Universal Online Learning with Gradient Variations: A Multi-layer Online Ensemble Approach (*NeurIPS'23 Spotlight*)

Yu-Hu Yan, Peng Zhao, Zhi-Hua Zhou   LAMDA Group, Nanjing University, China

Our Paper (OpenReview)   Homepage (Yu-Hu Yan)

NANJING UNIVERSITY   LAMDA Learning And Mining from DatA   NEURAL INFORMATION PROCESSING SYSTEMS

Online Convex Optimization (OCO)

Online Learning: data comes as a ***stream***, and model ***online updates***

At each round $t = 1, 2, \dots, T$:
- the learner submits $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$
- at the same time, environments decide a convex loss function f_t
- the learner suffers $f_t(\mathbf{x}_t)$ and receives gradient information

Goal: minimize ***regret*** $\sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$ *The learner's excess loss compared to the best fixed comparator in hindsight.*

Motivation

Traditional Methods: require knowing the ***function curvature*** and obtain ***worst-case*** regret guarantees

Function type	Algorithm	Regret
λ-strongly convex	OGD with $\eta_t = \frac{1}{\lambda t}$	$\mathcal{O}(\log T)$
α-exp-concave	ONS knowing α	$\mathcal{O}(d \log T)$
convex	OGD with $\eta_t \approx \frac{1}{\sqrt{t}}$	$\mathcal{O}(\sqrt{T})$

Burdensome in practice!

*Recent studies consider **two levels of adaptivity**.*

High-Level: adaptive to ***unknown*** function curvature

Target: *a single algorithm that is agnostic to function curvature*

universal algorithm → convex / exp-concave / strongly convex

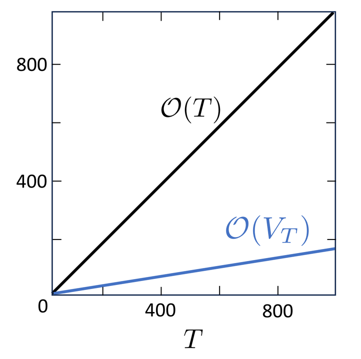
Low-Level: adaptive to ***unknown*** niceness of environments

Target: *regret bounds measured by problem-dependent quantities*

Gradient variation:

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

cumulative variations in gradients, reflecting the difficulty of online problems



The regret bounds can be strengthened to $\mathcal{O}(\log V_T)$, $\mathcal{O}(d \log V_T)$, and $\mathcal{O}(\sqrt{V_T})$.

⇨ **Challenge:** ***handling uncertainties of two levels simultaneously***

Main Result

Theorem 1. *Under standard assumptions, our algorithm obtains $\mathcal{O}(\log V_T)$, $\mathcal{O}(d \log V_T)$ and $\hat{\mathcal{O}}(\sqrt{V_T})$ regret bounds for strongly convex, exp-concave and convex functions, where $\hat{\mathcal{O}}(\cdot)$-notation omits $\log V_T$ factors.*

A single algorithm with simultaneously near-optimal problem-dependent regret bounds for convex/exp-concave/strongly convex functions.

Why Gradient Variation?

Importance in Theory and Practice:
- ➢ *Exploiting the niceness of environments, while safeguarding the worst case*
- ➢ *Applications in **Games** & **Stochastically Extended Adversarial (SEA)***

Gradient Variation in *Games*: [Syrgkanis-Agarwal-Luo-Schapire, NIPS'15]

Gradient of one player encodes the other players' decisions.

Example: 

y-player decision $\mathbf{y}_t = (^1\!/_2\ ^1\!/_2\ 0)^\top$

x-player decision $\mathbf{x}_t = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \end{pmatrix}$

Game matrix A

	Rock	Scissors	Paper
Rock	(0,0)	(1,−1)	(−1,1)
Scissors	(−1,1)	(0,0)	(−1,1)
Paper	(1,−1)	(−1,1)	(0,0)

x-player receives gradient $A\mathbf{y}_t$, y-player receives gradient $A\mathbf{x}_t$.

gradient variation   *negative stability*

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

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

$$\Longrightarrow \mathrm{Reg}_T^x + \mathrm{Reg}_T^y \le \mathcal{O}(1)$$

*The gradient variation is essential for obtaining **fast rates** in games.*

Gradient Variation in *SEA*: [Sachs-Hadiji-Van Erven-Guzmán, NeurIPS'22]

How is loss function f_t generated? — *stochastic* → stochastic optimization $f \sim \mathcal{D}$; *adversarial* → adversarial optimization $\{f_t\}_{t=1}^T$

Setup: f_t is chosen from a distribution $\mathcal{D}_t$: $f_t \sim \mathcal{D}_t$
F_t is the expected function of f_t: $F_t(\cdot) \triangleq \mathbb{E}_{f_t\sim\mathcal{D}_t}[f_t(\cdot)]$

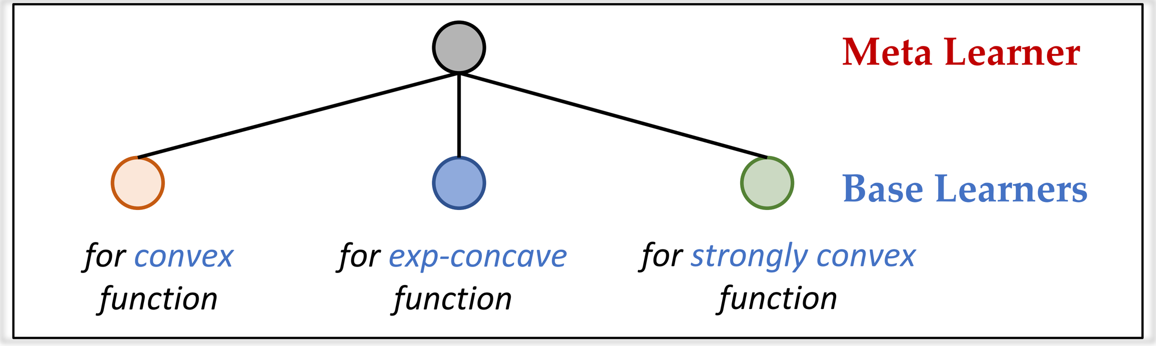
$$\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x}) = \underbrace{[\nabla f_t(\mathbf{x}) - \nabla F_t(\mathbf{x})]}_{\text{stochastic change}} + \underbrace{[\nabla F_t(\mathbf{x}) - \nabla F_{t-1}(\mathbf{x})]}_{\text{adversarial change}} + \underbrace{[\nabla F_{t-1}(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})]}_{\text{stochastic change}}$$

- Stochastic change is the *variance* in stochastic optimization
- Adversarial change is the *gradient variation* in adversarial optimization

Gradient variation successfully bridges stochastic and adversarial optimization.

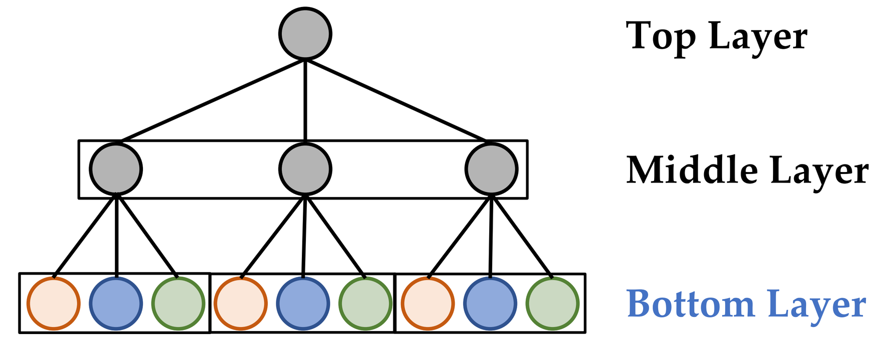
Approach: Multi-layer Online Ensemble

Basic Idea: **(Collaborative) Online Ensemble** [Zhao-Zhang-Zhang-Zhou, 2021]



Ensemble is effective in handling the uncertainty, e.g., in dynamic/adaptive regret minimization.

Our Approach: Multi-layer Online Ensemble



- ➢ Top layer & Middle layer: a *two-layer meta learner*
- ➢ Bottom layer: basic online ensemble idea

Key: *Effective collaboration among multiple layers to control the algorithmic stability.*

Why three layers?

Technically, this is due to the simultaneous requirements of second-order bound (for universality) and negative terms (for gradient variation). Thus we have to use a two-layer algorithm (MsMwC over MsMwC) [Chen-Wei-Luo, COLT'21] *as the meta-learner.*

Technical Challenge

Previous Progress for Universal Online Learning: [Zhang-Wang-Li-Yang, ICML'22]

Regret decomposition:

$$\mathrm{REG}_T = \underbrace{\left[\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_{t,i^\star})\right]}_{\text{meta regret}} + \underbrace{\left[\sum_{t=1}^T f_t(\mathbf{x}_{t,i^\star}) - \min_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^T f_t(\mathbf{x})\right]}_{\text{base regret}}$$

due to the ***online ensemble*** paradigm, i.e., $\mathbf{x}_t = \sum_{i=1}^N p_{t,i}\mathbf{x}_{t,i}$

- **base regret:** black-box optimization
- **meta regret:** e.g., exp-concave functions

$$\sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle \lesssim \sqrt{\sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle^2}$$

(second-order bound, e.g., Adapt-ML-Prod) [Gaillard-Stoltz-Van Erven, COLT'14]

$$\Longrightarrow \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_{t,i^\star}) \le \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle - \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle^2 \le \mathcal{O}(1)$$

Perfect for exp-concave and strongly convex functions, but not for convex ones.

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_{t,i^\star}) \le \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle \lesssim \sqrt{\sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle^2}$$

Cannot obtain the desired $\mathcal{O}(\sqrt{V_T})$ bound for the ***meta regret***!

Technical Contributions

Contribution I: novel ***optimism*** to reuse historical gradients ***universally***

To obtain gradient-variation bounds, we need to reuse historical data, i.e., optimistic online learning.

$$\text{Optimistic-Adapt-ML-Prod: } \sqrt{\sum_{t=1}^T (r_{t,i^\star} - m_{t,i^\star})^2} \quad (r_{t,i} \triangleq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i}\rangle)$$

[Wei-Hong-Lu, NIPS'16]

Challenge: *separate parameters*

- $m_{t,i} = \langle \nabla f_{t-1}(\mathbf{x}_{t-1}), \mathbf{x}_t - \mathbf{x}_{t,i}\rangle$ → convex
- $m_{t,i} = 0$ → exp-concave
- $m_{t,i} = 0$ → strongly convex

different parameters for different functions (not universal)

Our solution:

universal parameter

$m_{t,i} = r_{t-1,i} = \langle \nabla f_{t-1}(\mathbf{x}_{t-1}), \mathbf{x}_{t-1} - \mathbf{x}_{t-1,i}\rangle$ → convex / exp-concave / strongly convex

one parameter for different functions (universal)

$$\sum_{t=1}^T (r_{t,i^\star} - m_{t,i^\star})^2 \le \begin{cases} \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star}\rangle^2, & \text{(exp-concave \& strongly convex)} \\ V_T + \sum_{t=2}^T \|\mathbf{x}_{t,i^\star} - \mathbf{x}_{t-1,i^\star}\|^2 + \underbrace{\sum_{t=2}^T \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2}_{\text{algorithm stability}}. & \text{(convex)} \end{cases}$$

Contribution II: ***collaboration*** in multiple layers to improve ***stability***

Two layers: [Zhao-Zhang-Zhang-Zhou, 2021]

$$\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 \lesssim \underbrace{\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1^2}_{\text{meta stability}} + \underbrace{\sum_{i=1}^N p_{t,i}\|\mathbf{x}_{t,i} - \mathbf{x}_{t-1,i}\|_2^2}_{\text{weighted combination of base stability}}$$

- meta stability: handled by negative terms in meta regret
- weighted stability: collaboration among layers, penalizing unstable base learners

$$\sum_{t=1}^T \langle \boldsymbol{\ell}_t + \boldsymbol{b}_t, \boldsymbol{p}_t - \boldsymbol{e}_{i^\star}\rangle \le X \Longrightarrow \sum_{t=1}^T \langle \boldsymbol{\ell}_t, \boldsymbol{p}_t - \boldsymbol{e}_{i^\star}\rangle \le X - \sum_{t=1}^T\sum_{i=1}^N p_{t,i}b_{t,i} + \sum_{t=1}^T b_{t,i^\star}$$

Three layers:

Positive Term: $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2$

$\mathbf{x}_t = \sum_{k=1}^K q_{t,k}\mathbf{x}_{t,k}$
$\mathbf{x}_{t,k} = \sum_{i=1}^N p_{t,k,i}\mathbf{x}_{t,k,i}$

Top Layer: $\underbrace{\|\boldsymbol{q}_t - \boldsymbol{q}_{t-1}\|_1^2}_{\text{canceled by MsMwC-Top}}$ $\underbrace{\sum_{k=1}^K q_{t,k}\|\mathbf{x}_{t,k} - \mathbf{x}_{t-1,k}\|^2}_{\text{canceled by } -\sum_{k=1}^K q_{t,k}b_{t,k}} + \lambda_1\|\mathbf{x}_{t,k^\star} - \mathbf{x}_{t-1,k^\star}\|^2$

Middle Layer: $\underbrace{\|\boldsymbol{p}_{t,k^\star} - \boldsymbol{p}_{t-1,k^\star}\|_1^2}_{\text{canceled by MsMwC-Mid}}$ $\underbrace{\sum_{i=1}^N p_{t,k^\star,i}\|\mathbf{x}_{t,k^\star,i} - \mathbf{x}_{t-1,k^\star,i}\|^2}_{\text{canceled by } -\sum_{i=1}^N p_{t,k^\star,i}b_{t,k^\star,i}}$

Bottom Layer: $+\lambda_2\|\mathbf{x}_{t,k^\star,i^\star} - \mathbf{x}_{t-1,k^\star,i^\star}\|^2$

A principled way to control algorithmic stability in multi-layer structures.