Lecture 6. Online Optimization II

Advanced Optimization (Fall 2025)

Peng Zhao

zhaop@lamda.nju.edu.cn

Nanjing University

Outline

- Online Exp-concave Optimization

- Prediction with Expert Advice

Part 1. Online Exp-concave Optimization

- Exp-concave functions
- Online Newton Step
- Regret Analysis

Comparison of (Strongly) Convex Problems

Convex

Property: $f_t(\mathbf{x}) \geq f_t(\mathbf{y}) + \nabla f_t(\mathbf{y})^\top(\mathbf{x} - \mathbf{y})$

OGD: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}_t - \frac{1}{\sqrt{t}}\nabla f_t(\mathbf{x}_t)\right]$

$$\textsc{Reg}_T \leq \frac{3}{2}GD\sqrt{T}$$

Strongly Convex

Property: $f_t(\mathbf{y}) \geq f_t(\mathbf{x}) + \nabla f_t(\mathbf{x})^\top(\mathbf{y} - \mathbf{x}) + \frac{\sigma}{2}\|\mathbf{y} - \mathbf{x}\|^2$

OGD: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}_t - \frac{1}{\sigma t}\nabla f_t(\mathbf{x}_t)\right]$

$$\textsc{Reg}_T \leq \frac{G^2}{2\sigma}(1 + \log T)$$

Can we explore broader function classes with a regret rate faster than $\sqrt{T}$?

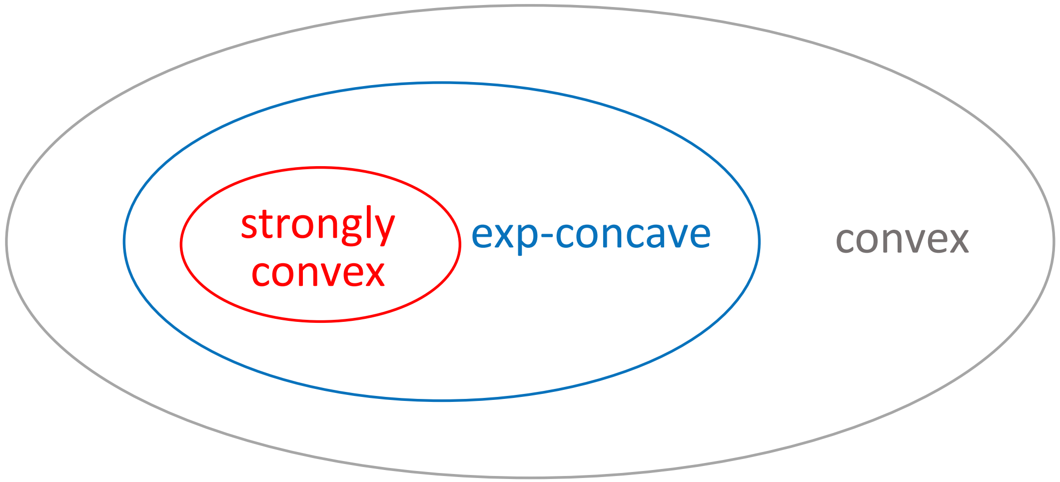
Exponentially-concave Functions

Definition 1 (Exp-concavity). A convex function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is defined to be α-exp-concave over $\mathcal{X} \subseteq \mathbb{R}^d$ if the function g is concave, where $g : \mathcal{X} \mapsto \mathbb{R}$ is defined as

$$g(\mathbf{x}) = e^{-\alpha f(\mathbf{x})}.$$

Directly employ OGD: Regret bound $\mathcal{O}(\sqrt{T})$.

But actually we can get a ***tighter*** bound!



An Example for Exp-concave Learning

- Universal Portfolio Selection 
  - a total of d stocks in the stock market.
  - each round, the player chooses stocks by a distribution $\mathbf{x}_t \in \Delta_d$.
  - the market returns the price ratio $\boldsymbol{\theta}_t$ between iter t and $t+1$,

$$\boldsymbol{\theta}_t(i) = \frac{\text{price of stock}_i \text{ at time } t+1}{\text{price of stock}_i \text{ at time } t}$$

which means that our final wealth W_T will be: $W_T = W_1 \cdot \prod_{t=1}^{T} \boldsymbol{\theta}_t^\top \mathbf{x}_t$

$\Longrightarrow$ Our goal is to maximize our wealth at time T.

An Example for Exp-concave Learning

- Universal Portfolio Selection 

  - we hope to maximize the logarithm of W_T

    $$\log \frac{W_T}{W_1} = \sum_{t=1}^{T} \log \boldsymbol{\theta}_t^\top \mathbf{x}_t$$

  - using OCO framework,

    $$f_t(\mathbf{x}) = \log(\boldsymbol{\theta}_t^\top \mathbf{x})$$

At each round $t = 1, 2, \cdots$

(1) the player first picks a model $\mathbf{x}_t \in \Delta_d$;

(2) and simultaneously environments pick an online function $f_t : \mathcal{X} \to \mathbb{R}$;

(3) the player get a ***gain*** $f_t(\mathbf{x}_t) = \log(\boldsymbol{\theta}_t^\top \mathbf{x}_t)$, observes f_t and updates the model.

- Goal: $\quad \textsc{Reg}_T = \max_{\mathbf{x}^\star \in \Delta_d} \sum_{t=1}^{T} f_t(\mathbf{x}^\star) - \sum_{t=1}^{T} f_t(\mathbf{x}_t)$

online function is exp-concave

Exponential-concave Function

Lemma 1 (Property of Exp-concavity). *Let $f : \mathcal{X} \to \mathbb{R}$ be an α-exp-concave function, and D, G denote the diameter of $\mathcal{X}$ and a bound on the (sub)gradients of f respectively. The following holds for all $\gamma \leq \frac{1}{2}\min\left\{\frac{1}{GD}, \alpha\right\}$ and all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top(\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2}(\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}).$$

Proof. Recall that f is α-exp-concave if and only if $e^{-\alpha f(\mathbf{x})}$ is concave.

As $2\gamma \leq \alpha$, $e^{-2\gamma f(\mathbf{x})} = \left(e^{-\alpha f(\mathbf{x})}\right)^{2\gamma/\alpha}$ is also concave and thus is 2γ-exp-concave.

$$e^{-2\gamma f(\mathbf{x})} - e^{-2\gamma f(\mathbf{y})} \leq \left\langle \mathbf{x} - \mathbf{y}, -2\gamma e^{-2\gamma f(\mathbf{y})} \nabla f(\mathbf{y}) \right\rangle.$$

(concavity)

Exponential-concave Function

Lemma 1 (Property of Exp-concavity). *Let $f : \mathcal{X} \to \mathbb{R}$ be an α-exp-concave function, and D, G denote the diameter of $\mathcal{X}$ and a bound on the (sub)gradients of f respectively. The following holds for all $\gamma \leq \frac{1}{2} \min\left\{\frac{1}{GD}, \alpha\right\}$ and all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:*

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2} (\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}).$$

Proof. Dividing $e^{-2\gamma f(\mathbf{y})}$ at both sides achieves

$$f(\mathbf{y}) - f(\mathbf{x}) \leq \frac{1}{2\gamma} \log\Big(1 + 2\gamma \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{y}) \rangle\Big).$$

Our constructive condition $\gamma \leq \frac{1}{2} \min\left\{\frac{1}{GD}, \alpha\right\}$ ensures $|2\gamma \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{y}) \rangle| \leq 1$,

$$f(\mathbf{y}) - f(\mathbf{x}) \leq \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{y}) \rangle - \frac{\gamma}{2} \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{y}) \rangle^2$$

$\left(\log(1+x) \leq x - \frac{1}{4}x^2\right)$ holds for $(|x| \leq 1)$ □

Exponential-concave Function

Lemma 1 (Property of Exp-concavity). *Let $f : \mathcal{X} \to \mathbb{R}$ be an α-exp-concave function, and D, G denote the diameter of $\mathcal{X}$ and a bound on the (sub)gradients of f respectively. The following holds for all $\gamma \leq \frac{1}{2}\min\left\{\frac{1}{GD}, \alpha\right\}$ and all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$:*

$$
\begin{aligned}
f(\mathbf{x}) &\geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2}(\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) \\
&= f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{y}\|^2_{\nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top}
\end{aligned}
$$

Algorithmic intuition:

- For convex loss, we use 2-norm to encode the structure of the space.
- Can we exploit ***local structures*** of exp-concave loss to improve the regret?

A Comparison of Different Curvatures

- Convex

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y})$$

- Strongly Convex

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\sigma}{2} \|\mathbf{x} - \mathbf{y}\|_2^2$$

- Exponentially Concave

$$\begin{aligned} f(\mathbf{x}) &\geq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2} (\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) \\ &= f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2} \|\mathbf{x} - \mathbf{y}\|^2_{\nabla f(\mathbf{y}) \nabla f(\mathbf{y})^\top} \end{aligned}$$

Intuition

- Convex $$f_t(\mathbf{x}) \geq f_t(\mathbf{x}_t) + \nabla f_t(\mathbf{x}_t)^\top (\mathbf{x} - \mathbf{x}_t)$$

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} \left[\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)\right] \qquad \text{OGD with } \eta_t = \mathcal{O}(1/\sqrt{t})$$

- Strongly convex $$f_t(\mathbf{x}) \geq f_t(\mathbf{x}_t) + \nabla f_t(\mathbf{x}_t)^\top (\mathbf{x} - \mathbf{x}_t) + \frac{\sigma}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2$$

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} \left[\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)\right] \qquad \text{OGD with } \eta_t = \mathcal{O}(1/t)$$

- Exp-concave $$f_t(\mathbf{x}) \geq f_t(\mathbf{x}_t) + \nabla f_t(\mathbf{x}_t)^\top (\mathbf{x} - \mathbf{x}_t) + \frac{\gamma}{2} \|\mathbf{x} - \mathbf{x}_t\|^2_{\nabla f_t(\mathbf{x}_t) \nabla f_t(\mathbf{x}_t)^\top}$$

$\Longrightarrow$ We may still GD update, but the step size should be "***data-dependent***".

Intuitively, step size should be stretched hetrogeneously in different directions, being smaller when $\nabla f_t(\mathbf{x}_t) \nabla f_t(\mathbf{x}_t)^\top$ is "larger".

Online Newton Step

Online Newton Step

Input: parameters $\gamma, \varepsilon > 0$, matrix $A_0 = \varepsilon I_d$

At each round $t = 1, 2, \cdots$

(1) the player first picks a model $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$;

(2) and simultaneously environments pick an ***exp-concave loss function*** $f_t : \mathcal{X} \to \mathbb{R}$;

(3) the player suffers loss $f_t(\mathbf{x}_t)$, observes the information (loss) f_t and update:

$$\text{Update } A_t = A_{t-1} + \nabla f_t(\mathbf{x}_t) \nabla f_t(\mathbf{x}_t)^\top$$

$$\text{Update } \mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} \left\| \mathbf{x} - \left(\mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \nabla f_t(\mathbf{x}_t) \right) \right\|_{A_t}^2$$

essentially Gradient Descent *stretched* by the matrix A_t

ONS: In a View of Proximal Gradient

Convex Problem

Property: $f_t(\mathbf{x}) \geq f_t(\mathbf{y}) + \nabla f_t(\mathbf{y})^\top(\mathbf{x}-\mathbf{y})$

OGD: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}_t - \frac{1}{\sqrt{t}}\nabla f_t(\mathbf{x}_t)\right]$

Proximal type update:

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

Exp-concave Problem

Property: $f_t(\mathbf{x}) \geq f_t(\mathbf{y}) + \nabla f_t(\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) + \frac{\gamma}{2}\|\mathbf{x}-\mathbf{y}\|^2_{\nabla f_t(\mathbf{y})\nabla f_t(\mathbf{y})^\top}$

ONS: $A_t = A_{t-1} + \nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top$

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}^{A_t}\left[\mathbf{x}_t - \frac{1}{\gamma}A_t^{-1}\nabla f_t(\mathbf{x}_t)\right]$$

Proximal type update:

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

ONS: In a View of Proximal Gradient

Proof.

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}^{A_t}\left[\mathbf{x}_t - \frac{A_t^{-1}}{\gamma}\mathbf{g}_t\right] \quad (\mathbf{g}_t \triangleq \nabla f_t(\mathbf{x}_t))$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}} \left(\mathbf{x} - \mathbf{x}_t + \frac{A_t^{-1}}{\gamma}\mathbf{g}_t\right)^\top A_t \left(\mathbf{x} - \mathbf{x}_t + \frac{A_t^{-1}}{\gamma}\mathbf{g}_t\right)$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}} \left(\mathbf{x} - \mathbf{x}_t + \frac{A_t^{-1}}{\gamma}\mathbf{g}_t\right)^\top \left(A_t\mathbf{x} - A_t\mathbf{x}_t + \frac{\mathbf{g}_t}{\gamma}\right)$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}} (\mathbf{x} - \mathbf{x}_t)^\top A_t (\mathbf{x} - \mathbf{x}_t) + \cancel{(A^{-1})^\top \mathbf{g}_t^\top \mathbf{g}_t} + 2\frac{\mathbf{g}_t^\top(\mathbf{x} - \cancel{\mathbf{x}_t})}{\gamma} \quad \text{(constant)}$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \mathbf{g}_t\rangle + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{x}_t\|_{A_t}^2$$

Exp-concave Problem

Property: $f_t(\mathbf{x}) \geq f_t(\mathbf{y}) + \nabla f_t(\mathbf{y})^\top(\mathbf{x} - \mathbf{y}) + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{y}\|^2_{\nabla f_t(\mathbf{y})\nabla f_t(\mathbf{y})^\top}$

ONS: $A_t = A_{t-1} + \nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top$

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}^{A_t}\left[\mathbf{x}_t - \frac{1}{\gamma}A_t^{-1}\nabla f_t(\mathbf{x}_t)\right]$$

Proximal type update:

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

ONS for Exp-concave Function

Theorem 1. *Under Assumption 1 (G-Lipschitz) and Assumption 2 (D-bounded domain), for α-exp-concave online functions, the ONS algorithm with parameters $\gamma = \frac{1}{2}\min\left\{\frac{1}{GD}, \alpha\right\}$ and $\varepsilon = \frac{1}{\gamma^2 D^2}$ (the initial matrix is $A_0 = \varepsilon I_d$) guarantees*

$$\textsc{Reg}_T \leq \mathcal{O}\left((\frac{1}{\alpha} + GD) d \log T \right) = \mathcal{O}(d \log T),$$

where d is the dimension of the feasible domain $\mathcal{X} \subseteq \mathbb{R}^d$.

- Acheving $\mathcal{O}(\log T)$ rate but without strong convexity; only exp-concavity is required.
- Pay attention to the d dimension dependency, and think of why?

Proof

- $A_t = A_{t-1} + \nabla f_t(\mathbf{x}_t) \nabla f_t(\mathbf{x}_t)^\top$
- $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\| \mathbf{x} - (\mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \mathbf{g}_t) \right\|_{A_t}^2$

Extending *the first GD lemma* to ***exp-concave case***:

Proof.

We use norm induced by A_t instead of 2-norm.

$$\|\mathbf{x}_{t+1} - \mathbf{u}\|_{A_t}^2 = \left\| \Pi_{\mathcal{X}}^{A_t} \left[\mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \nabla f_t(\mathbf{x}_t) \right] - \mathbf{u} \right\|_{A_t}^2$$

$\left(\Pi_{\mathcal{X}}^{A}[\mathbf{y}] \triangleq \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|_A^2 \right)$

$$\leq \left\| \mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \nabla f_t(\mathbf{x}_t) - \mathbf{u} \right\|_{A_t}^2$$

(A_t is semidefinite matrix)
(Pythagoras theorem)

$$= \left(\mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \nabla f_t(\mathbf{x}_t) - \mathbf{u} \right)^\top A_t \left(\mathbf{x}_t - \frac{1}{\gamma} A_t^{-1} \nabla f_t(\mathbf{x}_t) - \mathbf{u} \right)$$

(definition of $\|\cdot\|_{A_t}^2$)

$$= \left(\mathbf{x}_t - \mathbf{u} - \frac{1}{\gamma} A_t^{-1} \nabla f_t(\mathbf{x}_t) \right)^\top \left(A_t(\mathbf{x}_t - \mathbf{u}) - \frac{1}{\gamma} \nabla f_t(\mathbf{x}_t) \right)$$

Proof

- $A_t = A_{t-1} + \nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top$
- $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\| \mathbf{x} - (\mathbf{x}_t - \frac{1}{\gamma}A_t^{-1}\mathbf{g}_t) \right\|_{A_t}^2$

Extending ***the first GD lemma*** to ***exp-concave case***:

Proof.

$$\|\mathbf{x}_{t+1} - \mathbf{u}\|_{A_t}^2 = \left(\mathbf{x}_t - \mathbf{u} - \frac{1}{\gamma}A_t^{-1}\nabla f_t(\mathbf{x}_t)\right)^\top \left(A_t(\mathbf{x}_t - \mathbf{u}) - \frac{1}{\gamma}\nabla f_t(\mathbf{x}_t)\right)$$

$$= (\mathbf{x}_t - \mathbf{u})^\top A_t (\mathbf{x}_t - \mathbf{u}) - \frac{2}{\gamma}\nabla f_t(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{u}) + \frac{1}{\gamma^2}\nabla f_t(\mathbf{x}_t)^\top A_t^{-1}\nabla f_t(\mathbf{x}_t)$$

$$\leq \|\mathbf{x}_t - \mathbf{u}\|_{A_t}^2 - \frac{2}{\gamma}\left(f_t(\mathbf{x}_t) - f_t(\mathbf{u})\right) + \frac{1}{\gamma^2}\|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2$$

$$- (\mathbf{x}_t - \mathbf{u})^\top\nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{u})$$

(Exp-concave: $f_t(\mathbf{x}) \geq f_t(\mathbf{y}) + \nabla f_t(\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) + \frac{\gamma}{2}(\mathbf{x}-\mathbf{y})^\top\nabla f_t(\mathbf{y})\nabla f_t(\mathbf{y})^\top(\mathbf{x}-\mathbf{y})$)

Proof

Proof.

$$\begin{aligned}&\|\mathbf{x}_{t+1} - \mathbf{u}\|^2_{A_t} \\ &\le \|\mathbf{x}_t - \mathbf{u}\|^2_{A_t} - \frac{2}{\gamma}\big(f_t(\mathbf{x}_t) - f_t(\mathbf{u})\big) - \|\mathbf{x}_t - \mathbf{u}\|^2_{\nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top} + \frac{1}{\gamma^2}\|\nabla f_t(\mathbf{x}_t)\|^2_{A_t^{-1}}\end{aligned}$$

$$\Longrightarrow f_t(\mathbf{x}_t) - f_t(\mathbf{u}) \le \frac{\gamma}{2}\|\mathbf{x}_t - \mathbf{u}\|^2_{A_t} - \frac{\gamma}{2}\|\mathbf{x}_{t+1} - \mathbf{u}\|^2_{A_t} - \frac{\gamma}{2}\|\mathbf{x}_t - \mathbf{u}\|^2_{\nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top} + \frac{1}{2\gamma}\|\nabla f_t(\mathbf{x}_t)\|^2_{A_t^{-1}}$$

(rearranging)

Summing from $t = 1$ to T, by telescoping:

$$\begin{aligned}\sum_{t=1}^T \big(f_t(\mathbf{x}_t) - f_t(\mathbf{u})\big) &\le \frac{\gamma}{2}\sum_{t=1}^T \Big(\|\mathbf{x}_t - \mathbf{u}\|^2_{A_t} - \|\mathbf{x}_t - \mathbf{u}\|^2_{A_{t-1}}\Big) + \frac{1}{2\gamma}\sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t)\|^2_{A_t^{-1}} \\ &\quad + \frac{\gamma}{2}\|\mathbf{x}_1 - \mathbf{u}\|^2_{A_0} - \frac{\gamma}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}\|^2_{\nabla f(\mathbf{x}_t)\nabla f(\mathbf{x}_t)^\top} \\ &= \frac{\gamma}{2}\|\mathbf{x}_1 - \mathbf{u}\|^2_{A_0} + \frac{1}{2\gamma}\sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t)\|^2_{A_t^{-1}}\end{aligned}$$

cancellation

$(A_t = A_{t-1} + \nabla f_t(\mathbf{x}_t)\nabla f_t(\mathbf{x}_t)^\top)$

Proof

Proof. $$\sum_{t=1}^{T}\left(f_t(\mathbf{x}_t)-f_t(\mathbf{u})\right) \leq \frac{\gamma}{2}\left\|\mathbf{x}_1-\mathbf{u}\right\|_{A_0}^2+\frac{1}{2\gamma}\sum_{t=1}^{T}\left\|\nabla f_t(\mathbf{x}_t)\right\|_{A_t^{-1}}^2$$

By the definition that $A_0 \triangleq \varepsilon I_d$, $\varepsilon = \frac{1}{\gamma^2 D^2}$ and the diameter $\|\mathbf{x}_1-\mathbf{u}\|_2^2 \leq D^2$:

$$\begin{aligned}\sum_{t=1}^{T}\left(f_t(\mathbf{x}_t)-f_t(\mathbf{u})\right) &\leq \frac{\gamma}{2}\left(\mathbf{x}_1-\mathbf{u}\right)^{\top} A_0\left(\mathbf{x}_1-\mathbf{u}\right)+\frac{1}{2\gamma}\sum_{t=1}^{T}\left\|\nabla f_t(\mathbf{x}_t)\right\|_{A_t^{-1}}^2 \\ &\leq \frac{1}{2\gamma}+\frac{1}{2\gamma}\sum_{t=1}^{T}\left\|\nabla f_t(\mathbf{x}_t)\right\|_{A_t^{-1}}^2 .\end{aligned}$$

Next, we bound the term $\sum_{t=1}^{T}\left\|\nabla f_t(\mathbf{x}_t)\right\|_{A_t^{-1}}^2$.

Proof

Proof. Next, we bound the term $\sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2$.

Lemma 2 (Elliptical Potential Lemma). *For any sequence* $\{X_1, \ldots, X_T\} \in \mathbb{R}^{d \times T}$, *suppose* $U_0 = \lambda I, U_t = U_{t-1} + X_t X_t^\top$, *and* $\|X_t\|_2 \leq L$, *then*

$$\sum_{t=1}^{T} \|X_t\|_{U_t^{-1}}^2 \leq d \log\left(1 + \frac{L^2 T}{\lambda d}\right)$$

Proof.

$$U_{t-1} = U_t - X_t X_t^\top = U_t^{\frac{1}{2}} \left(I - U_t^{-\frac{1}{2}} X_t X_t^\top U_t^{-\frac{1}{2}}\right) U_t^{\frac{1}{2}} \qquad \text{(definition of } U_t\text{)}$$

$$\det(U_{t-1}) = \det(U_t) \det\left(I - U_t^{-\frac{1}{2}} X_t X_t^\top U_t^{-\frac{1}{2}}\right) \qquad \text{(determinant on both side)}$$

Proof

Proof. Next, we bound the term $\sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2$.

Lemma 3. *For any* $\mathbf{v} \in \mathbb{R}^d$*, we have*

$$\det\left(I - \mathbf{v}\mathbf{v}^\top\right) = 1 - \|\mathbf{v}\|_2^2$$

Proof.

(i) $\left(I - \mathbf{v}\mathbf{v}^\top\right)\mathbf{v} = \left(1 - \|\mathbf{v}\|_2^2\right)\mathbf{v}$, therefore, $\mathbf{v}$ is its eigenvector with $\left(1 - \|\mathbf{v}\|_2^2\right)$ as eigenvalue;

(ii) $\left(I - \mathbf{v}\mathbf{v}^\top\right)\mathbf{v}^\perp = \mathbf{v}^\perp$, therefore, $\mathbf{v}^\perp \perp \mathbf{v}$ is its eigenvector with 1 as the eigenvalue. $\square$

Proof

Proof. Next, we bound the term $\sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t)\|^2_{A_t^{-1}}$.

Lemma 2 (Elliptical Potential Lemma). *For any sequence* $\{X_1, \dots, X_T\} \in \mathbb{R}^{d\times T}$*, suppose* $U_0 = \lambda I, U_t = U_{t-1} + X_t X_t^\top$*, and* $\|X_t\|_2 \le L$*, then*

$$\sum_{t=1}^T \|X_t\|^2_{U_t^{-1}} \le d \log\left(1 + \frac{L^2 T}{\lambda d}\right)$$

Proof. $\det(U_{t-1}) = \det(U_t)\det\left(I - U_t^{-\frac{1}{2}} X_t X_t^\top U_t^{-\frac{1}{2}}\right) = \det(U_t)\left(1 - \left\|U_t^{-\frac{1}{2}} X_t\right\|_2^2\right)$

(by Lemma 3)

$$\Longrightarrow \|X_t\|^2_{U_t^{-1}} = \left\|U_t^{-\frac{1}{2}} X_t\right\|_2^2 = 1 - \frac{\det(U_{t-1})}{\det(U_t)}$$

(rearranging, U is a symmetric matrix)

Proof

Proof. Next, we bound the term $\sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2$.

Lemma 2 **(Elliptical Potential Lemma).** *For any sequence* $\{X_1, \ldots, X_T\} \in \mathbb{R}^{d \times T}$*, suppose* $U_0 = \lambda I, U_t = U_{t-1} + X_t X_t^\top$*, and* $\|X_t\|_2 \leq L$*, then*

$$\sum_{t=1}^{T} \|X_t\|_{U_t^{-1}}^2 \leq d \log\left(1 + \frac{L^2 T}{\lambda d}\right)$$

Proof.

$$\Longrightarrow \quad \sum_{t=1}^{T} X_t^\top U_t^{-1} X_t = \sum_{t=1}^{T}\left(1 - \frac{\det(U_{t-1})}{\det(U_t)}\right) \leq \sum_{t=1}^{T} \log \frac{\det(U_t)}{\det(U_{t-1})} \qquad (\forall x > 0, 1 - x \leq -\log x)$$

$$= \log \frac{\det(U_T)}{\det(U_0)} = d \log\left(1 + \frac{L^2 T}{\lambda d}\right) \quad \square$$

$$\mathrm{Tr}(U_T) \leq \mathrm{Tr}(U_0) + L^2 T = \lambda d + L^2 T$$

$$\Longrightarrow \quad \det(U_T) \leq (\lambda + L^2 T / d)^d$$

Proof

Proof. Next, we bound the term $\sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2$.

Lemma 2 **(Elliptical Potential Lemma).** *For any sequence* $\{X_1, \ldots, X_T\} \in \mathbb{R}^{d \times T}$*, suppose* $U_0 = \lambda I, U_t = U_{t-1} + X_t X_t^\top$*, and* $\|X_t\|_2 \leq L$*, then*

$$\sum_{t=1}^{T} \|X_t\|_{U_t^{-1}}^2 \leq d \log\left(1 + \frac{L^2 T}{\lambda d}\right)$$

Therefore, by Lemma 2, we have

$$\sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2 \leq d \log\left(1 + \frac{G^2 T}{\varepsilon d}\right).$$

Proof

Proof. To conclude,

$$\sum_{t=1}^{T}(f_t(\mathbf{x}_t) - f_t(\mathbf{u})) \leq \underbrace{\frac{\gamma}{2}\|\mathbf{x}_1 - \mathbf{u}\|_{A_0}^2}_{\leq \frac{1}{2\gamma}} + \underbrace{\frac{1}{2\gamma}\sum_{t=1}^{T}\|\nabla f_t(\mathbf{x}_t)\|_{A_t^{-1}}^2}_{\leq \frac{d}{2\gamma}\log\left(1+\frac{G^2T}{\varepsilon d}\right)}.$$

(bounded domain) (elliptical potential lemma)

Recall that $\gamma = \frac{1}{2}\min\left\{\frac{1}{GD}, \alpha\right\}$ and $\varepsilon = \frac{1}{\gamma^2 D^2}$,

$$\textsc{Reg}_T \leq \mathcal{O}\left(\left(\frac{1}{\alpha} + GD\right) d \log T\right).$$

□

Comparison

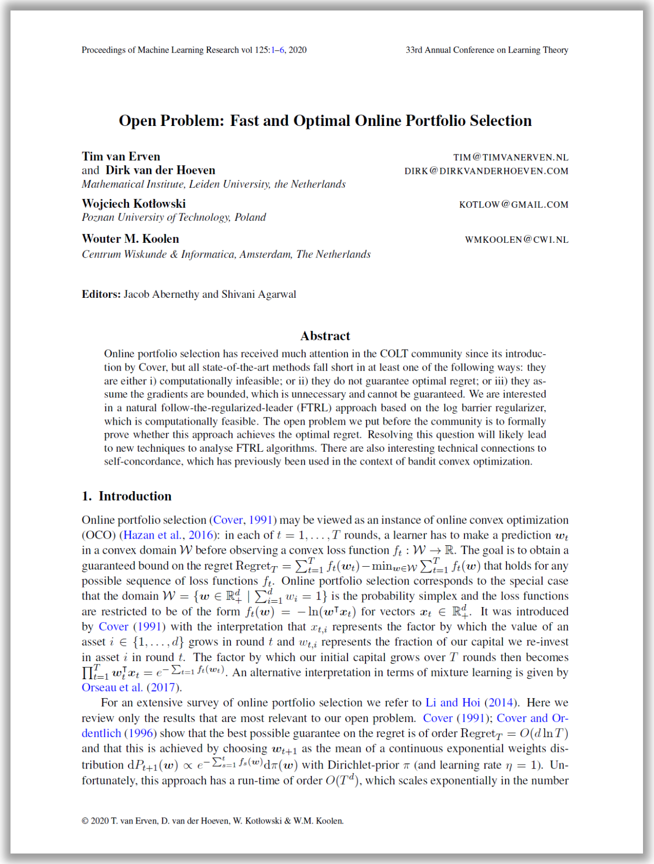
	Algorithm	Upper Bound	Lower Bound
Convex	OGD	$\mathcal{O}(\sqrt{T})$	$\Omega(\sqrt{T})$
σ-Strongly Convex	OGD	$\mathcal{O}\left(\frac{\log T}{\sigma}\right)$	$\Omega\left(\frac{\log T}{\sigma}\right)$
α-Exp-concave	ONS	$\mathcal{O}\left(\frac{d\log T}{\alpha}\right)$	$\Omega\left(\frac{d\log T}{\alpha}\right)$

Ordentlich and Cover. The Cost of Achieving the Best Portfolio in Hindsight. Operation Research,1998

Back to Exp-concave Learning

- Universal Portfolio Selection

Algorithm	Regret	Runtime (per round)
Universal Portfolios	$d\log(T)$	d^4T^{14}
Online Gradient Descent	$G_2\sqrt{T}$	d
Exponentiated Gradient	$G_\infty\sqrt{T\log(d)}$	d
Online Newton Step (ONS)	$G_\infty d\log(T)$	d^2+generalized projection on Δ_d
Soft-Bayes	$\sqrt{dT\log(d)}$	d
Ada-BARRONS	$d^2\log^4(T)$	$d^{2.5}T$
BISONS	$d^2\log^2(T)$	$\mathrm{poly}(d)$
AdaMix+DONS	$d^2\log^5(T)$	d^3
VB-FTRL	$d\log(T)$	d^2T


Proceedings of Machine Learning Research vol 125:1–6, 2020 33rd Annual Conference on Learning Theory

Open Problem: Fast and Optimal Online Portfolio Selection

Tim van Erven TIM@TIMVANERVEN.NL
and **Dirk van der Hoeven** DIRK@DIRKVANDERHOEVEN.COM
Mathematical Institute, Leiden University, the Netherlands
Wojciech Kotłowski KOTLOW@GMAIL.COM
Poznan University of Technology, Poland
Wouter M. Koolen WMKOOLEN@CWI.NL
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Editors: Jacob Abernethy and Shivani Agarwal

Abstract

Online portfolio selection has received much attention in the COLT community since its introduction by Cover, but all state-of-the-art methods fall short in at least one of the following ways: they are either i) computationally infeasible; or ii) they do not guarantee optimal regret; or iii) they assume the gradients are bounded, which is unnecessary and cannot be guaranteed. We are interested in a natural follow-the-regularized-leader (FTRL) approach based on the log barrier regularizer, which is computationally feasible. The open problem we put before the community is to formally prove whether this approach achieves the optimal regret. Resolving this question will likely lead to new techniques to analyse FTRL algorithms. There are also interesting technical connections to self-concordance, which has previously been used in the context of bandit convex optimization.

1. Introduction

Online portfolio selection (Cover, 1991) may be viewed as an instance of online convex optimization (OCO) (Hazan et al., 2016): in each of $t = 1, \dots, T$ rounds, a learner has to make a prediction $\boldsymbol{w}_t$ in a convex domain $\mathcal{W}$ before observing a convex loss function $f_t : \mathcal{W} \to \mathbb{R}$. The goal is to obtain a guaranteed bound on the regret $\mathrm{Regret}_T = \sum_{t=1}^T f_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w}\in\mathcal{W}} \sum_{t=1}^T f_t(\boldsymbol{w})$ that holds for any possible sequence of loss functions f_t. Online portfolio selection corresponds to the special case that the domain $\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}_+^d \mid \sum_{i=1}^d w_i = 1\}$ is the probability simplex and the loss functions are restricted to be of the form $f_t(\boldsymbol{w}) = -\ln(\boldsymbol{w}^\intercal \boldsymbol{x}_t)$ for vectors $\boldsymbol{x}_t \in \mathbb{R}_+^d$. It was introduced by Cover (1991) with the interpretation that $x_{t,i}$ represents the factor by which the value of an asset $i \in \{1, \dots, d\}$ grows in round t and $w_{t,i}$ represents the fraction of our capital we re-invest in asset i in round t. The factor by which our initial capital grows over T rounds then becomes $\prod_{t=1}^T \boldsymbol{w}_t^\intercal \boldsymbol{x}_t = e^{-\sum_{t=1} f_t(\boldsymbol{w}_t)}$. An alternative interpretation in terms of mixture learning is given by Orseau et al. (2017).

For an extensive survey of online portfolio selection we refer to Li and Hoi (2014). Here we review only the results that are most relevant to our open problem. Cover (1991); Cover and Ordentlich (1996) show that the best possible guarantee on the regret is of order $\mathrm{Regret}_T = O(d \ln T)$ and that this is achieved by choosing $\boldsymbol{w}_{t+1}$ as the mean of a continuous exponential weights distribution $\mathrm{d}P_{t+1}(\boldsymbol{w}) \propto e^{-\sum_{s=1}^t f_s(\boldsymbol{w})} \mathrm{d}\pi(\boldsymbol{w})$ with Dirichlet-prior π (and learning rate $\eta = 1$). Unfortunately, this approach has a run-time of order $O(T^d)$, which scales exponentially in the number

© 2020 T. van Erven, D. van der Hoeven, W. Kotłowski & W.M. Koolen.

[COLT'20 Open Problem]

⇨ still an important open problem: efficiency and optimality

Part 2. Prediction with Expert Advice

- Problem Setup

- Algorithms

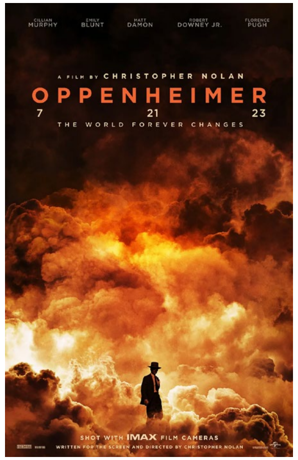
- Regret Analysis

Motivation

- Consider that we are making predictions based on external experts.

A Chinese Odyssey Part Two - Cinderella

9.2/10 87% 7.8/10



Oppenheimer

8.8/10 93% 8.5/10

Titanic

9.5/10 88% 7.9/10

Prediction with Expert Advice

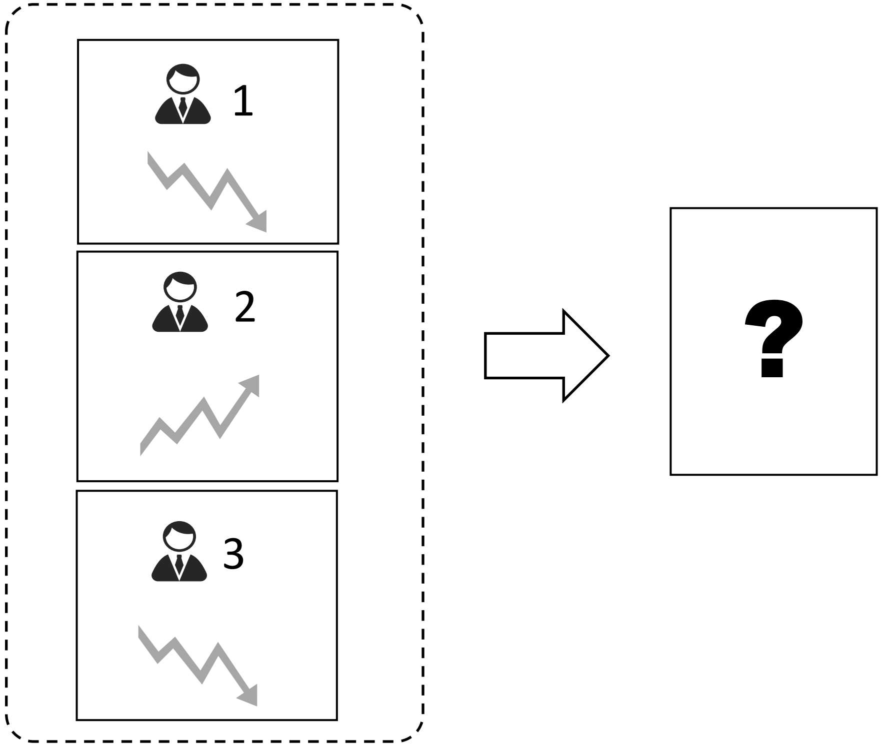
- Another Example: Universal Portfolio Selection

- Universal Portfolio Selection
  - a total of d stocks in the stock market.
  - each round, the player chooses stocks by a distribution $\mathbf{x}_t \in \Delta_d$.
  - the market returns the price ratio $\boldsymbol{\theta}_t$ between iter t and $t+1$,

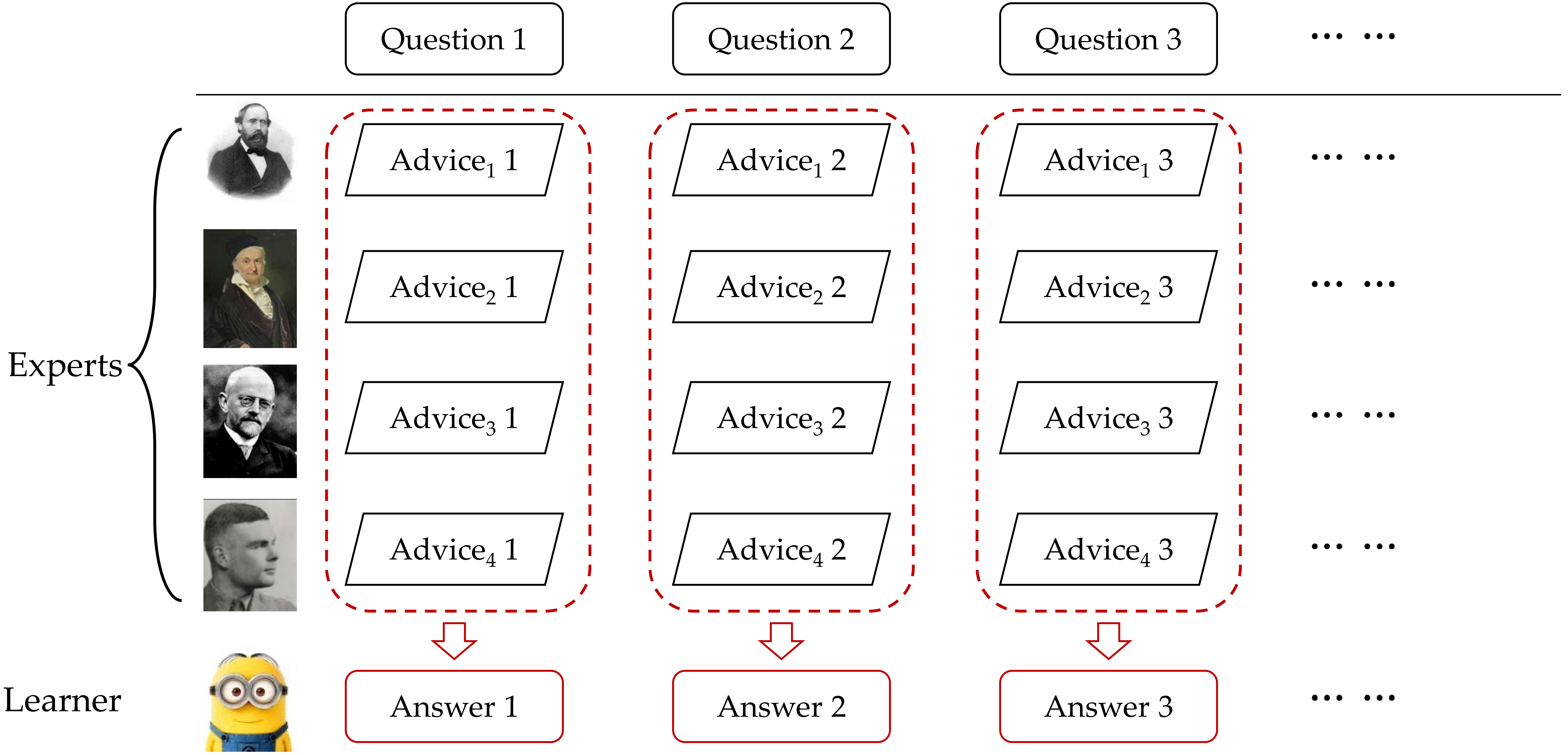
$$\boldsymbol{\theta}_t(i) = \frac{\text{price of stock}_i \text{ at time } t+1}{\text{price of stock}_i \text{ at time } t}$$

which means that our final wealth W_T will be: $W_T = W_1 \cdot \prod_{t=1}^{T} \boldsymbol{\theta}_t^\top \mathbf{x}_t$

⇨ Our goal is to maximize our wealth at time T.



PEA Problem Setup



PEA: Formulization

- The online learner (player) aims to make the prediction based by combining N experts' advice.

At each round $t = 1, 2, \cdots$

(1) the player first picks a weight $\boldsymbol{p}_t$ from a simplex Δ_N;

(2) and simultaneously environments pick a loss vector $\boldsymbol{\ell}_t \in \mathbb{R}^N$;

(3) the player suffers loss $f_t(\boldsymbol{p}_t) \triangleq \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, observes $\boldsymbol{\ell}_t$ and updates the model.

The feasible domain is the $(N-1)$-dim simplex $\Delta_N = \{\boldsymbol{p} \in \mathbb{R}^N \mid p_i \geq 0, \sum_{i=1}^N p_i = 1\}$.

We typically assume that $0 \leq \ell_{t,i} \leq 1$ holds for all $t \in [T]$ and $i \in [N]$.

PEA: Formulization

- The online learner (player) aims to make the prediction based by combining N experts' advice.

At each round $t = 1, 2, \cdots$

(1) the player first picks a weight $\boldsymbol{p}_t$ from a simplex Δ_N;

(2) and simultaneously environments pick a loss vector $\boldsymbol{\ell}_t \in \mathbb{R}^N$;

(3) the player suffers loss $f_t(\boldsymbol{p}_t) \triangleq \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, observes $\boldsymbol{\ell}_t$ and updates the model.

- The goal is to minimize the regret with respect to the ***best expert***:

$$\textsc{Reg}_T \triangleq \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{\boldsymbol{p} \in \Delta_N} \sum_{t=1}^{T} \langle \boldsymbol{p}, \boldsymbol{\ell}_t \rangle = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{i \in [N]} \sum_{t=1}^{T} \ell_{t,i}$$

A Natural Solution

- **Follow the Leader (FTL)**

Select the expert that ***performs best so far***, specifically,

$$\boldsymbol{p}_t^{\mathrm{FTL}} = \mathop{\arg\min}_{\boldsymbol{p}\in\Delta_N} \langle \boldsymbol{p}, \boldsymbol{L}_{t-1} \rangle = \mathop{\mathrm{argmin}}_{i\in[N]} L_{t-1,i}$$

where $\boldsymbol{L}_{t-1} \in \mathbb{R}^N$ is the cumulative loss vector with $L_{t-1,i} \triangleq \sum_{s=1}^{t-1} \ell_{s,i}$.

$\ell_{1,1} = 0.49$ ⇨ $\ell_{2,1} = 1$ ⇨ $\ell_{3,1} = 0$ ……

$\ell_{1,2} = 0.51$ ⇨ $\ell_{2,2} = 0$ ⇨ $\ell_{3,2} = 1$ ……

$$\mathrm{Reg}_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - \min_{i\in[N]} \sum_{t=1}^{T} \ell_{t,i}$$

$$= T - \frac{T}{2} = \mathcal{O}(T)$$

FTL achieves ***linear regret*** in the worst case!

A Natural Solution

- **Follow the Leader (FTL)**

Select the expert that *performs best so far*, specifically,

$$\boldsymbol{p}_t^{\mathrm{FTL}} = \mathop{\arg\min}_{\boldsymbol{p} \in \Delta_N} \langle \boldsymbol{p}, \boldsymbol{L}_{t-1} \rangle = \mathop{\mathrm{argmin}}_{i \in [N]} L_{t-1,i}$$

where $\boldsymbol{L}_{t-1} \in \mathbb{R}^N$ is the cumulative loss vector with $L_{t-1,i} \triangleq \sum_{s=1}^{t-1} \ell_{s,i}$.

⇨ Pitfall: online decision is made ***blindly*** based on the historical performance!

⇨ Replacing the "max" operation in FTL by "***softmax***".

Hedge: Algorithm

- Hedge: replacing the "*max*" operation in FTL by "***softmax***".

At each round $t = 1, 2, \cdots$

(1) compute $\boldsymbol{p}_t \in \Delta_N$ such that $\boldsymbol{p}_{t,i} \propto \exp\left(-\eta L_{t-1,i}\right)$ for $i \in [N]$

(2) the player submits $\boldsymbol{p}_t$, suffers loss $\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle$, and observes loss $\boldsymbol{\ell}_t \in \mathbb{R}^N$

(3) update $\boldsymbol{L}_t = \boldsymbol{L}_{t-1} + \boldsymbol{\ell}_t$

FTL update

$$\boldsymbol{p}_t^{\mathrm{FTL}} = \operatorname*{arg\,max}_{\boldsymbol{p} \in \Delta_N} \langle \boldsymbol{p}, -\boldsymbol{L}_{t-1} \rangle$$

Hedge update

$$p_{t,i} \propto \exp\left(-\eta L_{t-1,i}\right), \forall i \in [N]$$

Lazy and Greedy Updates

- Hedge algorithm

lazy update

$$p_{t+1,i} \propto \exp\left(-\eta L_{t,i}\right), \forall i \in [N] \qquad L_{t,i} = \sum_{s=1}^{t} \ell_{s,i}, \ \forall i \in [N]$$

- Another equivalent update (when the learning rate η is ***fixed***)

greedy update

$$p_{t+1,i} \propto p_{t,i} \exp\left(-\eta \ell_{t,i}\right), \forall i \in [N]$$

where we set the uniform initialization as $p_{0,i} = 1/N, \forall i \in [N]$.

⇨ But the two updates can be ***significantly different when learning rate is changing***.

Hedge: Regret Bound

Theorem 2. *Suppose that* $\forall t \in [T]$ *and* $i \in [N], 0 \le \boldsymbol{\ell}_{t,i} \le 1$, *then Hedge with learning rate* η *guarantees*

$$\textsc{Reg}_T \le \frac{\ln N}{\eta} + \eta T = \mathcal{O}\big(\sqrt{T \log N}\big),$$

where the last equality is by setting η *optimally as* $\sqrt{(\ln N)/T}$.

Proof. We present a *potential-based* proof here, where the potential is defined as

$$\Phi_t \triangleq \frac{1}{\eta} \ln \left(\sum_{i=1}^{N} \exp\left(-\eta L_{t,i}\right) \right).$$

Proof of Hedge Regret Bound

Proof.

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta}\ln\left(\frac{\sum_{i=1}^N \exp\left(-\eta L_{t,i}\right)}{\sum_{i=1}^N \exp\left(-\eta L_{t-1,i}\right)}\right) \qquad \Phi_t \triangleq \frac{1}{\eta}\ln\left(\sum_{i=1}^N \exp\left(-\eta L_{t,i}\right)\right)$$

$$= \frac{1}{\eta}\ln\left(\sum_{i=1}^N \left(\frac{\exp\left(-\eta L_{t-1,i}\right)}{\sum_{i=1}^N \exp\left(-\eta L_{t-1,i}\right)}\exp(-\eta \ell_{t,i})\right)\right)$$

$$= \frac{1}{\eta}\ln\left(\sum_{i=1}^N p_{t,i}\exp\left(-\eta \ell_{t,i}\right)\right) \qquad \text{(update step of } \boldsymbol{p}_t\text{)}$$

$$\leq \frac{1}{\eta}\ln\left(\sum_{i=1}^N p_{t,i}\left(1 - \eta \ell_{t,i} + \eta^2 \ell_{t,i}^2\right)\right) \qquad (\forall x \geq 0, e^{-x} \leq 1 - x + x^2)$$

$$= \frac{1}{\eta}\ln\left(1 - \eta\left\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t\right\rangle + \eta^2\sum_{i=1}^N p_{t,i}\ell_{t,i}^2\right)$$

Proof of Hedge Regret Bound

Proof.

$$\Phi_t - \Phi_{t-1} = \frac{1}{\eta} \ln \left(\frac{\sum_{i=1}^N \exp\left(-\eta L_{t,i}\right)}{\sum_{i=1}^N \exp\left(-\eta L_{t-1,i}\right)} \right)$$

$$\leq - \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle + \eta \sum_{i=1}^N p_{t,i} \ell_{t,i}^2 \qquad (\ln(1+x) \leq x)$$

Summing over t, we have

$$\sum_{t=1}^T \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle \leq \Phi_0 - \Phi_T + \eta \sum_{t=1}^T \sum_{i=1}^N p_{t,i} \ell_{t,i}^2 \qquad \Phi_t \triangleq \frac{1}{\eta} \ln \left(\sum_{i=1}^N \exp\left(-\eta L_{t,i}\right) \right)$$

$$\leq \frac{\ln N}{\eta} - \frac{1}{\eta} \ln\left(\exp\left(-\eta L_{T,i^\star}\right)\right) + \eta \sum_{t=1}^T \sum_{i=1}^N p_{t,i} \ell_{t,i}^2$$

$$\leq \frac{\ln N}{\eta} + L_{T,i^\star} + \eta \sum_{t=1}^T \sum_{i=1}^N p_{t,i} \ell_{t,i}^2$$

Proof of Hedge Regret Bound

Proof.

$$\sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle \le \frac{\ln N}{\eta} + L_{T,i^\star} + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} p_{t,i} \ell_{t,i}^2$$

Rearranging the term gives

$$\begin{aligned} \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle - L_{T,i^\star} &\le \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} p_{t,i} \ell_{t,i}^2 \\ &\le \frac{\ln N}{\eta} + \eta T \qquad (\ell_{t,i} \le 1) \end{aligned}$$

Thus, setting $\eta = \sqrt{\ln N / T}$ yields

$$\text{REG}_T \le \frac{\ln N}{\eta} + \eta T = 2\sqrt{T \ln N}. \qquad \square$$

Lower bound of PEA

- As above, we have proved the regret bound for Hedge:

$$\textsc{Reg}_T \le 2\sqrt{T \ln N}$$

- A natural question: can we further improve the bound?

Theorem 3 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\ell_1,\ldots,\ell_T} \frac{\textsc{Reg}_T}{\sqrt{T \ln N}} \ge \frac{1}{\sqrt{2}}.$$

Hedge achieves minimax optimal regret (up to a constant of $2\sqrt{2}$) for PEA.

Lower bound of PEA

Theorem 3 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\textsc{Reg}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Proof. We construct the 'hard' instance by randomization. Let $\mathcal{D}$ be the uniform distribution over $\{0,1\}$. We have

$$\max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \textsc{Reg}_T \geq \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T \overset{\text{i.i.d.}}{\sim} \mathcal{D}^N} [\textsc{Reg}_T]$$

(conditional expectation decomposition)

$$= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_{t-1}} \mathbb{E}_{\boldsymbol{\ell}_t} [\langle \boldsymbol{p}_t, \boldsymbol{\ell}_t \rangle \mid \boldsymbol{\ell}_{t-1},\ldots,\boldsymbol{\ell}_1] - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T}\left[\min_{i\in[N]} \sum_{t=1}^{T} \ell_{t,i}\right]$$

$$= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_{t-1}} \langle \boldsymbol{p}_t, \mathbb{E}_{\boldsymbol{\ell}_t}[\boldsymbol{\ell}_t \mid \boldsymbol{\ell}_{t-1},\ldots,\boldsymbol{\ell}_1] \rangle - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T}\left[\min_{i\in[N]} \sum_{t=1}^{T} \ell_{t,i}\right]$$

Lower bound of PEA

Theorem 3 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\textsc{Reg}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Proof.

$$\max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \textsc{Reg}_T \geq \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_{t-1}} \langle \boldsymbol{p}_t, \mathbb{E}_{\boldsymbol{\ell}_t} [\boldsymbol{\ell}_t \mid \boldsymbol{\ell}_{t-1},\ldots,\boldsymbol{\ell}_1] \rangle - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \left[\min_{i \in [N]} \sum_{t=1}^{T} \ell_{t,i} \right]$$

$$= T/2 - \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \left[\min_{i \in [N]} \sum_{t=1}^{T} \ell_{t,i} \right] = \mathbb{E}_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \left[\max_{i \in [N]} \sum_{t=1}^{T} \left(\frac{1}{2} - \ell_{t,i} \right) \right]$$

($\boldsymbol{\ell}_t \overset{\text{i.i.d.}}{\sim} \mathcal{D}$ with $\mathcal{D}$ be the uniform distribution over $\{0,1\}$)

$$= \frac{1}{2} \mathbb{E}_{\sigma_1,\ldots,\sigma_T} \left[\max_{i \in [N]} \sum_{t=1}^{T} \sigma_{t,i} \right],$$

(σ_t for $i \in [N], t \in [T]$ are i.i.d. Rademacher random variables)

Lower bound of PEA

Theorem 3 (Lower Bound of PEA). *For any algorithm $\mathcal{A}$, we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\text{REG}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Proof.

$$\max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \text{REG}_T \geq \frac{1}{2}\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_{t,i}\right]$$

($\sigma_{t,i}$ for $i \in [N], t \in [T]$ are i.i.d. Rademacher random variables)

Using the result from probability theory (*Prediction, Learning, and Games*, Chapter 3.7) of Rademacher variables,

$$\Longrightarrow \quad \lim_{T\to\infty} \lim_{N\to\infty} \frac{\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\max_{i\in[N]} \sum_{t=1}^{T} \sigma_{t,i}\right]}{\sqrt{T \ln N}} = \sqrt{2}. \qquad \square$$

Upper Bound and Lower Bound

Theorem 2. *Suppose that* $\forall t \in [T]$ *and* $i \in [N], 0 \leq \boldsymbol{\ell}_{t,i} \leq 1$, *then Hedge with learning rate* η *guarantees*

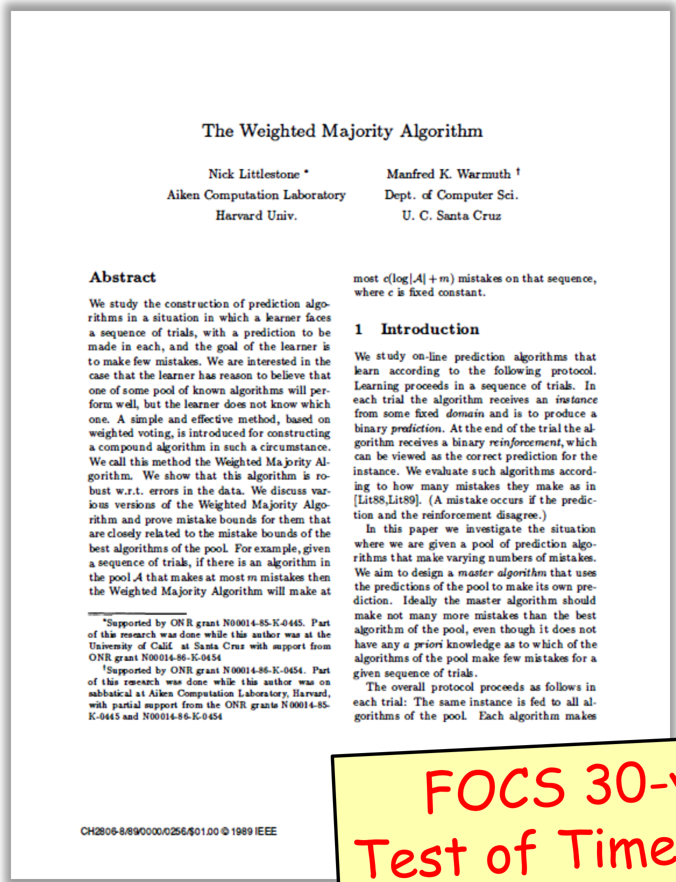
$$\textsc{Reg}_T \leq \frac{\ln N}{\eta} + \eta T = \mathcal{O}\big(\sqrt{T \log N}\big),$$

where the last equality is by setting η *optimally as* $\sqrt{(\ln N)/T}$.

Theorem 3 (Lower Bound of PEA). *For any algorithm* $\mathcal{A}$, *we have that*

$$\sup_{T,N} \max_{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_T} \frac{\textsc{Reg}_T}{\sqrt{T \ln N}} \geq \frac{1}{\sqrt{2}}.$$

Prediction with Expert Advice: history bits



The Weighted Majority Algorithm

Nick Littlestone * — Aiken Computation Laboratory, Harvard Univ.

Manfred K. Warmuth † — Dept. of Computer Sci., U. C. Santa Cruz

Abstract

We study the construction of prediction algorithms in a situation in which a learner faces a sequence of trials, with a prediction to be made in each, and the goal of the learner is to make few mistakes. We are interested in the case that the learner has reason to believe that one of some pool of known algorithms will perform well, but the learner does not know which one. A simple and effective method, based on weighted voting, is introduced for constructing a compound algorithm in such a circumstance. We call this method the Weighted Majority Algorithm. We show that this algorithm is robust w.r.t. errors in the data. We discuss various versions of the Weighted Majority Algorithm and prove mistake bounds for them that are closely related to the mistake bounds of the best algorithms of the pool. For example, given a sequence of trials, if there is an algorithm in the pool $\mathcal{A}$ that makes at most m mistakes then the Weighted Majority Algorithm will make at most $c(\log|\mathcal{A}| + m)$ mistakes on that sequence, where c is fixed constant.

1 Introduction

We study on-line prediction algorithms that learn according to the following protocol. Learning proceeds in a sequence of trials. In each trial the algorithm receives an *instance* from some fixed *domain* and is to produce a binary *prediction*. At the end of the trial the algorithm receives a binary *reinforcement*, which can be viewed as the correct prediction for the instance. We evaluate such algorithms according to how many mistakes they make as in [Lit88,Lit89]. (A mistake occurs if the prediction and the reinforcement disagree.)

In this paper we investigate the situation where we are given a pool of prediction algorithms that make varying numbers of mistakes. We aim to design a *master algorithm* that uses the predictions of the pool to make its own prediction. Ideally the master algorithm should make not many more mistakes than the best algorithm of the pool, even though it does not have any *a priori* knowledge as to which of the algorithms of the pool make few mistakes for a given sequence of trials.

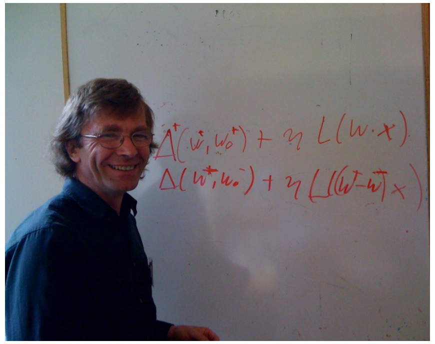
The overall protocol proceeds as follows in each trial: The same instance is fed to all algorithms of the pool. Each algorithm makes

*Supported by ONR grant N00014-85-K-0445. Part of this research was done while this author was at the University of Calif. at Santa Cruz with support from ONR grant N00014-86-K-0454

†Supported by ONR grant N00014-86-K-0454. Part of this research was done while this author was on sabbatical at Aiken Computation Laboratory, Harvard, with partial support from the ONR grants N00014-85-K-0445 and N00014-86-K-0454

CH2806-8/89/0000/0256$01.00 © 1989 IEEE

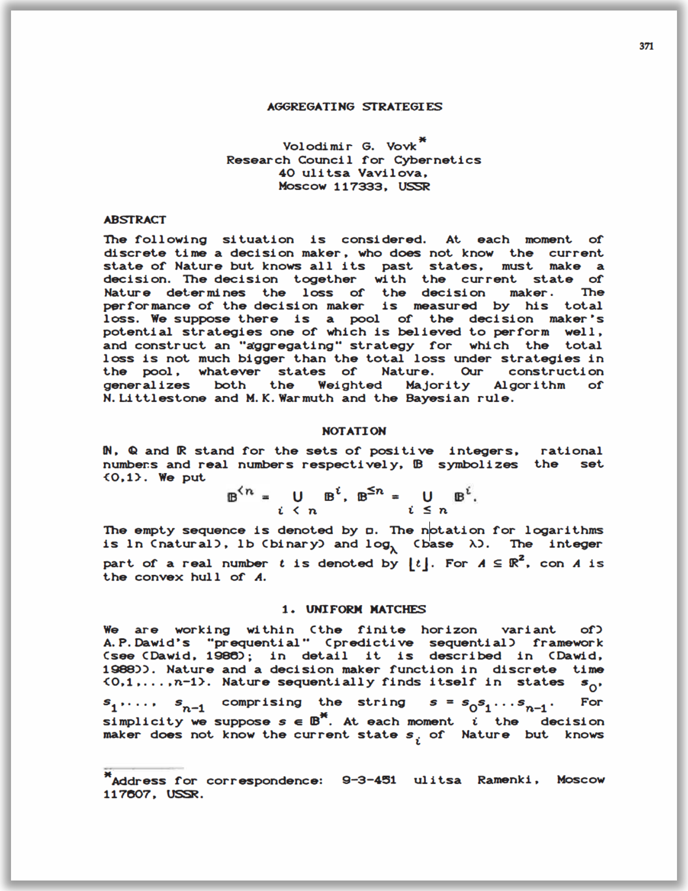
FOCS 30-year
Test of Time Award!



Manfred Warmuth

UC Santa Cruz

Nick Littlestone and Manfred K. Warmuth.
"The Weighted Majority Algorithm." FOCS 1989: 256-261.



371

AGGREGATING STRATEGIES

Volodimir G. Vovk*
Research Council for Cybernetics
40 ulitsa Vavilova,
Moscow 117333, USSR

ABSTRACT

The following situation is considered. At each moment of discrete time a decision maker, who does not know the current state of Nature but knows all its past states, must make a decision. The decision together with the current state of Nature determines the loss of the decision maker. The performance of the decision maker is measured by his total loss. We suppose there is a pool of the decision maker's potential strategies one of which is believed to perform well, and construct an "aggregating" strategy for which the total loss is not much bigger than the total loss under strategies in the pool, whatever states of Nature. Our construction generalizes both the Weighted Majority Algorithm of N. Littlestone and M. K. Warmuth and the Bayesian rule.

NOTATION

$\mathbb{N}$, $\mathbb{Q}$ and $\mathbb{R}$ stand for the sets of positive integers, rational numbers and real numbers respectively, $\mathbb{B}$ symbolizes the set {0,1}. We put

$$\mathbb{B}^{<n} = \bigcup_{i<n} \mathbb{B}^i, \quad \mathbb{B}^{\le n} = \bigcup_{i \le n} \mathbb{B}^i.$$

The empty sequence is denoted by □. The notation for logarithms is ln (natural), lb (binary) and $\log_\lambda$ (base λ). The integer part of a real number t is denoted by $\lfloor t \rfloor$. For $A \subseteq \mathbb{R}^2$, con A is the convex hull of A.

1. UNIFORM MATCHES

We are working within (the finite horizon variant of) A.P. Dawid's "prequential" (predictive sequential) framework (see (Dawid, 1986); in detail it is described in (Dawid, 1988)). Nature and a decision maker function in discrete time {0,1,...,n-1}. Nature sequentially finds itself in states s_0, s_1,..., s_{n-1} comprising the string $s = s_0 s_1 \ldots s_{n-1}$. For simplicity we suppose $s \in \mathbb{B}^*$. At each moment i the decision maker does not know the current state s_i of Nature but knows

*Address for correspondence: 9-3-451 ulitsa Ramenki, Moscow 117607, USSR.

Volodimir G. Vovk
Royal Holloway,
University of London

Volodimir G. Vovk. "Aggregating Strategies."
COLT 1990: 371-383.

Prediction with Expert Advice: history bits

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.

Prediction with Expert Advice: history bits

Yoav Freund

Robert Schapire

Goldel Prize 2003

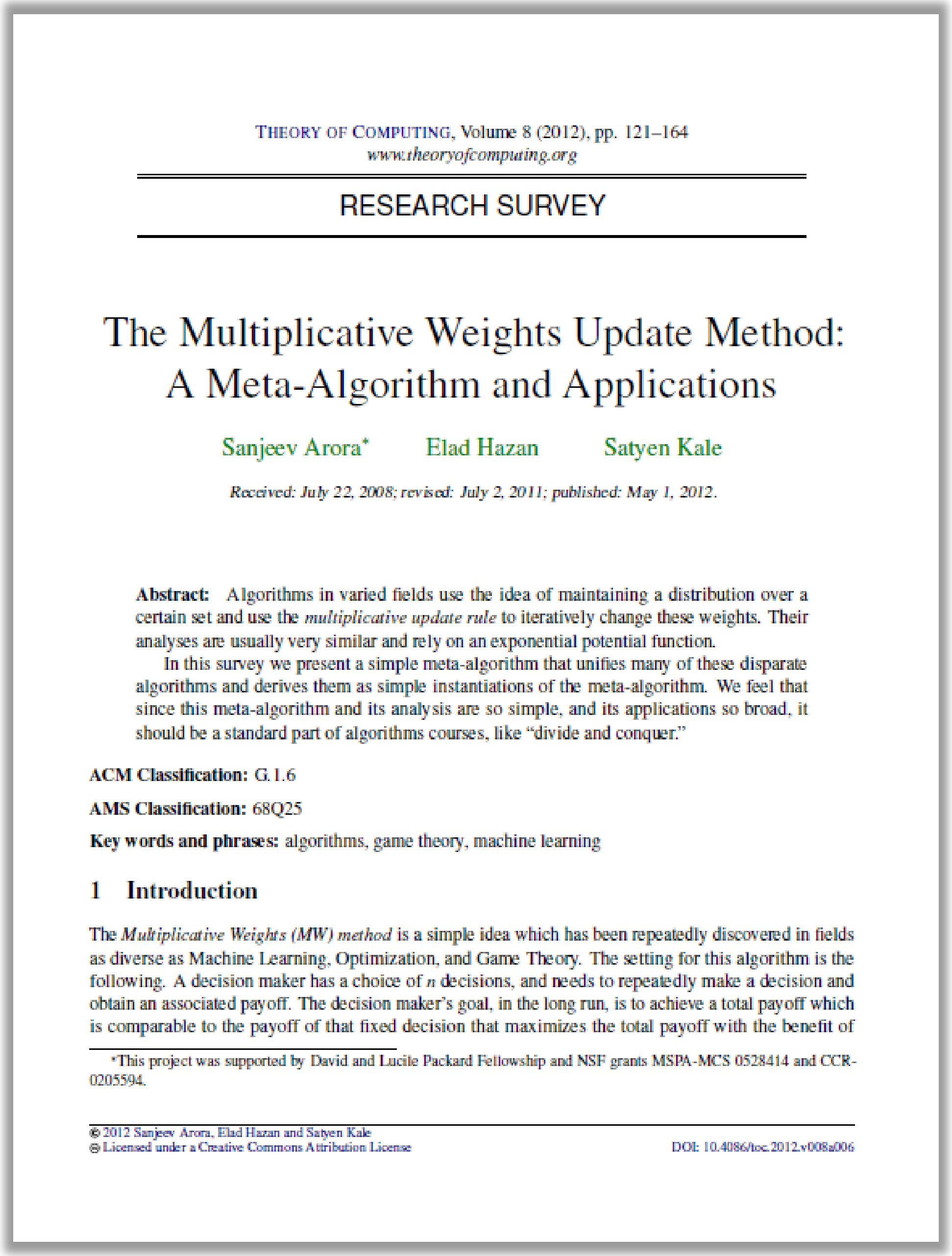
This paper introduced AdaBoost, an adaptive algorithm to improve the accuracy of hypotheses in machine learning. The algorithm demonstrated novel possibilities in analyzing data and is a permanent contribution to science even beyond computer science.

Photo@ICML’24（维也纳，July 22, 2024）

Why is PEA useful?

- Prediction with Expert Advice is essentially a **meta-algorithm** for combining different experts, and the "expert" can be interpreted as any learning model with a particular kind of expertise.

- It is used in a variety of algorithmic design, e.g.,
  - Online Ensemble: A Theoretical Framework for Non-stationary Online Learning @ 2025.05.31
  - Gradient-Variation Online Learning: Theory and Applications @ 2024.06.07.



THEORY OF COMPUTING, Volume 8 (2012), pp. 121–164
www.theoryofcomputing.org

RESEARCH SURVEY

The Multiplicative Weights Update Method: A Meta-Algorithm and Applications

Sanjeev Arora* Elad Hazan Satyen Kale

Received: July 22, 2008; revised: July 2, 2011; published: May 1, 2012.

Abstract: Algorithms in varied fields use the idea of maintaining a distribution over a certain set and use the *multiplicative update rule* to iteratively change these weights. Their analyses are usually very similar and rely on an exponential potential function.

In this survey we present a simple meta-algorithm that unifies many of these disparate algorithms and derives them as simple instantiations of the meta-algorithm. We feel that since this meta-algorithm and its analysis are so simple, and its applications so broad, it should be a standard part of algorithms courses, like "divide and conquer."

ACM Classification: G.1.6

AMS Classification: 68Q25

Key words and phrases: algorithms, game theory, machine learning

1 Introduction

The *Multiplicative Weights (MW) method* is a simple idea which has been repeatedly discovered in fields as diverse as Machine Learning, Optimization, and Game Theory. The setting for this algorithm is the following. A decision maker has a choice of n decisions, and needs to repeatedly make a decision and obtain an associated payoff. The decision maker's goal, in the long run, is to achieve a total payoff which is comparable to the payoff of that fixed decision that maximizes the total payoff with the benefit of

*This project was supported by David and Lucile Packard Fellowship and NSF grants MSPA-MCS 0528414 and CCR-0205594.

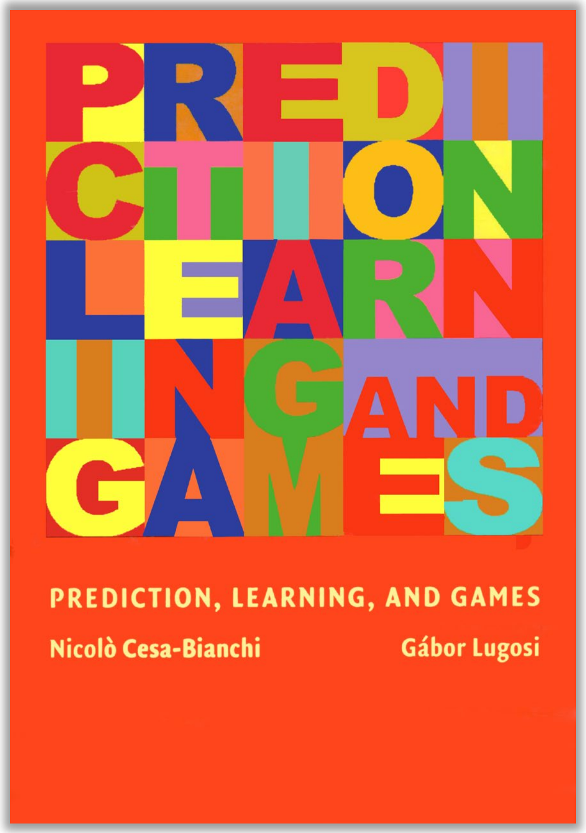
© 2012 Sanjeev Arora, Elad Hazan and Satyen Kale
Licensed under a Creative Commons Attribution License

DOI: 10.4086/toc.2012.v008a006

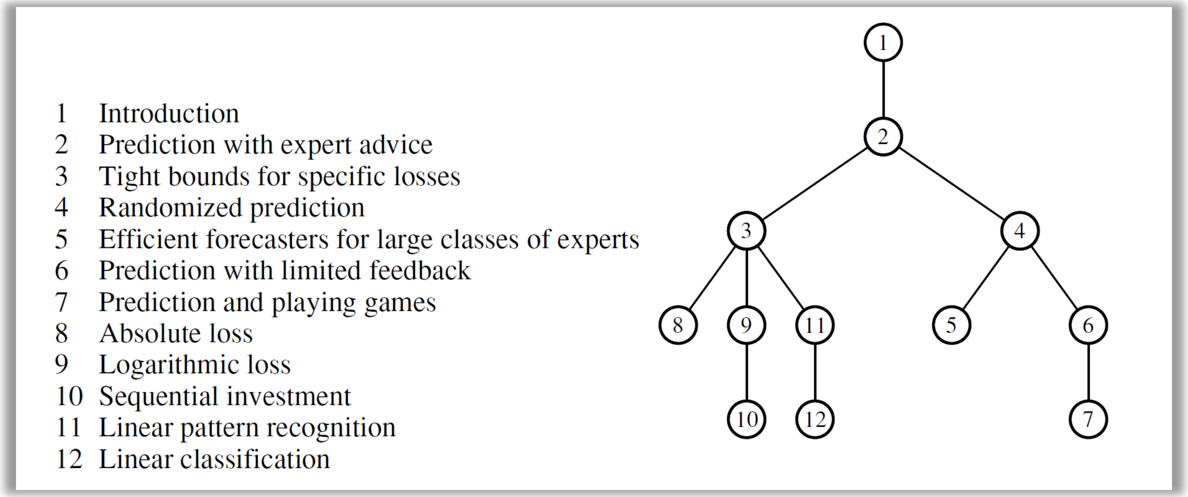
- Applications
  - Learning a linear classifier: the Winnow algorithm
  - Solving zero-sum games approximately
  - Plotkin, Shmoys, Tardos framework for packing/covering LPs
  - Approximating multicommodity flow problems
  - O(log n)-approximation for many NP-hard problems
  - Learning theory and boosting
  - Hard-core sets and the XOR Lemma
  - Hannan's algorithm and multiplicative weights
  - Online convex optimization
  - Other applications
  - Design of competitive online algorithms

The multiplicative weights update method: a meta-algorithm and applications. S Arora, E Hazan, S Kale. Theory of Computing, 2012

More Results on PEA



Prediction, Learning and Games.
Nicolò Cesa-Bianchi and Gabor Lugosi.
Cambridge University Press, 2006.



Nicolò Cesa-Bianchi

Gabor Lugosi

Summary

ONLINE EXP-CONCAVE OPTIMIZATION
- Exp-concave functions
- Online Newton Step
- Regret analysis

PREDICTION WITH EXPERT ADVICE
- Problem setup
- Algorithms
- Regret analysis

Q & A

Thanks!