing the average convergence to a Nash Equilibrium:
All the guarantees presented so far are for the average iterates:

\[ \bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t, \quad \bar{y}_t = \frac{1}{T} \sum_{t=1}^{T} y_t \]

How about convergence in \( x_T, y_T \), i.e., last-iterate convergence?
Last-iterate convergence

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

\[ \bar{x}_T = \frac{1}{T} \sum_{t=1}^{T} x_t, \quad \bar{y}_t = \frac{1}{T} \sum_{t=1}^{T} y_t \]

How about convergence in \( x_T, y_T \), 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.
Our contributions 1/3

⇒ RM$^+$ and Predictive RM$^+$ may diverge on a simple $3 \times 3$ matrix game.
⇒ Poor performance of the last iterates of RM$^+$ / PRM$^+$:

**Figure 5:** Last iterate performance of RM$^+$, PRM$^+$ and Smooth PRM$^+$. 
We could only prove convergence of $\text{RM}^+$ under very strong assumptions.

**Theorem**

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

- $x^*$ is the unique best-response to $y^*$
- $y^*$ is the unique best-response to $x^*$

Then $\text{RM}^+$ converges: the sequence $(x_t, y_t)_{t \in \mathbb{N}}$ has a limit.

Note: strict N.E. implies N.E. is unique and $(x^*, y^*)$ are deterministic.
Our contributions 2/3

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

**Theorem**

For Smooth Predictive RM$^+$, we show

1. Last-iterate convergence: the sequence $(x_t, y_t)_{t \in \mathbb{N}}$ has a limit.

2. Best-iterate convergence:
   For some $\alpha > 0$ and starting at $(x_0, y_0)$,
   \[
   \min_{t \in \{1, \ldots, T\}} \text{DualityGap} (x_t, y_t) = \frac{\alpha \cdot \text{dist} ((x_0, y_0), \mathcal{Z}^*)}{\sqrt{T}}
   \]
Let $\mathcal{Z}^* \subset \Delta_n \times \Delta_m$ be the set of Nash equilibria.

**Theorem**

For Smooth Predictive RM$^+$, we show

1. **Last-iterate convergence:** the sequence $(x_t, y_t)_{t \in \mathbb{N}}$ has a limit.

2. **Best-iterate convergence:**
   For some $\alpha > 0$ and starting at $(x_0, y_0)$,
   \[
   \min_{t \in \{1, \ldots, T\}} \text{DualityGap} (x_t, y_t) = \frac{\alpha \cdot \text{dist} ((x_0, y_0), \mathcal{Z}^*)}{\sqrt{T}}
   \]

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

\[
 c \cdot \text{dist} ((x_t, y_t), \mathcal{Z}^*) \leq \text{DualityGap}(x_t, y_t).
\]
There exists a time $\tilde{t} \in \{1, \ldots, T\}$ such that

$$\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), \mathcal{Z}^* \right) \leq \frac{\alpha}{c \sqrt{T}} \cdot \text{dist} \left( (x_0, y_0), \mathcal{Z}^* \right).$$
There exists a time \( \tilde{t} \in \{1, \ldots, T\} \) such that

\[
\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), Z^* \right) \leq \frac{\alpha}{c \sqrt{T}} \cdot \text{dist} \left( (x_0, y_0), Z^* \right).
\]

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

\( \Rightarrow \) in a constant number of steps, we halve the distance to \( Z^* \):

\[
\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), Z^* \right) \leq \frac{1}{2} \text{dist} \left( (x_0, y_0), Z^* \right).
\]
There exists a time $\tilde{t} \in \{1, \ldots, T\}$ such that

$$\operatorname{dist}((x_\tilde{t}, y_\tilde{t}), Z^*) \leq \frac{\alpha}{c\sqrt{T}} \cdot \operatorname{dist}((x_0, y_0), Z^*).$$

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

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

$$\operatorname{dist}((x_\tilde{t}, y_\tilde{t}), Z^*) \leq \frac{1}{2} \operatorname{dist}((x_0, y_0), Z^*).$$

$\Rightarrow$ Why not reinitializing the algorithm at time $\tilde{t}$: $(x_0, y_0) \leftarrow (x_\tilde{t}, y_\tilde{t})$?
There exists a time $\tilde{t} \in \{1, \ldots, T\}$ such that

$$\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), Z^* \right) \leq \frac{\alpha}{c\sqrt{T}} \cdot \text{dist} \left( (x_0, y_0), Z^* \right).$$

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

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

$$\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), Z^* \right) \leq \frac{1}{2} \text{dist} \left( (x_0, y_0), Z^* \right).$$

$\Rightarrow$ Why not reinitializing the algorithm at time $\tilde{t}$: $(x_0, y_0) \leftarrow (x_{\tilde{t}}, y_{\tilde{t}})$?

Problem: of course we can’t identify the time $\tilde{t}$...
There exists a time \( \tilde{t} \in \{1, \ldots, T\} \) such that

\[
\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), Z^* \right) \leq \frac{\alpha}{c \sqrt{T}} \cdot \text{dist} \left( (x_0, y_0), Z^* \right).
\]

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

\( \Rightarrow \) in a constant number of steps, we halve the distance to \( Z^* \):

\[
\text{dist} \left( (x_{\tilde{t}}, y_{\tilde{t}}), Z^* \right) \leq \frac{1}{2} \text{dist} \left( (x_0, y_0), Z^* \right).
\]

\( \Rightarrow \) Why not reinitializing the algorithm at time \( \tilde{t} \): \( (x_0, y_0) \leftarrow (x_{\tilde{t}}, y_{\tilde{t}}) \) ?

Problem: of course we can’t identify the time \( \tilde{t} \) ...

Solution: bound the distance to \( Z^* \) by distances between \( \hat{R}^t, R^{t+1}, R^t \).
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:

\[
\text{DualityGap} (x_t, y_t) = O (\beta^t) \quad \text{for some } \beta \in (0, 1)
\]
Conclusion

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

Limitations:
1. We lose 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!
Conclusion

• Better understanding of Regret Matching\(^+\) and predictive variants

• New algorithms with strong theoretical guarantees

• Limitations:
  1. We lose 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 lose 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.**  

Santiago R Balseiro and Yonatan Gur.  
**Learning in repeated auctions with budgets: Regret minimization and equilibrium.**  

Santiago R Balseiro, Haihao Lu, and Vahab Mirrokni.  
**The best of many worlds: Dual mirror descent for online allocation problems.**  
Noam Brown and Tuomas Sandholm.  
**Superhuman AI for heads-up no-limit poker: Libratus beats top professionals.**  

Noam Brown and Tuomas Sandholm.  
**Superhuman AI for multiplayer poker.**  

**Finite-time last-iterate convergence for learning in multi-player games.**  
Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. 
**Near-optimal no-regret learning in general games.**

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 
**Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent.**

Yoav Freund and Robert E Schapire. 
**Adaptive game playing using multiplicative weights.**

**Deepstack: Expert-level artificial intelligence in heads-up no-limit poker.**

Alexander Rakhlin and Karthik Sridharan.

**Online learning with predictable sequences.**


**Fast convergence of regularized learning in games.**
28, 2015.
**Mastering the game of go with deep neural networks and tree search.**  

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.
Theorem
Consider running Smooth Predictive RM$^+$, with the following trick:
At iteration $t$,

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

and similarly for the $y$-player.

Then we have linear last-iterate convergence:

$$\text{DualityGap} \left( x_t, y_t \right) = O \left( \beta^t \right) \text{ for some } \beta \in (0, 1)$$
Zero-sum game $G$: $\min_{x \in \Delta_{d_1}} \max_{y \in \Delta_m} \langle x, Ay \rangle$.

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

This is a monotone operator:

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

OGD has last-iterate convergence for monotone operators [COZ22].
Smooth PRM$^+$ $\iff$ running OGD with operator $F$ defined as

$$F(z) := \begin{pmatrix} A \frac{z_2}{\|z_2\|_1} - \frac{z_1^\top}{\|z_1\|_1} A \frac{z_2}{\|z_2\|_1} \cdot 1_n \\ -A^\top \frac{z_1}{\|z_1\|_1} + \frac{z_2^\top}{\|z_2\|_1} A^\top \frac{z_1}{\|z_1\|_1} \cdot 1_m \end{pmatrix}$$

for all $z = (z_1, z_2) \in \mathbb{R}_n^+ \times \mathbb{R}_m^+$. 
Smooth PRM\(^+\) \iff running OGD with operator \(F\) defined as

\[
F(z) := \begin{pmatrix}
A \frac{z_2}{\|z_2\|_1} - \frac{z_1^\top}{\|z_1\|_1} A \frac{z_2}{\|z_2\|_1} \cdot 1_n \\
-A^\top \frac{z_1}{\|z_1\|_1} + \frac{z_2^\top}{\|z_2\|_1} A^\top \frac{z_1}{\|z_1\|_1} \cdot 1_m
\end{pmatrix}
\]

for all \(z = (z_1, z_2) \in \mathbb{R}_+^n \times \mathbb{R}_+^m\).

A simpler form:

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

for \(x = \frac{z_1}{\|z_1\|_1}, y = \frac{z_2}{\|z_2\|_1}\) for \(z = (z_1, z_2) \in \mathbb{R}_+^n \times \mathbb{R}_+^m\).

The operator \(F\) is **not** monotone.