Lecture 9. Optimism for Fast Rates

Advanced Optimization (Fall 2024)

Peng Zhao

zhaop@lamda.nju.edu.cn

Nanjing University

Outline

- Online Games

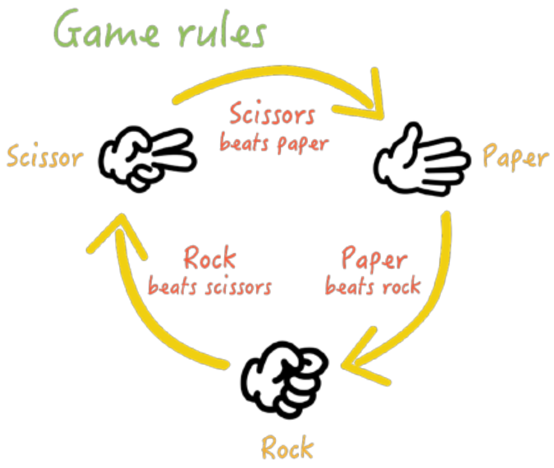
- Accelerated Methods

Part 1. Online Games

- Two-player Zero-sum Games
- Minimax Theorem
- Repeated Play
- Faster Convergence via Adaptivity

Classic Game: *Rock-Paper-Scissors game*

- Rock-Paper-Scissors game



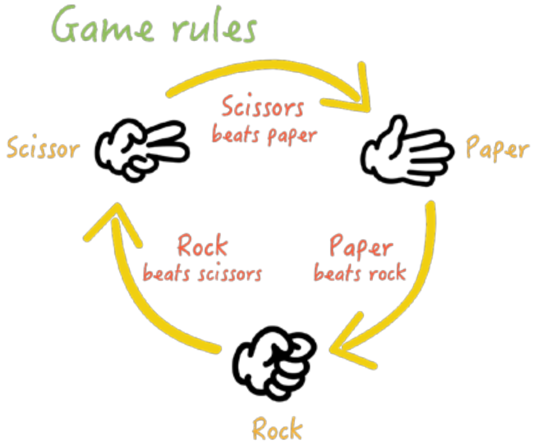
$$
\begin{array}{cc}
 & \begin{array}{ccc} \textbf{Rock} & \textbf{Paper} & \textbf{Scissors} \end{array} \\
\begin{array}{c} \textbf{Rock} \\ \textbf{Paper} \\ \textbf{Scissors} \end{array} & \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}
\end{array}
$$

- Strategy
  - *Pure* strategy: a fixed action, e.g., "Rock".
  - *Mixed* strategy: a *distribution* on all actions, e.g., ("Rock", "Paper", "Scissors") $= (1/3, 1/3, 1/3)$.

Two-Player Zero-Sum Games

- Terminology
  - ◇ game/payoff matrix $A \in [-1, 1]^{m \times n}$
  - ◇ two players
    - – player #1: x-player, row player, min player
    - – player #2: y-player, colume player, max player
  - ◇ action set (focusing on mixed strategy)
    - – player #1: $\Delta_m = \{\boldsymbol{p} \mid \sum_{i=1}^m p_i = 1, \text{ and } p_i \geq 0, \forall i \in [m]\}$.
    - – player #2: $\Delta_n = \{\boldsymbol{q} \mid \sum_{j=1}^n q_j = 1, \text{ and } q_j \geq 0, \forall j \in [n]\}$.

$$
\begin{array}{c c}
 & \begin{array}{ccc} \textbf{Rock} & \textbf{Paper} & \textbf{Scissors} \end{array} \\
\begin{array}{c} \textbf{Rock} \\ \textbf{Paper} \\ \textbf{Scissors} \end{array} & \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}
\end{array}
$$



Two-Player Zero-Sum Games

- The protocol:
  - The repeated game is denoted by a (payoff) matrix $A \in [-1, 1]^{m \times n}$.
  - The $\mathbf{x}$-player has m actions, and the $\mathbf{y}$-player has n actions.
  - The goal of $\mathbf{x}$-player is to ***minimize her loss*** and the goal of $\mathbf{y}$-player is to ***maximize her reward***.

- Given the action $(\mathbf{x}, \mathbf{y}) \in \Delta_m \times \Delta_n$, the loss and reward are the same.
  - expected loss of $\mathbf{x}$-player is $\mathbb{E}[\text{loss}] = \sum_{i \in [m]} x_i \sum_{j \in [n]} y_j A_{ij} = \mathbf{x}^\top A \mathbf{y}$.
  - expected reward of $\mathbf{y}$-player is $\mathbb{E}[\text{reward}] = \sum_{i \in [m]} x_i \sum_{j \in [n]} y_j A_{ij} = \mathbf{x}^\top A \mathbf{y}$.

Two-Player Zero-Sum Games

- Best response:
  - when $\mathbf{x}$-player plays a strategy $\bar{\mathbf{x}} \in \Delta_m$, the best response of $\mathbf{y}$-player is $\mathbf{y}^\dagger \in \arg\max_{\mathbf{y} \in \Delta_n} \bar{\mathbf{x}}^\top A \mathbf{y}$;
  - when $\mathbf{y}$-player plays a strategy $\bar{\mathbf{y}} \in \Delta_n$, the best response of $\mathbf{x}$-player is $\mathbf{x}^\dagger \in \arg\min_{\mathbf{x} \in \Delta_m} \mathbf{x}^\top A \bar{\mathbf{y}}$;

Nash Equilibrium

- What is a desired state for the two players in games?

Definition 2 (Nash equilibrium). A mixed strategy $(\mathbf{x}^\star, \mathbf{y}^\star)$ is called a Nash equilibrium if neither player has an incentive to change her strategy given that the opponent is keeping hers, i.e., for all $\mathbf{x} \in \Delta_m$ and $\mathbf{y} \in \Delta_n$, it holds that

$$\mathbf{x}^{\star\top} A\mathbf{y} \leq \mathbf{x}^{\star\top} A\mathbf{y}^\star \leq \mathbf{x}^\top A\mathbf{y}^\star.$$

y-player's goal is to *maximize* her reward, changing from $\mathbf{y}^\star$ to $\mathbf{y}$ will decrease reward.

x-player's goal is to *minimize* her loss, changing from $\mathbf{x}^\star$ to $\mathbf{x}$ will increase loss.

$\Rightarrow$ *Does the Nash equilibrium always exist for zero-sum games?*

Von Neumann's Minimax Theorem

- For two-player zero-sum games, minimax equals maximin.

Theorem 1. *For any two-player zero-sum game $A \in [-1, 1]^{m \times n}$, we have*

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} = \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}.$$

- Relationship between **Nash equilibrium** and **minimax solution.**

Theorem 2. *A pair of mixed strategy $(\mathbf{x}^\star, \mathbf{y}^\star)$ is a Nash equilibrium of the game A, if and only if it is also a minimax solution (the optimizer of problem $\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} = \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}$), i.e.,*

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y}, \mathbf{y}^\star \in \arg\max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}.$$

Proof of Theorem 2

Theorem 2. *A pair of mixed strategy* $(\mathbf{x}^\star, \mathbf{y}^\star)$ *is a Nash equilibrium of the game* A*, **if and only if** it is also a minimax solution (the optimizer of problem* $\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} = \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}$*), i.e.,*

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y}, \mathbf{y}^\star \in \arg\max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}.$$

Proof: ***(Nash ⇒ minimax solution)***

Denote by $(\mathbf{x}^\star, \mathbf{y}^\star)$ a Nash equilibrium, and we have

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} \leq \max_{\mathbf{y}} \ \mathbf{x}^{\star\top} A\mathbf{y} \underset{\text{(Nash)}}{=} \mathbf{x}^{\star\top} A\mathbf{y}^\star \underset{\text{(Nash)}}{=} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}^\star \leq \max_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$$

By Von Neumann's minimax theorem, the above inequality is in fact an equality. $\square$

Proof of Theorem 2

Theorem 2. *A pair of mixed strategy* $(\mathbf{x}^\star, \mathbf{y}^\star)$ *is a Nash equilibrium of the game* A*, **if and only if** it is also a minimax solution (the optimizer of problem* $\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} = \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}$*), i.e.,*

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y}, \mathbf{y}^\star \in \arg\max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}.$$

***Proof*:** *(minimax solution ⇒ Nash)*

Denote by $(\mathbf{x}^\dagger, \mathbf{y}^\dagger)$ a minimax solution, we have

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} \underset{\text{(def)}}{=} \max_{\mathbf{y}} \ \mathbf{x}^{\dagger\top} A\mathbf{y} \geq \mathbf{x}^{\dagger\top} A\mathbf{y}^\dagger \geq \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}^\dagger \underset{\text{(def)}}{=} \max_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$$

By Von Neumann's minimax theorem, the above inequality is again an equality. □

Minimax Strategy and Maximin Strategy

- ***minimax*** strategy

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$$

x-player goes first, and given $\mathbf{x}$, the worst-case response of $\mathbf{y}$-player is $\max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$, so the best way for $\mathbf{x}$-player would be $\operatorname{argmin}_{\mathbf{x}} \max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$.

- ***maximin*** strategy

$$\mathbf{y}^\star \in \arg\max_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$$

y-player goes first, and given $\mathbf{y}$, the worst-case response of $\mathbf{x}$-player is $\min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$, so the best way for $\mathbf{y}$-player would be $\operatorname{argmax}_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$.

Minimax Strategy and Maximin Strategy

- A natural consequence

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} \geq \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}$$

Intuition: *there should be no disadvantage of playing second*

- ***minimax*** strategy

  $\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$

  x-player goes first, and given $\mathbf{x}$, the worst-case response of $\mathbf{y}$-player is $\max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$, so the best way for $\mathbf{x}$-player would be $\mathrm{argmin}_{\mathbf{x}} \max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y}$.

- ***maximin*** strategy

  $\mathbf{y}^\star \in \arg\max_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$

  y-player goes first, and given $\mathbf{y}$, the worst-case response of $\mathbf{x}$-player is $\min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$, so the best way for $\mathbf{y}$-player would be $\mathrm{argmax}_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$.

Proof: Define $\mathbf{x}^\star \in \arg\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y}$ and $\mathbf{y}^\star \in \arg\max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}$.

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A\mathbf{y} \underset{\text{(def)}}{=} \max_{\mathbf{y}} \ \mathbf{x}^{\star\top} A\mathbf{y} \geq \mathbf{x}^{\star\top} A\mathbf{y}^\star \geq \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}^\star \underset{\text{(def)}}{=} \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A\mathbf{y}.$$

$\square$

Von Neumann's Minimax Theorem

- For two-player zero-sum games, minimax equals maximin.

Theorem 1. *For any two-player zero-sum game $A \in [-1,1]^{m \times n}$, we have*

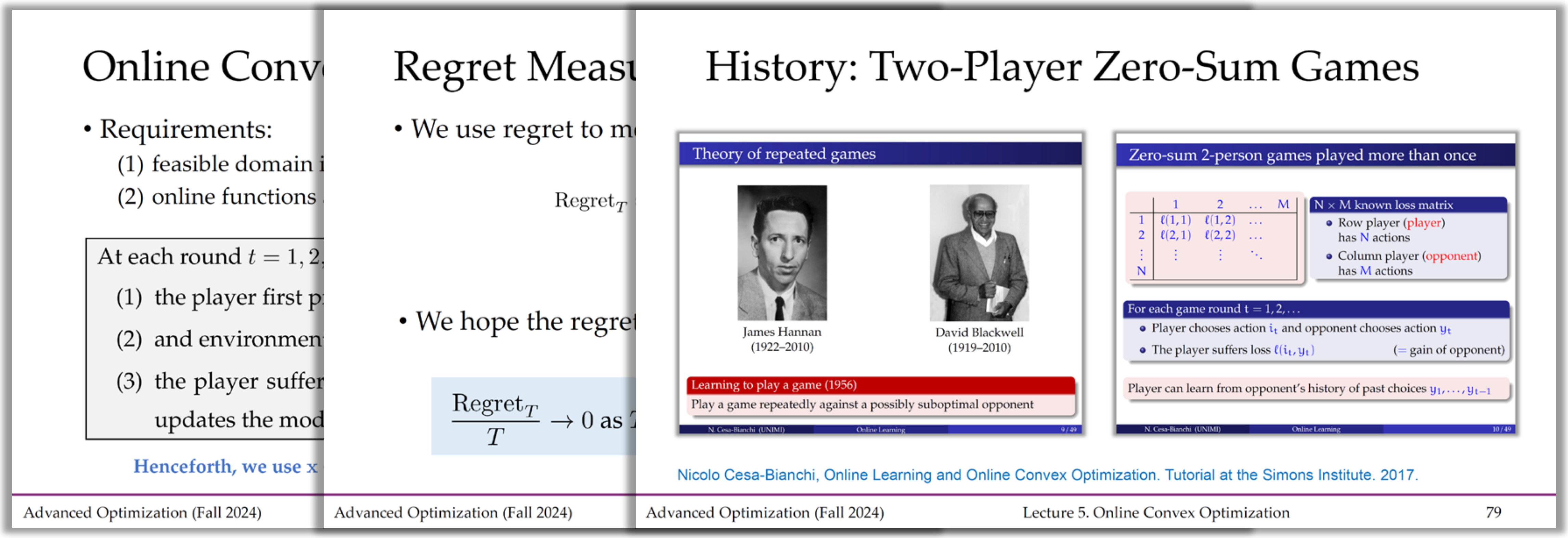
$$\min_{\mathbf{x}} \max_{\mathbf{y}} \; \mathbf{x}^\top A\mathbf{y} = \max_{\mathbf{y}} \min_{\mathbf{x}} \; \mathbf{x}^\top A\mathbf{y}.$$

We have proved the direction that $\min_{\mathbf{x}} \max_{\mathbf{y}} \mathbf{x}^\top A\mathbf{y} \geq \max_{\mathbf{y}} \min_{\mathbf{x}} \mathbf{x}^\top A\mathbf{y}$, whereas the reverse direction is not straightforward.

The original proof relies on a fixed-point theorem (which is highly non-trivial), and we here will present a simple and ***constructive*** proof from an online learning perspective.

Connection with Online Learning

- Recall the OCO framework, regret notion, and the history bits.



Constructive Proof of Theorem 1

- ***Our goal***: to prove Von Neumann's Minimax Theorem

Theorem 1. *For any two-player zero-sum game $A \in [-1,1]^{m \times n}$, we have*

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A \mathbf{y} = \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A \mathbf{y}.$$

As the one side is trivial, we only need to prove

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \ \mathbf{x}^\top A \mathbf{y} \leq \max_{\mathbf{y}} \min_{\mathbf{x}} \ \mathbf{x}^\top A \mathbf{y}$$

which can be realized by the ***repeated play***.

Online Games with Repeated Play

- Consider the following ***repeated play*** setting.

At each round $t = 1, 2, \ldots, T$:

(1) x-player picks a mixed strategy $\mathbf{x}_t \in \Delta_m$

(2) simitaneously y-player picks a mixed strategy $\mathbf{y}_t \in \Delta_n$

(3) x-player and y-player submit their strategies together

(4) x-player receives loss $\mathbf{x}_t^\top A \mathbf{y}_t$ and observes $A\mathbf{y}_t$; y-player receives loss $-\mathbf{x}_t^\top A \mathbf{y}_t$ and observes $-A^\top \mathbf{x}_t$

The online function that x-player receives is $f_t^{\mathbf{x}}(\cdot) \triangleq \cdot^\top A \mathbf{y}_t$. *assume gradient feedback*

$\Longrightarrow$ $\mathbf{y}_t$ can depend on $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$, meaning that x-player is facing an *adptive adversary*.

Online Games with Repeated Play

Deploying the no-regret online algorithm for two players

- denote by $\mathrm{Reg}_T^{\mathbf{x}}$ the regret suffered by the x-player
- denote by $\mathrm{Reg}_T^{\mathbf{y}}$ the regret suffered by the y-player

Key idea: use $\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top A \mathbf{y}_t$ as a bridge between $\min_{\mathbf{x}} \max_{\mathbf{y}}$ and $\max_{\mathbf{y}} \min_{\mathbf{x}}$

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top A \mathbf{y}_t &\leq \min_{\mathbf{x}\in\Delta_m} \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}^\top A \mathbf{y}_t + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} \\
&= \min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A \bar{\mathbf{y}}_T + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} \qquad (\bar{\mathbf{y}}_T \triangleq \tfrac{1}{T}\textstyle\sum_{t=1}^{T} \mathbf{y}_t) \\
&\leq \max_{\mathbf{y}\in\Delta_n} \min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A \mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T}
\end{aligned}$$

Online Games with Repeated Play

Deploying the no-regret online algorithm for two players

- denote by $\mathrm{Reg}_T^{\mathbf{x}}$ the regret suffered by the x-player
- denote by $\mathrm{Reg}_T^{\mathbf{y}}$ the regret suffered by the y-player

Key idea: use $\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t^\top A\mathbf{y}_t$ as a bridge between $\min_{\mathbf{x}} \max_{\mathbf{y}}$ and $\max_{\mathbf{y}} \min_{\mathbf{x}}$

$$
\begin{aligned}
-\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t^\top A\mathbf{y}_t &\leq \min_{\mathbf{y}\in\Delta_n} -\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \\
&= \min_{\mathbf{y}\in\Delta_n} -\bar{\mathbf{x}}_T^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \qquad \left(\bar{\mathbf{x}}_T \triangleq \frac{1}{T}\sum_{t=1}^T \mathbf{x}_t\right) \\
&\leq \max_{\mathbf{x}\in\Delta_m} \min_{\mathbf{y}\in\Delta_n} -\mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} = -\min_{\mathbf{x}\in\Delta_m} \max_{\mathbf{y}\in\Delta_n} \mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T}
\end{aligned}
$$

Online Games with Repeated Play

Key idea: use $\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top A\mathbf{y}_t$ as a bridge between $\min_{\mathbf{x}}\max_{\mathbf{y}}$ and $\max_{\mathbf{y}}\min_{\mathbf{x}}$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top A\mathbf{y}_t \leq \max_{\mathbf{y}\in\Delta_n}\min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} \tag{1}$$

$$-\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top A\mathbf{y}_t \leq -\min_{\mathbf{x}\in\Delta_m}\max_{\mathbf{y}\in\Delta_n} \mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \tag{2}$$

$$\min_{\mathbf{x}\in\Delta_m}\max_{\mathbf{y}\in\Delta_n} \mathbf{x}^\top A\mathbf{y} \overset{(2)}{\leq} \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t^\top A\mathbf{y}_t + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \overset{(1)}{\leq} \max_{\mathbf{y}\in\Delta_n}\min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T}$$

If $\mathrm{Reg}_T^{\mathbf{x}}$ and $\mathrm{Reg}_T^{\mathbf{y}}$ are sublinear in T, the gap becomes to 0 when $T \to \infty$.

$\square$

Nash Equilibrium Calculation

- How to **compute** an approximate a Nash equilibrium?

At each round $t = 1, 2, \ldots, T$:

(1) $\mathbf{x}$-player picks a mixed strategy $\mathbf{x}_t \in \Delta_m$

(2) simitaneously $\mathbf{y}$-player picks a mixed strategy $\mathbf{y}_t \in \Delta_n$

(3) $\mathbf{x}$-player and $\mathbf{y}$-player submit their strategies together

(4) $\mathbf{x}$-player receives loss $\mathbf{x}_t^\top A\mathbf{y}_t$ and observes $A\mathbf{y}_t$; $\mathbf{y}$-player receives loss $-\mathbf{x}_t^\top A\mathbf{y}_t$ and observes $-A^\top\mathbf{x}_t$

Submit $\bar{\mathbf{x}}_T \triangleq \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$, ***and*** $\bar{\mathbf{y}}_T \triangleq \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t$

Nash Equilibrium Calculation

- **Duality Gap:** measure the quality

$$\textsc{Dual-Gap}(\bar{\mathbf{x}}_T, \bar{\mathbf{y}}_T) \triangleq \max_{\mathbf{y}\in\Delta_n} \bar{\mathbf{x}}_T^\top A\mathbf{y} - \min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A\bar{\mathbf{y}}_T$$

- From the previous analysis, we know that the algorithm ensures:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y}_t \leq \min_{\mathbf{x}\in\Delta_m} \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}^\top A\mathbf{y}_t + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} = \min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A\bar{\mathbf{y}}_T + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} \leq \max_{\mathbf{y}\in\Delta_n}\min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T}$$

$$-\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y}_t \leq \min_{\mathbf{y}\in\Delta_n} -\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} = \min_{\mathbf{y}\in\Delta_n} -\bar{\mathbf{x}}_T^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \leq \max_{\mathbf{x}\in\Delta_m}\min_{\mathbf{y}\in\Delta_n} -\mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T}$$

Nash Equilibrium Calculation

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y}_t \le \min_{\mathbf{x}\in\Delta_m}\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}^\top A\mathbf{y}_t + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} = \min_{\mathbf{x}\in\Delta_m}\mathbf{x}^\top A\bar{\mathbf{y}}_T + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} \le \max_{\mathbf{y}\in\Delta_n}\min_{\mathbf{x}\in\Delta_m}\mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T}$$

$$-\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y}_t \le \min_{\mathbf{y}\in\Delta_n} -\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} = \min_{\mathbf{y}\in\Delta_n} -\bar{\mathbf{x}}_T^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \le \max_{\mathbf{x}\in\Delta_m}\min_{\mathbf{y}\in\Delta_n} -\mathbf{x}^\top A\mathbf{y} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T}$$

$$\Longrightarrow \quad {\mathbf{x}^\star}^\top A\mathbf{y}^\star \le \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y}_t + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \le \min_{\mathbf{x}\in\Delta_m}\mathbf{x}^\top A\bar{\mathbf{y}}_T + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T}$$

$$\max_{\mathbf{y}\in\Delta_n}\bar{\mathbf{x}}_T^\top A\mathbf{y} \le \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t^\top A\mathbf{y}_t + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T} \le {\mathbf{x}^\star}^\top A\mathbf{y}^\star + \frac{\mathrm{Reg}_T^{\mathbf{x}}}{T} + \frac{\mathrm{Reg}_T^{\mathbf{y}}}{T}$$

$$\Longrightarrow \quad \textsc{Dual-Gap}(\bar{\mathbf{x}}_T, \bar{\mathbf{y}}_T) = \max_{\mathbf{y}\in\Delta_n}\bar{\mathbf{x}}_T^\top A\mathbf{y} - \min_{\mathbf{x}\in\Delta_m}\mathbf{x}^\top A\bar{\mathbf{y}}_T \le 2(\mathrm{Reg}_T^{\mathbf{x}} + \mathrm{Reg}_T^{\mathbf{y}})/T$$

Nash Equilibrium Calculation

- So far, we have

$$\textsc{Dual-Gap}(\bar{\mathbf{x}}_T, \bar{\mathbf{y}}_T) = \max_{\mathbf{y}\in\Delta_n} \bar{\mathbf{x}}_T^\top A\mathbf{y} - \min_{\mathbf{x}\in\Delta_m} \mathbf{x}^\top A\bar{\mathbf{y}}_T \leq 2(\mathrm{Reg}_T^{\mathbf{x}} + \mathrm{Reg}_T^{\mathbf{y}})/T$$

This result implies a ***constructive algorithm*** for Nash equilibrium calculation with a non-asymptotic guarantee.

⇨

If $\mathbf{x}$-player and $\mathbf{y}$-player both run ***Hedge*** algorithm, then

- $\mathrm{Reg}_T^{\mathbf{x}} = \mathcal{O}(\sqrt{T})$, and $\mathrm{Reg}_T^{\mathbf{y}} = \mathcal{O}(\sqrt{T})$,
- the convergence rate is $\mathcal{O}(T^{-1/2})$.

Faster Convergence via Gradient Variation

- Can we do ***faster*** than the $\mathcal{O}(\sqrt{T})$ rate?

  Yes! We can use the Optimistic Online Mirror Descent of the last lecture.

- Recall in gradient-variation regret, the negative term is crucial.

$$\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta \langle \nabla f_{t-1}(\mathbf{x}_{t-1}), \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$$

$$\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$$

Gradient Variation

$$\Longrightarrow \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{u}) \leq \eta \sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1})\|_2^2 + \frac{D^2}{2\eta} - \frac{1}{4\eta}\sum_{t=1}^T \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2$$

(negative term)

Gradient-Variation Bound

Definition 3 (Gradient Variation)**.** Let T be the time horizon and $\mathcal{X} \subseteq \mathbb{R}^d$ be the feasible domain. For the function sequence $f_1, \ldots, f_T$ with $f_t : \mathcal{X} \mapsto \mathbb{R}$ for $t \in [T]$, its gradient variation is defined as

$$V_T = \sum_{t=2}^{T} \sup_{\mathbf{x} \in \mathcal{X}} \|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|_2^2$$

Optimistic OMD for Gradient-Variation Bound

$$\mathbf{x}_t = \arg\min_{\mathbf{x} \in \mathcal{X}} \eta_t \langle \nabla f_{t-1}(\mathbf{x}_{t-1}), \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$$

$$\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \eta_t \langle \nabla f_t(\mathbf{x}_t), \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$$

Gradient-Variation Bound

Theorem 4 (Gradient Variation Regret Bound). *Assume that $\psi(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ and f_t is L-smooth for all $t \in [T]$, when setting $\eta_t = \min\{\frac{1}{4L}, \frac{D}{\sqrt{1+\widetilde{V}_{t-1}}}\}$ and $M_t = \nabla f_{t-1}(\mathbf{x}_{t-1})$, the regret of Optimistic OMD to any comparator $\mathbf{u} \in \mathcal{X}$ is*

$$\mathrm{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \le \mathcal{O}\left(\sqrt{1+V_T}\right)$$

where $\widetilde{V}_{t-1} = \sum_{s=2}^{t-1} \|\nabla f_s(\mathbf{x}_{s-1}) - \nabla f_{s-1}(\mathbf{x}_{s-1})\|_2^2$ is the empirical estimates of V_t.

Proof.

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \le \sum_{t=1}^{T} \eta_t \|\nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1})\|_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta_t}\left(\|\mathbf{u} - \widehat{\mathbf{x}}_t\|_2^2 - \|\mathbf{u} - \widehat{\mathbf{x}}_{t+1}\|_2^2\right)$$
$$- \sum_{t=1}^{T} \frac{1}{2\eta_t}\left(\|\widehat{\mathbf{x}}_{t+1} - \mathbf{x}_t\|_2^2 + \|\mathbf{x}_t - \widehat{\mathbf{x}}_t\|_2^2\right)$$

(negative term)

Faster Convergence via Gradient Variation

- Can we do ***faster*** than the $\mathcal{O}(\sqrt{T})$ rate?

Yes! We can use the Optimistic Online Mirror Descent of the last lecture.

If $\mathbf{x}$-player runs OOMD with NE-entropy and gradients $\mathbf{g}_t^{\mathbf{x}} \triangleq A\mathbf{y}_t$ for $t \in [T]$:

$$\mathrm{Reg}_T^{\mathbf{x}} = \sum_{t=1}^{T} \langle A\mathbf{y}_t, \mathbf{x}_t - \mathbf{x} \rangle \lesssim \frac{1}{\eta^{\mathbf{x}}} \boxed{+\eta^{\mathbf{x}} \sum_{t=2}^{T} \|A\mathbf{y}_t - A\mathbf{y}_{t-1}\|_\infty^2} \boxed{- \frac{1}{\eta^{\mathbf{x}}} \sum_{t=2}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_1^2}$$

Similarly,

$$\mathrm{Reg}_T^{\mathbf{y}} = \sum_{t=1}^{T} \langle -A^\top \mathbf{x}_t, \mathbf{y}_t - \mathbf{y} \rangle \lesssim \frac{1}{\eta^{\mathbf{y}}} \boxed{+\eta^{\mathbf{y}} \sum_{t=2}^{T} \|A^\top \mathbf{x}_t - A^\top \mathbf{x}_{t-1}\|_\infty^2} \boxed{- \frac{1}{\eta^{\mathbf{y}}} \sum_{t=2}^{T} \|\mathbf{y}_t - \mathbf{y}_{t-1}\|_1^2}$$

$\Rightarrow$ $\mathrm{Reg}_T^{\mathbf{x}} + \mathrm{Reg}_T^{\mathbf{y}} = \mathcal{O}(1)$, which leads to a much faster $\mathcal{O}(T^{-1})$ convergence rate! $\square$

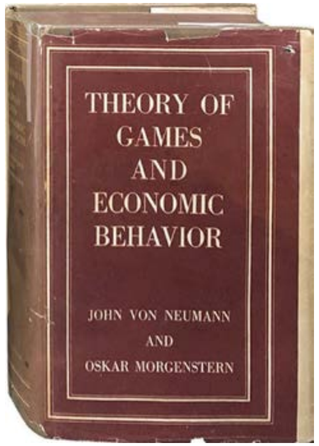
History bits: Game Theory

- **John von Neumann**

John von Neumann was a Hungarian mathematician.

- ❑ He has been credited with founding game theory based on his paper in ***1928***.
- ❑ In 1944, he wrote, alongside Oskar Morgestern, the seminal book ***Theory of Games and Economic Behavior***.
- ❑ Definitely, he also has a lot of other achievements in mathematics, computer science, and many other areas.

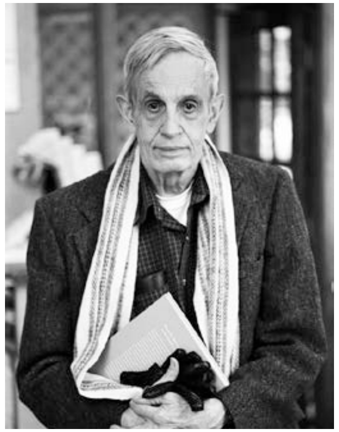
John von Neumann
1903-1957



History bits: Game Theory

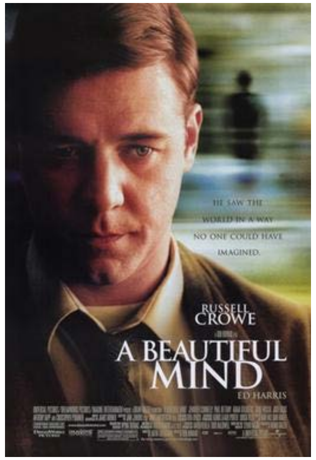
- **John Forbes Nash Jr.**

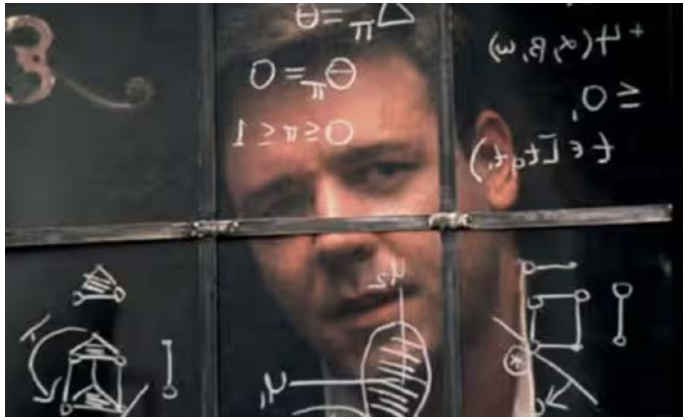
John Forbes Nash Jr., American mathematician who was awarded the ***1994 Nobel Prize*** for Economics.



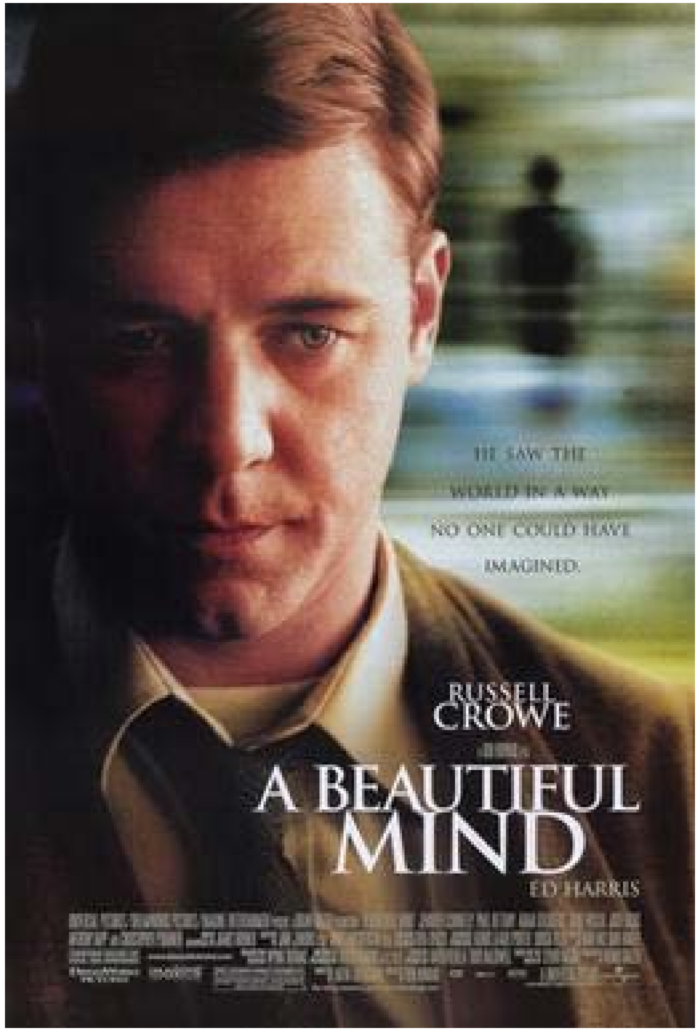
John Forbes Nash Jr.
1928-2015

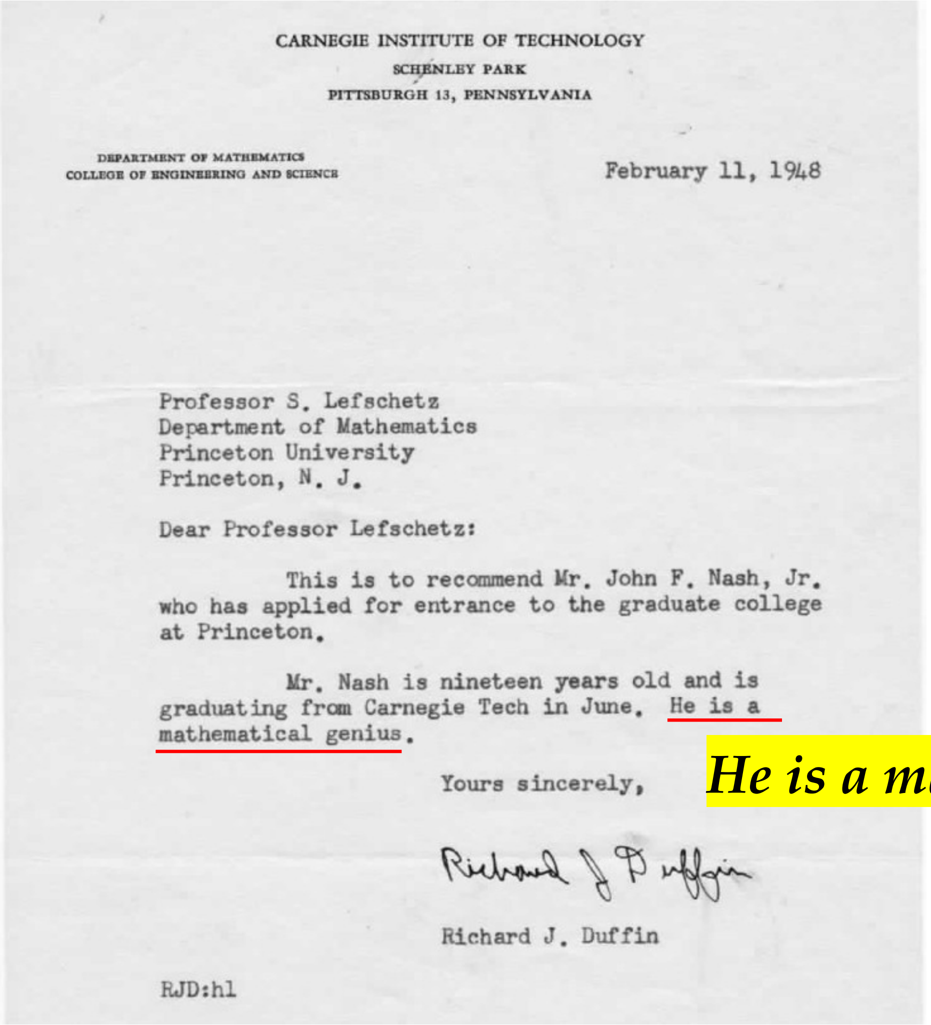
He submitted a paper to the Proceedings of the National Academy of Sciences in 1949, where he proved that ***an equilibrium exists in every finite game***.





History bits: Game Theory





CARNEGIE INSTITUTE OF TECHNOLOGY
SCHENLEY PARK
PITTSBURGH 13, PENNSYLVANIA

DEPARTMENT OF MATHEMATICS
COLLEGE OF ENGINEERING AND SCIENCE

February 11, 1948

Professor S. Lefschetz
Department of Mathematics
Princeton University
Princeton, N. J.

Dear Professor Lefschetz:

This is to recommend Mr. John F. Nash, Jr. who has applied for entrance to the graduate college at Princeton.

Mr. Nash is nineteen years old and is graduating from Carnegie Tech in June. He is a mathematical genius.

Yours sincerely,

Richard J. Duffin

Richard J. Duffin

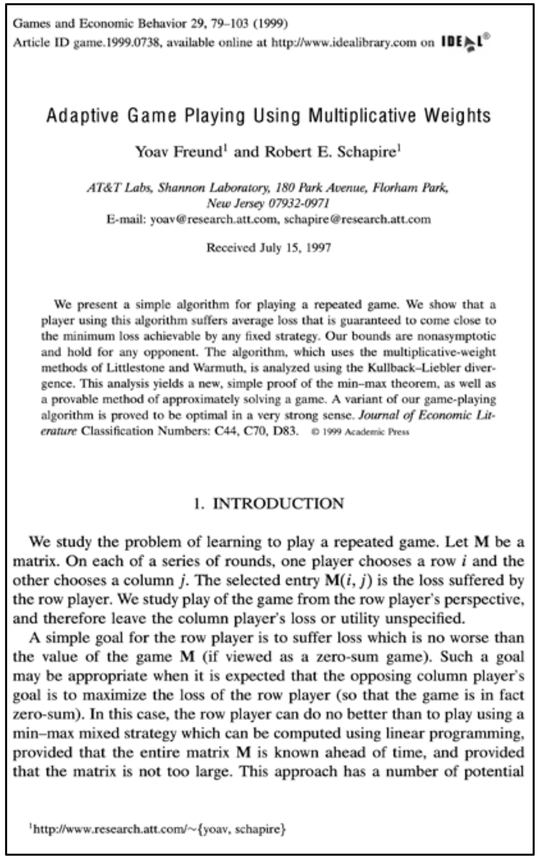
RJD:hl

He is a mathematical genius.

History bits: Online Learning in Games

- **Yoav Freund & Robert Schapire**

Yoav Freund and Robert Schapire's seminal paper in 1999 reveals the fundamental relationship between game theory and online learning, specifically, ***"a simple proof of the min-max theorem"***.



Games and Economic Behavior 29, 79–103 (1999)
Article ID game.1999.0738, available online at http://www.idealibrary.com on IDEAL®

Adaptive Game Playing Using Multiplicative Weights

Yoav Freund[1] and Robert E. Schapire[1]

AT&T Labs, Shannon Laboratory, 180 Park Avenue, Florham Park, New Jersey 07932-0971
E-mail: yoav@research.att.com, schapire@research.att.com

Received July 15, 1997

We present a simple algorithm for playing a repeated game. We show that a player using this algorithm suffers average loss that is guaranteed to come close to the minimum loss achievable by any fixed strategy. Our bounds are nonasymptotic and hold for any opponent. The algorithm, which uses the multiplicative-weight methods of Littlestone and Warmuth, is analyzed using the Kullback–Liebler divergence. This analysis yields a new, simple proof of the min–max theorem, as well as a provable method of approximately solving a game. A variant of our game-playing algorithm is proved to be optimal in a very strong sense. *Journal of Economic Literature* Classification Numbers: C44, C70, D83.

1. INTRODUCTION

We study the problem of learning to play a repeated game. Let **M** be a matrix. On each of a series of rounds, one player chooses a row i and the other chooses a column j. The selected entry $\mathbf{M}(i, j)$ is the loss suffered by the row player. We study play of the game from the row player's perspective, and therefore leave the column player's loss or utility unspecified.

A simple goal for the row player is to suffer loss which is no worse than the value of the game **M** (if viewed as a zero-sum game). Such a goal may be appropriate when it is expected that the opposing column player's goal is to maximize the loss of the row player (so that the game is in fact zero-sum). In this case, the row player can do no better than to play using a min–max mixed strategy which can be computed using linear programming, provided that the entire matrix **M** is known ahead of time, and provided that the matrix is not too large. This approach has a number of potential

[1] http://www.research.att.com/~{yoav, schapire}

Robert Schapire
1963-now

Yoav Freund
1961-now

Reference: Y. Freund and R. Schapire. Adaptive Game Playing Using Multiplicative Weights. Games and Economic Behavior, 1999.

History bits: Prediction with Expert Advice

Yoav Freund

Robert Schapire

Goldel Prize 2003

This paper introduced AdaBoost, an adaptive algorithm to improve the accuracy of hypotheses in machine learning. The algorithm demonstrated novel possibilities in analyzing data and is a permanent contribution to science even beyond computer science.

JOURNAL OF COMPUTER AND SYSTEM SCIENCES **55**, 119–139 (1997)
ARTICLE NO. SS971504

A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*

Yoav Freund and Robert E. Schapire†

AT&T Labs, 180 Park Avenue, Florham Park, New Jersey 07932

Received December 19, 1996

In the first part of the paper we consider the problem of dynamically apportioning resources among a set of options in a worst-case on-line framework. The model we study can be interpreted as a broad, abstract extension of the well-studied on-line prediction model to a general decision-theoretic setting. We show that the multiplicative weight-update Littlestone–Warmuth rule can be adapted to this model, yielding bounds that are slightly weaker in some cases, but applicable to a considerably more general class of learning problems. We show how the resulting learning algorithm can be applied to a variety of problems, including gambling, multiple-outcome prediction, repeated games, and prediction of points in $\mathbb{R}^n$. In the second part of the paper we apply the multiplicative weight-update technique to derive a new boosting algorithm. This boosting algorithm does not require any prior knowledge about the performance of the weak learning algorithm. We also study generalizations of the new boosting algorithm to the problem of learning functions whose range, rather than being binary, is an arbitrary finite set or a bounded segment of the real line.

converting a "weak" PAC learning algorithm that performs just slightly better than random guessing into one with arbitrarily high accuracy.

We formalize our *on-line allocation model* as follows. The allocation agent A has N options or *strategies* to choose from; we number these using the integers $1, ..., N$. At each time step $t = 1, 2, ..., T$, the allocator A decides on a distribution $\mathbf{p}^t$ over the strategies; that is $p_i^t \geqslant 0$ is the amount allocated to strategy i, and $\sum_{i=1}^N p_i^t = 1$. Each strategy i then suffers some *loss* ℓ_i^t which is determined by the (possibly adversarial) "environment." The loss suffered by A is then $\sum_{i=1}^n p_i^t \ell_i^t = \mathbf{p}^t \cdot \boldsymbol{\ell}^t$, i.e., the average loss of the strategies with respect to A's chosen allocation rule. We call this loss function the *mixture loss*.

In this paper, we always assume that the loss suffered by any strategy is bounded so that, without loss of generality, $\ell_i^t \in [0, 1]$. Besides this condition, we make no assumptions

Reference: Y. Freund and R. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. JCSS 1997.

History bits: Online Learning in Games

Optimization, Learning, and Games with Predictable Sequences

Alexander Rakhlin
University of Pennsylvania

Karthik Sridharan
University of Pennsylvania

Abstract

We provide several applications of Optimistic Mirror Descent, an online learning algorithm based on the idea of predictable sequences. First, we recover the Mirror Prox algorithm for offline optimization, prove an extension to Hölder-smooth functions, and apply the results to saddle-point type problems. Next, we prove that a version of Optimistic Mirror Descent (which has a close relation to the Exponential Weights algorithm) can be used by two strongly-uncoupled players in a finite zero-sum matrix game to converge to the minimax equilibrium at the rate of $\mathcal{O}((\log T)/T)$. This addresses a question of Daskalakis et al [6]. Further, we consider a partial information version of the problem. We then apply the results to convex programming and exhibit a simple algorithm for the approximate Max Flow problem.

1 Introduction

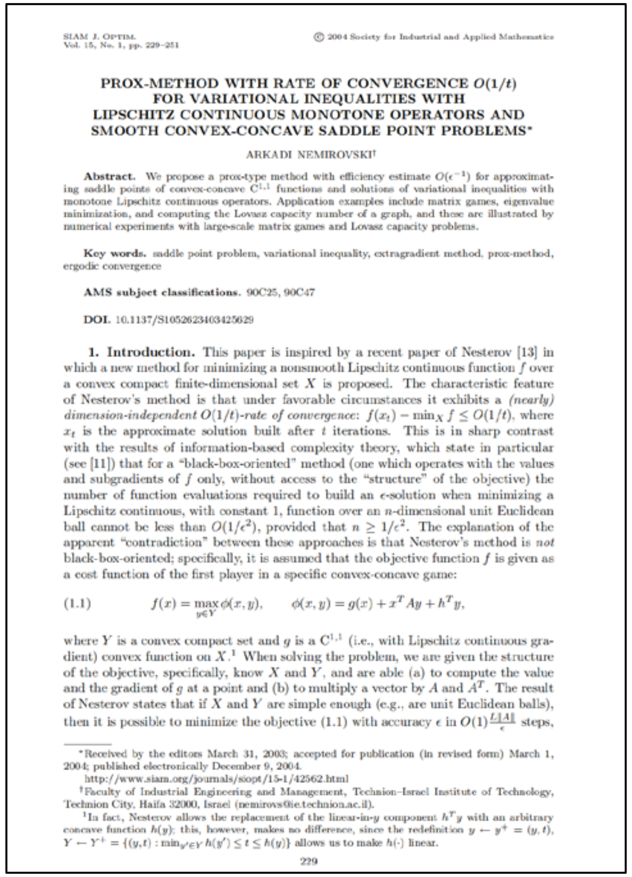
Recently, no-regret algorithms have received increasing attention in a variety of communities, including theoretical computer science, optimization, and game theory [3, 1]. The wide applicability of these algorithms is arguably due to the black-box regret guarantees that hold for arbitrary sequences. However, such regret guarantees can be loose if the sequence being encountered is not "worst-case". The reduction in "arbitrariness" of the sequence can arise from the particular structure of the problem at hand, and should be exploited. For instance, in some applications of online methods, the sequence comes from an additional computation done by the learner, thus being far from arbitrary.

One way to formally capture the partially benign nature of data is through a notion of predictable sequences [11]. We exhibit applications of this idea in several domains. First, we show that the Mirror Prox method [9], designed for optimizing non-smooth structured saddle-point problems, can be viewed as an instance of the predictable sequence approach. Predictability in this case is due precisely to smoothness of the inner optimization part and the saddle-point structure of the problem. We extend the results to Hölder-smooth functions, interpolating between the case of well-predictable gradients and "unpredictable" gradients.

Second, we address the question raised in [6] about existence of "simple" algorithms that converge at the rate of $\tilde{\mathcal{O}}(T^{-1})$ when employed in an uncoupled manner by players in a zero-sum finite matrix game, yet maintain the usual $\mathcal{O}(T^{-1/2})$ rate against arbitrary sequences. We give a positive answer and exhibit a fully adaptive algorithm that does not require the prior knowledge of whether the other player is collaborating. Here, the additional predictability comes from the fact that both players attempt to converge to the minimax value. We also tackle a partial information version of the problem where the player has only access to the real-valued payoff of the mixed actions played by the two players on each round rather than the entire vector.

Our third application is to convex programming: optimization of a linear function subject to convex constraints. This problem often arises in theoretical computer science, and we show that the idea of

Optimization, learning, and games with predictable sequences. NIPS 2013.



SIAM J. OPTIM.
Vol. 15, No. 1, pp. 229–251

PROX-METHOD WITH RATE OF CONVERGENCE $O(1/t)$ FOR VARIATIONAL INEQUALITIES WITH LIPSCHITZ CONTINUOUS MONOTONE OPERATORS AND SMOOTH CONVEX-CONCAVE SADDLE POINT PROBLEMS*

ARKADI NEMIROVSKI†

Abstract. We propose a prox-type method with efficiency estimate $O(\epsilon^{-1})$ for approximating saddle points of convex-concave $C^{1,1}$ functions and solutions of variational inequalities with monotone Lipschitz continuous operators. Application examples include matrix games, eigenvalue minimization, and computing the Lovasz capacity number of a graph, and these are illustrated by numerical experiments with large-scale matrix games and Lovasz capacity problems.

Key words. saddle point problem, variational inequality, extragradient method, prox-method, ergodic convergence

AMS subject classifications. 90C25, 90C47

DOI. 10.1137/S1052623403425629

1. Introduction. This paper is inspired by a recent paper of Nesterov [13] in which a new method for minimizing a nonsmooth Lipschitz continuous function f over a convex compact finite-dimensional set X is proposed. The characteristic feature of Nesterov's method is that under favorable circumstances it exhibits a *(nearly) dimension-independent* $O(1/t)$*-rate of convergence*: $f(x_t) - \min_X f \leq O(1/t)$, where x_t is the approximate solution built after t iterations. This is in sharp contrast with the results of information-based complexity theory, which state in particular (see [11]) that for a "black-box-oriented" method (one which operates with the values and subgradients of f only, without access to the "structure" of the objective) the number of function evaluations required to build an ϵ-solution when minimizing a Lipschitz continuous, with constant 1, function over an n-dimensional unit Euclidean ball cannot be less than $O(1/\epsilon^2)$, provided that $n \geq 1/\epsilon^2$. The explanation of the apparent "contradiction" between these approaches is that Nesterov's method is *not* black-box-oriented; specifically, it is assumed that the objective function f is given as a cost function of the first player in a specific convex-concave game:

$$f(x) = \max_{y \in Y} \phi(x, y), \qquad \phi(x, y) = g(x) + x^T A y + h^T y, \tag{1.1}$$

where Y is a convex compact set and g is a $C^{1,1}$ (i.e., with Lipschitz continuous gradient) convex function on X.[1] When solving the problem, we are given the structure of the objective, specifically, know X and Y, and are able (a) to compute the value and the gradient of g at a point and (b) to multiply a vector by A and A^T. The result of Nesterov states that if X and Y are simple enough (e.g., are unit Euclidean balls), then it is possible to minimize the objective (1.1) with accuracy ϵ in $O(1)\frac{L\|A\|}{\epsilon}$ steps,

*Received by the editors March 31, 2003; accepted for publication (in revised form) March 1, 2004; published electronically December 9, 2004.
http://www.siam.org/journals/siopt/15-1/42562.html

†Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel (nemirovs@ie.technion.ac.il).

[1] In fact, Nesterov allows the replacement of the linear-in-y component $h^T y$ with an arbitrary concave function $h(y)$; this, however, makes no difference, since the redefinition $y \leftarrow y^+ = (y, t)$, $Y \leftarrow Y^+ = \{(y, t) : \min_{y' \in Y} h(y') \leq t \leq h(y)\}$ allows us to make $h(\cdot)$ linear.

229

Nemirovski. Prox-Method with Rate of Convergence O(1/t) for Variational Inequalities with Lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems. SIAM Journal on OPT., 2004.

History bits: Online Learning in Games

Fast Convergence of Regularized Learning in Games

Vasilis Syrgkanis
Microsoft Research
New York, NY
vasy@microsoft.com

Alekh Agarwal
Microsoft Research
New York, NY
alekha@microsoft.com

Haipeng Luo
Princeton University
Princeton, NJ
haipengl@cs.princeton.edu

Robert E. Schapire
Microsoft Research
New York, NY
schapire@microsoft.com

Abstract

We show that natural classes of regularized learning algorithms with a form of recency bias achieve faster convergence rates to approximate efficiency and to coarse correlated equilibria in multiplayer normal form games. When each player in a game uses an algorithm from our class, their individual regret decays at $O(T^{-3/4})$, while the sum of utilities converges to an approximate optimum at $O(T^{-1})$–an improvement upon the worst case $O(T^{-1/2})$ rates. We show a black-box reduction for any algorithm in the class to achieve $\tilde{O}(T^{-1/2})$ rates against an adversary, while maintaining the faster rates against algorithms in the class. Our results extend those of Rakhlin and Shridharan [17] and Daskalakis et al. [4], who only analyzed two-player zero-sum games for specific algorithms.

1 Introduction

What happens when players in a game interact with one another, all of them acting independently and selfishly to maximize their own utilities? If they are smart, we intuitively expect their utilities — both individually and as a group — to grow, perhaps even to approach the best possible. We also expect the dynamics of their behavior to eventually reach some kind of equilibrium. Understanding these dynamics is central to game theory as well as its various application areas, including economics, network routing, auction design, and evolutionary biology.

It is natural in this setting for the players to each make use of a no-regret learning algorithm for making their decisions, an approach known as *decentralized no-regret dynamics*. No-regret algorithms are a strong match for playing games because their regret bounds hold even in adversarial environments. As a benefit, these bounds ensure that each player's utility approaches optimality. When played against one another, it can also be shown that the sum of utilities approaches an approximate optimum [2, 18], and the player strategies converge to an equilibrium under appropriate conditions [6, 1, 8], at rates governed by the regret bounds. Well-known families of no-regret algorithms include multiplicative-weights [13, 7], Mirror Descent [14], and Follow the Regularized/Perturbed Leader [12]. (See [3, 19] for excellent overviews.) For all of these, the average regret vanishes at the worst-case rate of $O(1/\sqrt{T})$, which is unimprovable in fully adversarial scenarios.

However, the players in our setting are facing other similar, predictable no-regret learning algorithms, a chink that hints at the possibility of improved convergence rates for such dynamics. This was first observed and exploited by Daskalakis et al. [4]. For two-player zero-sum games, they developed a decentralized variant of Nesterov's accelerated saddle point algorithm [15] and showed that each player's average regret converges at the remarkable rate of $O(1/T)$. Although the resulting

NIPS 2015
best paper award

No-Regret Learning in Time-Varying Zero-Sum Games

Mengxiao Zhang [*1] **Peng Zhao** [*2] **Haipeng Luo** [1] **Zhi-Hua Zhou** [2]

Abstract

Learning from repeated play in a fixed two-player zero-sum game is a classic problem in game theory and online learning. We consider a variant of this problem where the game payoff matrix changes over time, possibly in an adversarial manner. We first present three performance measures to guide the algorithmic design for this problem: 1) the well-studied *individual regret*, 2) an extension of *duality gap*, and 3) a new measure called *dynamic Nash Equilibrium regret*, which quantifies the cumulative difference between the player's payoff and the minimax game value. Next, we develop a single parameter-free algorithm that *simultaneously* enjoys favorable guarantees under all these three performance measures. These guarantees are adaptive to different non-stationarity measures of the payoff matrices and, importantly, recover the best known results when the payoff matrix is fixed. Our algorithm is based on a two-layer structure with a meta-algorithm learning over a group of black-box base-learners satisfying a certain property, along with several novel ingredients specifically designed for the time-varying game setting. Empirical results further validate the effectiveness of our algorithm.

1. Introduction

Repeated play in a fixed two-player zero-sum game, a fundamental problem in the interaction between game theory and online learning, has been extensively studied in recent decades. In particular, many efforts have been devoted to designing online algorithms such that both players achieve small individual regret (that is, difference between one's cumulative payoff and that of the best fixed action) while at the same time the dynamics of the players' strategy leads to a Nash equilibrium, a pair of strategies that neither player has incentive to deviate from; see for example (Freund & Schapire, 1999; Rakhlin & Sridharan, 2013; Daskalakis et al., 2015; Syrgkanis et al., 2015; Chen & Peng, 2020; Wei et al., 2021; Hsieh et al., 2021; Daskalakis et al., 2021).

In contrast to this large body of studies for learning over a *fixed* zero-sum game, repeated play over a sequence of *time-varying* games, the focus of this paper and a ubiquitous scenario in practice, is much less explored. While minimizing individual regret still makes perfect sense in this case, it is not immediately clear what other desirable game-theoretic guarantees are that generalize the concept of approaching a Nash equilibrium when the game is fixed. As far as we know, Cardoso et al. (2019) are the first to explicitly consider this problem. They proposed the notion of Nash-Equilibrium regret (NE-regret) as the performance measure, which quantifies the difference between the learners' cumulative payoff and the minimax value of the cumulative payoff matrix. The authors proposed an algorithm with $\tilde{\mathcal{O}}(\sqrt{T})$ NE-regret after T rounds of play and, importantly, proved that no algorithm can simultaneously achieve sublinear NE-regret and sublinear individual regret for both players.

Our work starts by questioning whether the NE-regret of Cardoso et al. (2019) is indeed a good performance measure for the problem of learning in time-varying games, especially given its incompatibility with the arguably most standard goal of having small individual regret. We then discover that measuring performance with NE-regret can in fact be highly unreasonable: we show an example (in Section 3) where even the two players perform perfectly (in the sense that they play the corresponding Nash equilibrium in every round), the resulting NE-regret is still *linear* in T!

Motivated by this observation, we revisit the basic problem of how to measure the algorithm's performance in such a time-varying game setting. Concretely, we consider three performance measures that we believe are appropriate and natural: 1) the standard individual regret; 2) the direct generalization of cumulative duality gap from a fixed game to a varying game; and 3) a new measure called *dynamic NE-regret*, which quantifies the difference between the learner's cumulative payoff and the cumulative minimax game value (instead of the minimax value of the cumulative payoff ma-

*Equal contribution, in alphabetical order. [1]University of Southern California [2]National Key Laboratory for Novel Software Technology, Nanjing University. Correspondence to: Mengxiao Zhang <mengxiao.zhang@usc.edu>, Peng Zhao <zhaop@lamda.nju.edu.cn>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Fast convergence of regularized learning in games. NIPS 2015.

No-Regret Learning in Time-Varying Games. ICML 2022.

Part 2. Accelerated Methods

- Weighted Online-to-Batch Conversion

- Accelerated Rates by Optimistic OMD

Accelerated Methods

- Recall that ***accelerated*** rates can be achieved for smooth convex optimization using Nesterov's Accelerated GD.

Theorem 3. *Let f be convex and L-smooth. Nesterov's accelerated GD is configured as*

$$\mathbf{x}_{t+1} = \mathbf{y}_t - \frac{1}{L}\nabla f(\mathbf{y}_t), \quad \mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t),$$

where $\lambda_0 = 0$, $\lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}$, *and* $\beta_t = \frac{\lambda_t - 1}{\lambda_{t+1}}$. *Then, we have*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{2L\|\mathbf{x}_1 - \mathbf{x}^\star\|^2}{T^2} = \mathcal{O}\left(\frac{1}{T^2}\right).$$

In our previous lecture, we prove this accelerated rate by the generalized one-step improvement property and a variety of algebraic tricks.

Acceleration by Optimistic OMD

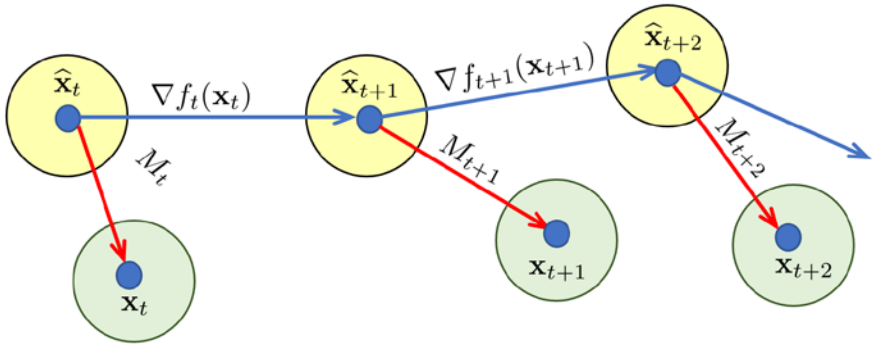
- We now present ***a new algorithm based on optimistic OMD*** with an accelerated rate for smooth convex optimization.

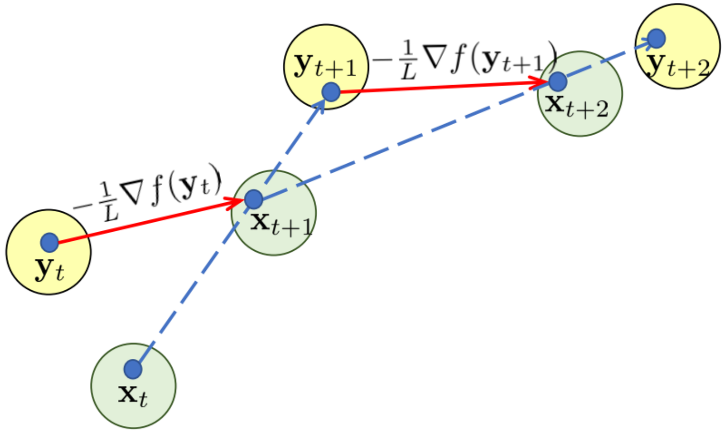
$$\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{\eta_t \langle M_t, \mathbf{x}\rangle + \mathcal{D}_\psi(\mathbf{x}, \widehat{\mathbf{x}}_t)\right\}$$

$$\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{\eta_t \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}\rangle + \mathcal{D}_\psi(\mathbf{x}, \widehat{\mathbf{x}}_t)\right\}$$

$$\mathbf{x}_{t+1} = \mathbf{y}_t - \frac{1}{L}\nabla f(\mathbf{y}_t)$$

$$\mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t)$$





Acceleration by Optimistic OMD

There are two key components:

- **Weighted Online-to-Batch Conversion**

  This is used to reduce the offline optimization to online optimization, but now we need a weighted version to achieve the potential acceleration.

- **Optimism Design**

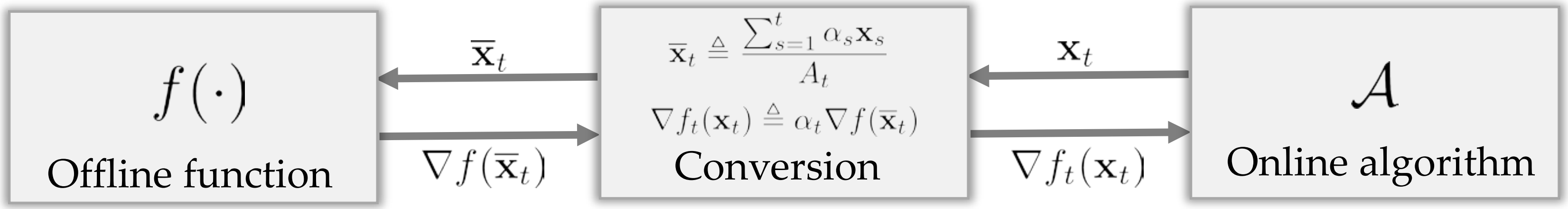
  This is used to achieve the desired vanishing regret in online optimization, in which the optimism design is crucial. It is essential to leverage the special structure of the problem.

Weighted Online-to-Batch Conversion

- Reducing *offline optimization* as an *online optimization*.

Algorithm 1 Weighted Online-to-Batch Conversion Template

Online algorithm $\mathcal{A}$, $\alpha_t > 0$.
for $t = 1, 2, \ldots, T$ **do**
  Obtain $\mathbf{x}_t$ from $\mathcal{A}$
  Submit $\overline{\mathbf{x}}_t = \frac{\sum_{s=1}^{t} \alpha_s \mathbf{x}_s}{A_t}$ with $A_t \triangleq \sum_{s=1}^{t} \alpha_s$
  Receive $\nabla f(\overline{\mathbf{x}}_t)$
  Send $\alpha_t \nabla f(\overline{\mathbf{x}}_t)$ as $\nabla f_t(\mathbf{x}_t)$ to $\mathcal{A}$
end for



Weighted Online-to-Batch Conversion

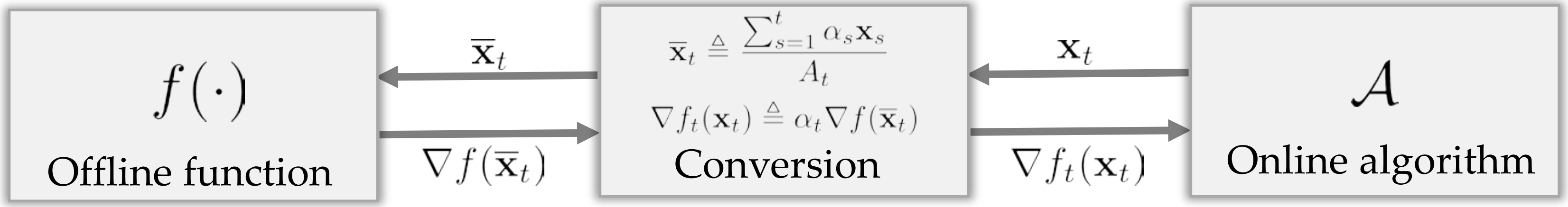
- Reducing *offline optimization* as an *online optimization*.

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated):*

$$\overline{\mathbf{x}}_t = \frac{\sum_{s=1}^t \alpha_s \mathbf{x}_s}{A_t},$$

with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \leq \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$



Weighted Online-to-Batch Conversion

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated): $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$, with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:*

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \le \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$

- When $\alpha_t = 1$ for all $t \in [T]$, it recovers the standard online-to-batch conversion, with $A_T = T$.
- But we can set α_t larger to make the denominator larger, such that we may have a chance to achieve a faster rate than the standard $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ one.

Weighted Online-to-Batch Conversion

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated): $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$, with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:*

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \leq \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$

Proof: First, by convexity we have

$$\begin{aligned}
\sum_{t=1}^T \alpha_t (f(\overline{\mathbf{x}}_t) - f(\mathbf{x}^\star)) &\leq \sum_{t=1}^T \alpha_t \langle \nabla f(\overline{\mathbf{x}}_t), \overline{\mathbf{x}}_t - \mathbf{x}^\star \rangle \\
&= \underbrace{\sum_{t=1}^T \alpha_t \langle \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}_{\triangleq \mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)} + \sum_{t=1}^T \alpha_t \langle \nabla f(\overline{\mathbf{x}}_t), \overline{\mathbf{x}}_t - \mathbf{x}_t \rangle
\end{aligned}$$

Weighted Online-to-Batch Conversion

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated): $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$, with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:*

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \le \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$

Proof: First, by convexity we have

$$\sum_{t=1}^T \alpha_t (f(\overline{\mathbf{x}}_t) - f(\mathbf{x}^\star)) \le \mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star) + \sum_{t=1}^T \alpha_t \langle \nabla f(\overline{\mathbf{x}}_t), \overline{\mathbf{x}}_t - \mathbf{x}_t \rangle$$

Notice the following two facts

$$\begin{aligned} \sum_{s=1}^t \alpha_s \mathbf{x}_s &= A_t \overline{\mathbf{x}}_t = A_{t-1}\overline{\mathbf{x}}_t + \alpha_t \overline{\mathbf{x}}_t \\ \sum_{s=1}^t \alpha_s \mathbf{x}_s &= \sum_{s=1}^{t-1} \alpha_s \mathbf{x}_s + \alpha_t \mathbf{x}_t = A_{t-1}\overline{\mathbf{x}}_{t-1} + \alpha_t \mathbf{x}_t \end{aligned} \implies \alpha_t(\overline{\mathbf{x}}_t - \mathbf{x}_t) = A_{t-1}(\overline{\mathbf{x}}_{t-1} - \overline{\mathbf{x}}_t)$$

Weighted Online-to-Batch Conversion

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated): $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$, with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:*

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \leq \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star\rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$

Proof: Further using the convexity property, we get

$$\begin{aligned}\sum_{t=1}^T \alpha_t (f(\overline{\mathbf{x}}_t) - f(\mathbf{x}^\star)) &\leq \mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star) - \sum_{t=1}^T A_{t-1}\langle \nabla f(\overline{\mathbf{x}}_t), \overline{\mathbf{x}}_t - \overline{\mathbf{x}}_{t-1}\rangle \\ &\leq \mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star) - \sum_{t=1}^T A_{t-1}(f(\overline{\mathbf{x}}_t) - f(\overline{\mathbf{x}}_{t-1}))\end{aligned}$$

This implies that $\quad A_T(f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star)) \leq \mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)$ $\square$

Weighted Online-to-Batch Conversion

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated): $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$, with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:*

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \le \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$

Set weights $\alpha_t = t$ for all $t \in [T]$, then $A_T = \mathcal{O}(T^2)$.

We aim to use online algorithm ensuring $\mathcal{O}(1)$ regret.

⇒ *Optimistic OMD with a suitable optimism design!*

Theorem 3. *Let f be convex and L-smooth. Nesterov's accelerated GD is configured as*

$$\mathbf{x}_{t+1} = \mathbf{y}_t - \frac{1}{L}\nabla f(\mathbf{y}_t), \quad \mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t),$$

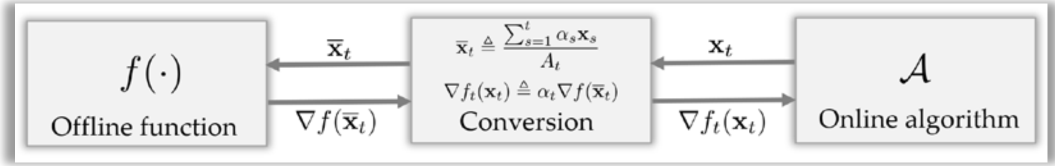
where $\lambda_0 = 0$, $\lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}$, and $\beta_t = \frac{\lambda_t - 1}{\lambda_{t+1}}$. Then, we have

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \le \frac{2L\|\mathbf{x}_1 - \mathbf{x}^\star\|^2}{T^2} = \mathcal{O}\left(\frac{1}{T^2}\right).$$

Accelerated Rates by Optimistic OMD

$\overline{\mathbf{x}}_t = \frac{1}{A_t} \sum_{s=1}^{t} \alpha_s \mathbf{x}_s$

- Can we achieve an $\mathcal{O}(1)$ regret for weighted online-to-batch conversion?



Yes! We can use the Optimistic Online Mirror Descent of the last lecture.

- Recall in gradient-variation regret, the negative term is crucial.

$$\boxed{\begin{aligned} \mathbf{x}_t &= \arg\min_{\mathbf{x} \in \mathcal{X}} \eta \langle M_t, \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2 \\ \widehat{\mathbf{x}}_{t+1} &= \arg\min_{\mathbf{x} \in \mathcal{X}} \eta \langle \nabla f_t(\mathbf{x}_t), \mathbf{x} \rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2 \end{aligned}}$$

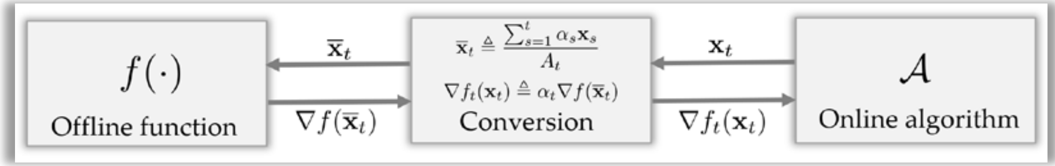
$$\Longrightarrow \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \le \frac{D^2}{2\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t) - M_t\|_2^2 - \frac{1}{4\eta} \sum_{t=1}^{T} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2$$

(negative term)

Accelerated Rates by Optimistic OMD

$\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^{t}\alpha_s \mathbf{x}_s$

- Can we achieve an $\mathcal{O}(1)$ regret for weighted online-to-batch conversion?



Yes! We can use the Optimistic Online Mirror Descent of the last lecture.

$\nabla f_t(\mathbf{x}_t) = \alpha_t \nabla f(\overline{\mathbf{x}}_t)$, $M_t = \alpha_t \nabla f(\widetilde{\mathbf{x}}_t)$, with $\widetilde{\mathbf{x}}_t$ to be determined:

$$\Rightarrow \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \leq \frac{D^2}{2\eta} + \eta \sum_{t=1}^{T} \|\alpha_t \nabla f(\overline{\mathbf{x}}_t) - \alpha_t \nabla f(\widetilde{\mathbf{x}}_t)\|_2^2 - \frac{1}{4\eta}\sum_{t=1}^{T}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2$$

(L-smoothness)

$$\leq \frac{D^2}{2\eta} + \eta \sum_{t=1}^{T} \alpha_t^2 L^2 \|\overline{\mathbf{x}}_t - \widetilde{\mathbf{x}}_t\|_2^2 - \frac{1}{4\eta}\sum_{t=1}^{T}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2$$

Optimism Design

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \le \frac{D^2}{2\eta} + \eta \sum_{t=1}^{T} \alpha_t^2 L^2 \|\bar{\mathbf{x}}_t - \widetilde{\mathbf{x}}_t\|_2^2 - \frac{1}{4\eta} \sum_{t=1}^{T} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2$$

- Optimism design: approximate $\bar{\mathbf{x}}_t$ as possible as we can

by def $\bar{\mathbf{x}}_t = \frac{1}{A_t}\left(\sum_{s=1}^{t-1} \alpha_s \mathbf{x}_s + \alpha_t \mathbf{x}_t\right)$,

we set $\widetilde{\mathbf{x}}_t \triangleq \frac{1}{A_t}\left(\sum_{s=1}^{t-1} \alpha_s \mathbf{x}_s + \alpha_t \mathbf{x}_{t-1}\right)$

$\Longrightarrow$ $\bar{\mathbf{x}}_t - \widetilde{\mathbf{x}}_t = \frac{\alpha_t}{A_t}(\mathbf{x}_t - \mathbf{x}_{t-1})$

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}) \le \frac{D^2}{2\eta} + \boxed{\eta \sum_{t=1}^{T} \frac{\alpha_t^4 L^2}{A_t^2} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 - \frac{1}{4\eta} \sum_{t=1}^{T} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_2^2}$$

ensure that $\left(\frac{\eta \alpha_t^4 L^2}{A_t^2} - \frac{1}{4\eta}\right) \le 0$ with $\alpha_t = t \quad \Longrightarrow \eta \le \frac{1}{4L}$

$\Longrightarrow$ Therefore, by setting $\eta = \frac{1}{4L}$, we have $\mathrm{Reg}_T^{\mathcal{A}} \le 2D^2 L = \mathcal{O}(1)$.

Accelerated Rates by Optimistic OMD

- Combining the weighted online-to-batch conversion and a careful optimism design (for constant regret), we achieve the acceleration.

Lemma 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a convex function with a convex and compact set $\mathcal{X}$. Then, for the following output with weighted average (regardless of how the $\{\mathbf{x}_t\}_{t=1}^T$ are generated): $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^t \alpha_s \mathbf{x}_s$, with $A_t \triangleq \sum_{s=1}^t \alpha_s$ and $\alpha_t > 0$, we have the following online-to-batch conversion:*

$$f(\overline{\mathbf{x}}_T) - f(\mathbf{x}^\star) \leq \frac{\sum_{t=1}^T \langle \alpha_t \nabla f(\overline{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle}{A_T} \triangleq \frac{\mathrm{Reg}_T^{\mathcal{A}}(\mathbf{x}^\star)}{A_T}.$$

$\Rightarrow$ $\mathrm{Reg}_T^{\mathcal{A}} = \mathcal{O}(1), A_T^{-1} = \mathcal{O}(T^{-2})$, which leads to an $\mathcal{O}(T^{-2})$ convergence rate!

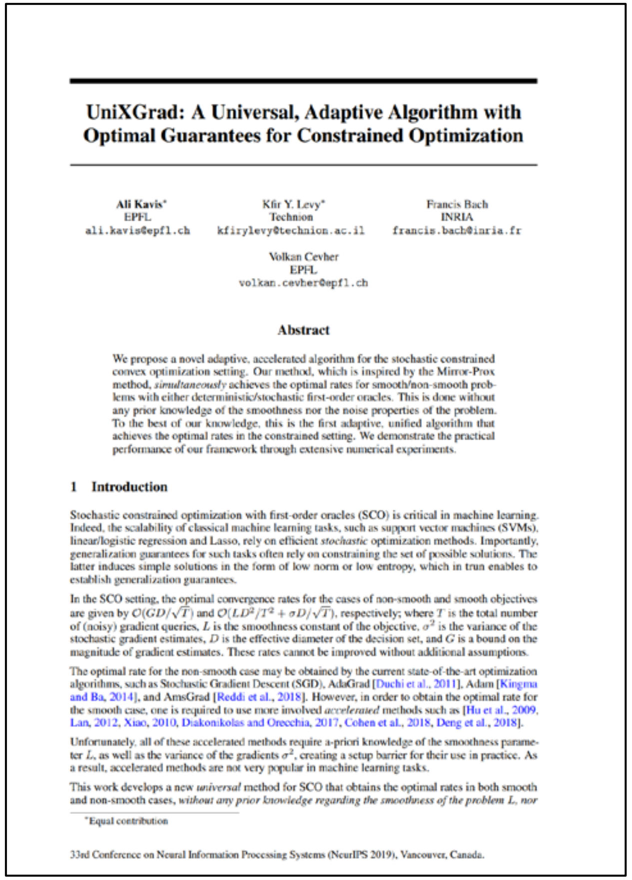
Accelerated Rates by Optimistic OMD

- Combining the weighted online-to-batch conversion and a careful optimism design (for constant regret), we achieve the acceleration.

Algorithm 2 Simple Accelerated Method based on Optimistic OMD

Initialization: Set $\alpha_t = t$, $A_t = \sum_{s=1}^{t} \alpha_s$, $\eta = \frac{1}{4L}$.
for $t = 1, 2, \ldots, T$ **do**
  Submit $\widetilde{\mathbf{x}}_t \triangleq \frac{1}{A_t}\sum_{s=1}^{t-1} \alpha_s \mathbf{x}_s + \alpha_t \mathbf{x}_{t-1}$
  Receive $\nabla f(\widetilde{\mathbf{x}}_t)$, set $M_t = \alpha_t \nabla f(\widetilde{\mathbf{x}}_t)$
  Update $\mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta \langle M_t, \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$
  Submit $\overline{\mathbf{x}}_t = \frac{1}{A_t}\sum_{s=1}^{t} \alpha_s \mathbf{x}_s$
  Receive $\nabla f(\overline{\mathbf{x}}_t)$, set $\nabla f_t(\mathbf{x}_t) = \alpha_t \nabla f(\overline{\mathbf{x}}_t)$
  Update $\widehat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}\rangle + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}_t\|_2^2$
end for

History bits: Optimism for Acceleration



UniXGrad: A Universal, Adaptive Algorithm with Optimal Guarantees for Constrained Optimization

Ali Kavis* EPFL ali.kavis@epfl.ch · Kfir Y. Levy* Technion kfirylevy@technion.ac.il · Francis Bach INRIA francis.bach@inria.fr · Volkan Cevher EPFL volkan.cevher@epfl.ch

Abstract

We propose a novel adaptive, accelerated algorithm for the stochastic constrained convex optimization setting. Our method, which is inspired by the Mirror-Prox method, *simultaneously* achieves the optimal rates for smooth/non-smooth problems with either deterministic/stochastic first-order oracles. This is done without any prior knowledge of the smoothness nor the noise properties of the problem. To the best of our knowledge, this is the first adaptive, unified algorithm that achieves the optimal rates in the constrained setting. We demonstrate the practical performance of our framework through extensive numerical experiments.

1 Introduction

Stochastic constrained optimization with first-order oracles (SCO) is critical in machine learning. Indeed, the scalability of classical machine learning tasks, such as support vector machines (SVMs), linear/logistic regression and Lasso, rely on efficient *stochastic* optimization methods. Importantly, generalization guarantees for such tasks often rely on constraining the set of possible solutions. The latter induces simple solutions in the form of low norm or low entropy, which in trun enables to establish generalization guarantees.

In the SCO setting, the optimal convergence rates for the cases of non-smooth and smooth objectives are given by $\mathcal{O}(GD/\sqrt{T})$ and $\mathcal{O}(LD^2/T^2 + \sigma D/\sqrt{T})$, respectively; where T is the total number of (noisy) gradient queries, L is the smoothness constant of the objective, σ^2 is the variance of the stochastic gradient estimates, D is the effective diameter of the decision set, and G is a bound on the magnitude of gradient estimates. These rates cannot be improved without additional assumptions.

The optimal rate for the non-smooth case may be obtained by the current state-of-the-art optimization algorithms, such as Stochastic Gradient Descent (SGD), AdaGrad [Duchi et al., 2011], Adam [Kingma and Ba, 2014], and AmsGrad [Reddi et al., 2018]. However, in order to obtain the optimal rate for the smooth case, one is required to use more involved *accelerated* methods such as [Hu et al., 2009, Lan, 2012, Xiao, 2010, Diakonikolas and Orecchia, 2017, Cohen et al., 2018, Deng et al., 2018].

Unfortunately, all of these accelerated methods require a-priori knowledge of the smoothness parameter L, as well as the variance of the gradients σ^2, creating a setup barrier for their use in practice. As a result, accelerated methods are not very popular in machine learning tasks.

This work develops a new *universal* method for SCO that obtains the optimal rates in both smooth and non-smooth cases, *without any prior knowledge regarding the smoothness of the problem L, nor*

*Equal contribution

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Proceedings of Machine Learning Research vol 247:1–68, 2024 — 37th Annual Conference on Learning Theory

Accelerated Parameter-Free Stochastic Optimization

Itai Kreisler KREISLER@MAIL.TAU.AC.IL
Maor Ivgi MAOR.IVGI@CS.TAU.AC.IL
Oliver Hinder OHINDER@PITT.EDU
Yair Carmon YCARMON@TAUEX.TAU.AC.IL

Editors: Shipra Agrawal and Aaron Roth

Abstract

We propose a method that achieves near-optimal rates for *smooth* stochastic convex optimization and requires essentially no prior knowledge of problem parameters. This improves on prior work which requires knowing at least the initial distance to optimality d_0. Our method, U-DoG, combines UniXGrad (Kavis et al. [30]) and DoG (Ivgi et al. [27]) with novel iterate stabilization techniques. It requires only loose bounds on d_0 and the noise magnitude, provides high probability guarantees under sub-Gaussian noise, and is also near-optimal in the non-smooth case. Our experiments show consistent, strong performance on convex problems and mixed results on neural network training.

Keywords: Parameter-free, Adaptive, Stochastic convex optimization, Smooth optimization.

1. Introduction

We consider the problem of minimizing a smooth convex function using access to an unbiased stochastic gradient oracle. This is a fundamental problem in machine learning, including many important special cases such as logistic and linear regression. Moreover, the smoothness assumption is crucial for developing one of the most widely used improvements for the classical gradient method: Nesterov acceleration [44].

Nesterov acceleration obtains the optimal rate of convergence for this problem but is strongly reliant on knowing the problem parameters. Specifically, Lan [35], who first demonstrated the theoretical value of Nesterov acceleration on smooth *stochastic* convex functions, requires knowledge of the smoothness parameter β, the distance d_0 from the initial point to the optimum, and a value σ for which the noise is σ-sub-Gaussian. Accelerated adaptive methods [14, 30] do not require knowledge of β and σ, but assume knowledge of d_0. For *non-smooth* stochastic convex optimization, *parameter-free methods* [e.g., 7, 9, 16, 27, 28, 41, 49] require only loose knowledge of problem parameters to obtain near-optimal rates. Finding such parameter-free methods for *smooth* stochastic optimization is a longstanding open problem.

Our contribution. We solve this open problem, designing an accelerated parameter-free method which we call UNIXGRAD-DOG, or U-DOG for short. U-DOG combines the "universal extra-gradient" (UNIXGRAD) framework [30] with the "distance over gradient" (DOG) technique [27]. More specifically, we replace the domain diameter D in the UNIXGRAD step size numerator with the maximum distance from the initial point, similar to the DoG step size numerator. Furthermore, we use this maximum distance to automatically tune the "momentum" parameter α_t of UNIXGRAD.

© 2024 I. Kreisler, M. Ivgi, O. Hinder & Y. Carmon.

UniXGrad: A Universal, Adaptive Algorithm with Optimal Guarantees for Constrained Optimization. NeurIPS 2019.

Accelerated Parameter-Free Stochastic Optimization. COLT 2024.

Summary

Q & A

Thanks!