Lecture 13. Advanced Topics

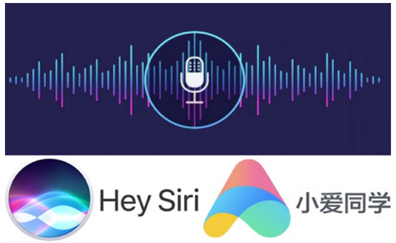
Peng Zhao
School of AI, Nanjing University
zhaop@lamda.nju.edu.cn

Machine Learning

- Machine Learning has achieved great success in recent years.

image recognition

search engine



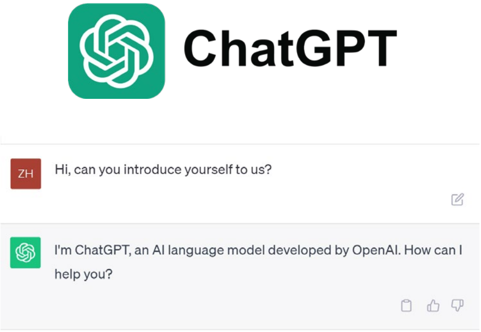
voice assistant

recommendation

AlphaGo Games

automatic driving

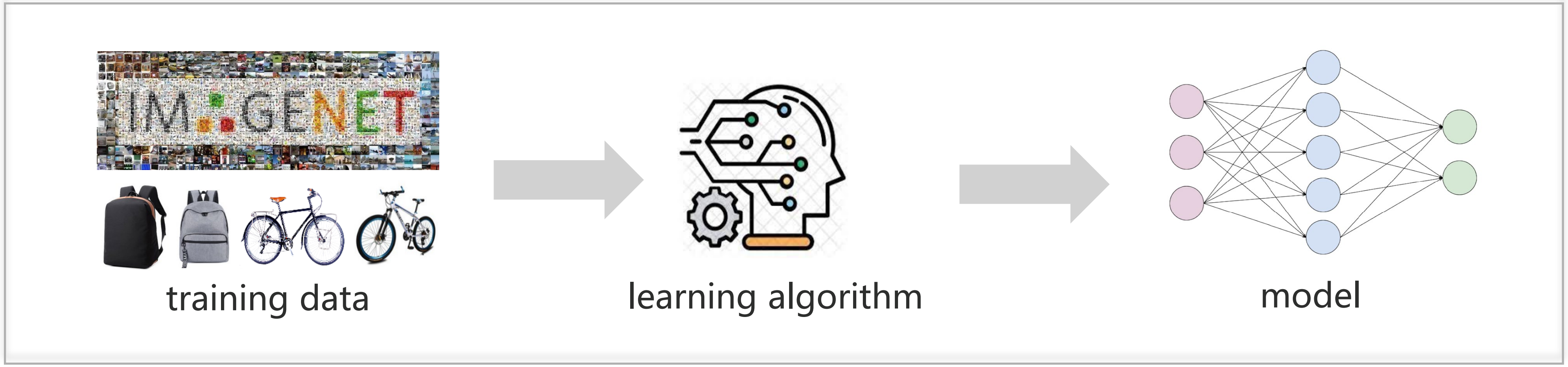
medical diagnosis



large language model

Machine Learning

- A standard pipeline for machine learning deployments.



- Learning as optimization: using ERM to learn the model

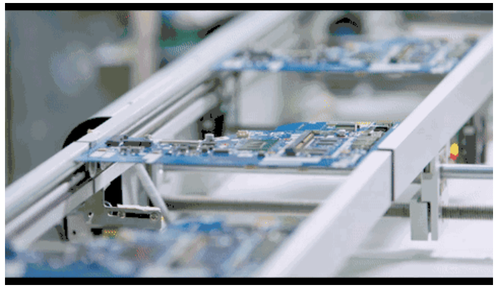
$$\min_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{m} \ell(\mathbf{x}; z_i)$$

learning the model based on the (offline) training dataset $S = \{z_1, \ldots, z_m\}$

Online Learning

- In many applications, data are coming in an ***online*** fashion

base station



manufacturing

- Online learning/optimization

  - update the model in an iterated optimization fashion

  - need to have guarantees for the online update

Outline

- Problem Setup
- Non-stationary Online Learning
- Universal Online Learning
- Conclusion

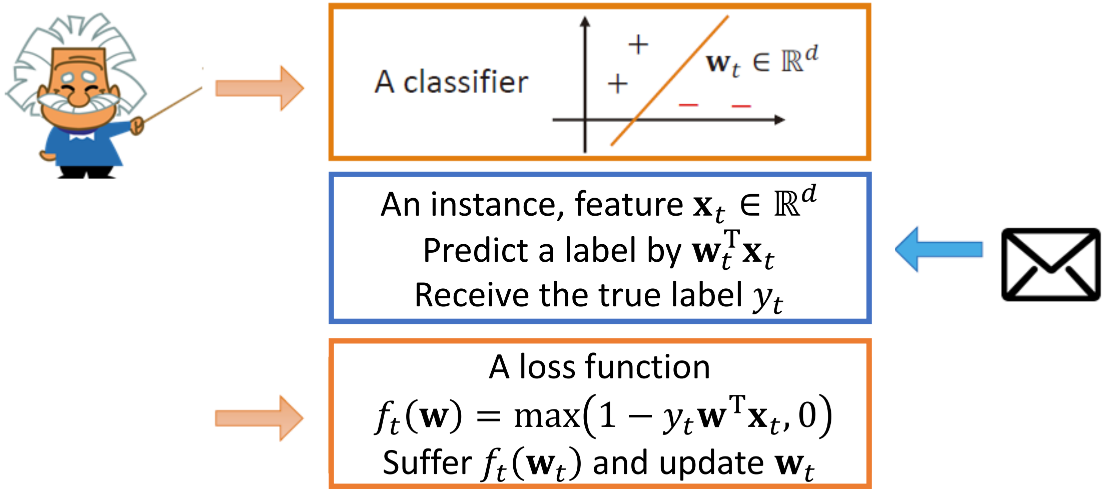
Online Learning

- View online learning as a game between *learner* and *environment*.

Online Convex Optimization

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

1. learner first provides a model $\mathbf{w}_t \in \mathcal{W}$;
2. and simutaneously the environment picks a convex online function $f_t : \mathcal{W} \mapsto [0, 1]$;
3. the learner then suffers loss $f_t(\mathbf{w}_t)$ and observes some information of f_t.



Example: online function $f_t : \mathcal{W} \mapsto \mathbb{R}$ is composition of

(i) loss $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \mapsto \mathbb{R}$, and

(ii) data item: $(\mathbf{x}_t, y_t) \in \mathcal{X} \times \mathcal{Y}$.

$$\Longrightarrow \quad f_t(\mathbf{w}) = \ell(\mathbf{w}^\top \mathbf{x}_t, y_t)$$

Spam Filtering

Regular vs Spam ?

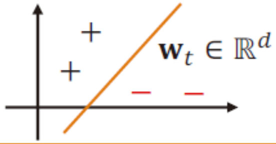
Online Learning

- View online learning as a game between *learner* and *environment*.

Online Convex Optimization

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

1. learner first provides a model $\mathbf{w}_t \in \mathcal{W}$;
2. and simutaneously the environment picks a convex online function $f_t : \mathcal{W} \mapsto [0, 1]$;
3. the learner then suffers loss $f_t(\mathbf{w}_t)$ and observes some information of f_t.

A classifier


An instance, feature $\mathbf{x}_t \in \mathbb{R}^d$
Predict a label by $\mathbf{w}_t^{\mathrm{T}}\mathbf{x}_t$
Receive the true label y_t

A loss function
$f_t(\mathbf{w}) = \max(1 - y_t \mathbf{w}^{\mathrm{T}}\mathbf{x}_t, 0)$
Suffer $f_t(\mathbf{w}_t)$ and update $\mathbf{w}_t$

full information

horse racing

partial information

multi-armed bandits

Spam Filtering

Regular vs Spam ?

Outline

- Problem Setup
- Non-stationary Online Learning
- Universal Online Learning
- Conclusion

Non-stationary Online Learning

- **Distribution shift**: data are usually collected in open environments

species monitoring

urban computing

route planning

- For the online learning scenario, the distributions will evolve over time.

continuous distribution shift

provably robust methods for non-stationary online learning

Community Discussions

“Deep Learning for AI”

Communication of ACM

July, 2021. Vol 64. No 7.

2018 Turing Award Recipients

turing lecture

DOI:10.1145/3448250

How can neural networks learn the rich internal representations required for difficult tasks such as recognizing objects or understanding language?

BY YOSHUA BENGIO, YANN LECUN, AND GEOFFREY HINTON

Deep Learning for AI

TURING LECTURE

Yoshua Bengio, Yann LeCun, and Geoffrey Hinton are recipients of the 2018 ACM A.M. Turing Award for breakthroughs that have made deep neural networks a critical component of computing.

RESEARCH ON ARTIFICIAL neural networks was motivated by the observation that human intelligence emerges from highly parallel networks of relatively simple, non-linear neurons that learn by adjusting the strengths of their connections. This observation leads to a central computational question: How is it possible for networks of this general kind to learn the complicated internal representations that are required for difficult tasks such as recognizing objects or understanding language? Deep learning seeks to answer this question by using many layers of activity vectors as representations and learning the connection strengths that give rise to these vectors by following the stochastic gradient of an objective function that measures how well the network is performing. It is very surprising that such a conceptually simple approach has proved to be so effective when applied to large training sets using huge amounts of computation and it appears that a key ingredient is depth: shallow networks simply do not work as well.

We reviewed the basic concepts and some of the breakthrough achievements of deep learning several years ago.[63] Here we briefly describe the origins of deep learning, describe a few of the more recent advances, and discuss some of the future challenges. These challenges include learning with little or no external supervision, coping with test examples that come from a different distribution than the training examples, and using the deep learning approach for tasks that humans solve by using a deliberate sequence of steps which we attend to consciously—tasks that Kahneman[56] calls *system 2* tasks as opposed to *system 1* tasks like object recognition or immediate natural language understanding, which generally feel effortless.

From Hand-Coded Symbolic Expressions to Learned Distributed Representations

There are two quite different paradigms for AI. Put simply, the logic-inspired paradigm views sequential reasoning as the essence of intelligence and aims to implement reasoning in computers using hand-designed rules of inference that operate on hand-designed symbolic expressions that formalize knowledge. The brain-inspired paradigm views learning representations from data as the essence of intelligence and aims to implement learning by hand-designing or evolving rules for modifying the connec-

58 COMMUNICATIONS OF THE ACM | JULY 2021 | VOL. 64 | NO. 7

What needs to be improved. From the early days, theoreticians of machine learning have focused on the iid assumption, which states that the test cases are expected to come from the same distribution as the training examples. Unfortunately, this is not a realistic assumption in the real world: just consider the non-stationarities due to actions of various agents changing the world, or the gradually expanding mental horizon of a learning agent which always has more to learn and discover. As a practical consequence, the performance of today's best AI systems tends to take a hit when they go from the lab to the field.

Our desire to achieve greater robustness when confronted with changes in distribution (called out-of-distribution generalization) is a special case of the more general objective of reducing sample complexity (the number of examples needed to generalize well) when faced with a new task—as in transfer learning and lifelong learning[81]—or simply with a change in distribution or

Performance Measure

Regret: online prediction as good as the best offline model

$$\text{Regret}_T \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$

cumulative loss of the best offline model

Dynamic Regret

optimal model changes in non-stationary environments

$$\text{D-Regret}(\mathbf{u}_1, \cdots, \mathbf{u}_T) \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

allow changing comparators

The comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T$ essentially depict the underlying (unknown) distributions of all rounds.

- stationary environments: $\mathbf{u}_t = \mathbf{w}_* \in \arg\min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$
- piecewise-stationary environments: $\mathbf{u}_t = \mathbf{w}_*^{\mathcal{I}_k}$ for a stationary interval $t \in \mathcal{I}_k$

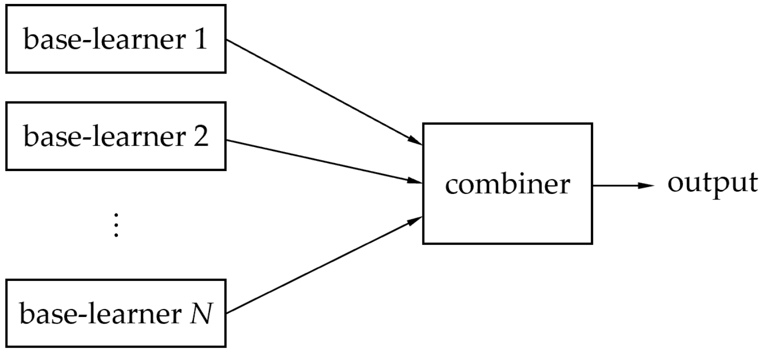
Fundamental Challenge

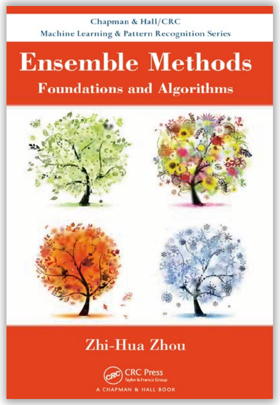
$$\text{D-Regret}(\mathbf{u}_1, \cdots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

Key difficulty: the ***uncertainty*** due to unknown environmental changes.

Basic idea: **Ensemble Methods**

- ***Protocol***: combine multiple base learners to achieve robustness
- ***Advantage***: achieve more robust results under uncertain or even changing environments





Zhi-Hua Zhou. Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC, Jun. 2012.

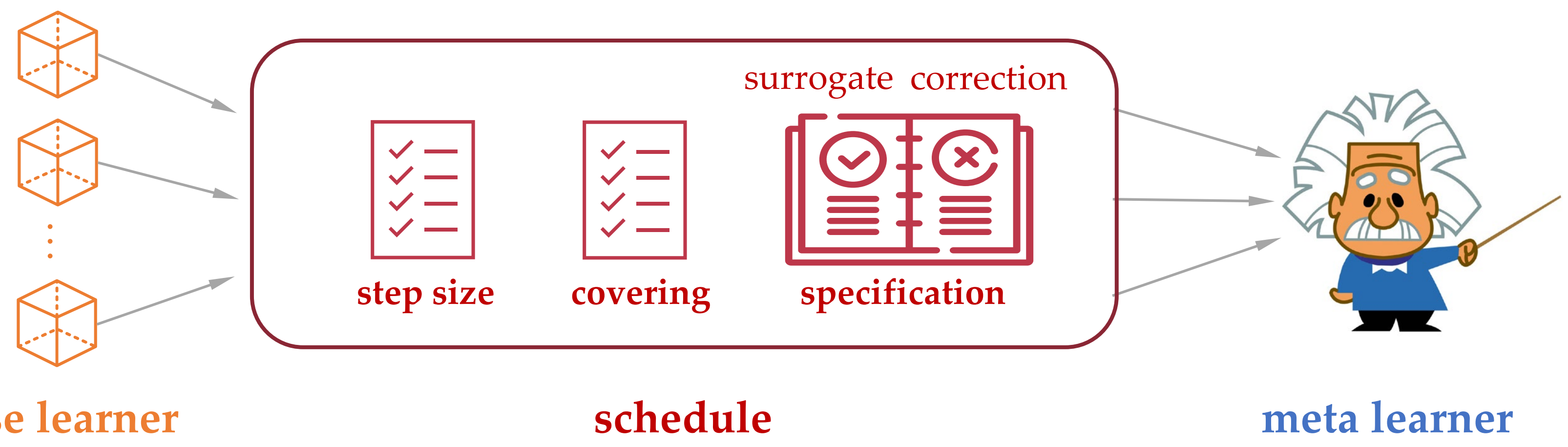
Online Ensemble (在线集成)

Basic Components

(1) **base learner**: an online learner to cope with a certain amount of non-stationarity

(2) **schedule**: a set of parameters for initiating base learners that encourage diversity

(3) **meta learner**: an expert-tracking learner that can combine base learners' decisions



Deploying Online Ensemble

We will showcase that properly deploying online ensemble can effectively resolve several important online learning problems.

- Dynamic Regret of Bandit Convex Optimization

- Problem-dependent Dynamic Regret

Deploying Online Ensemble

We will showcase that properly deploying online ensemble can effectively resolve several important online learning problems.

- Dynamic Regret of Bandit Convex Optimization

- Problem-dependent Dynamic Regret

Bandit Convex Optimization (BCO)

- BCO with one-point feedback

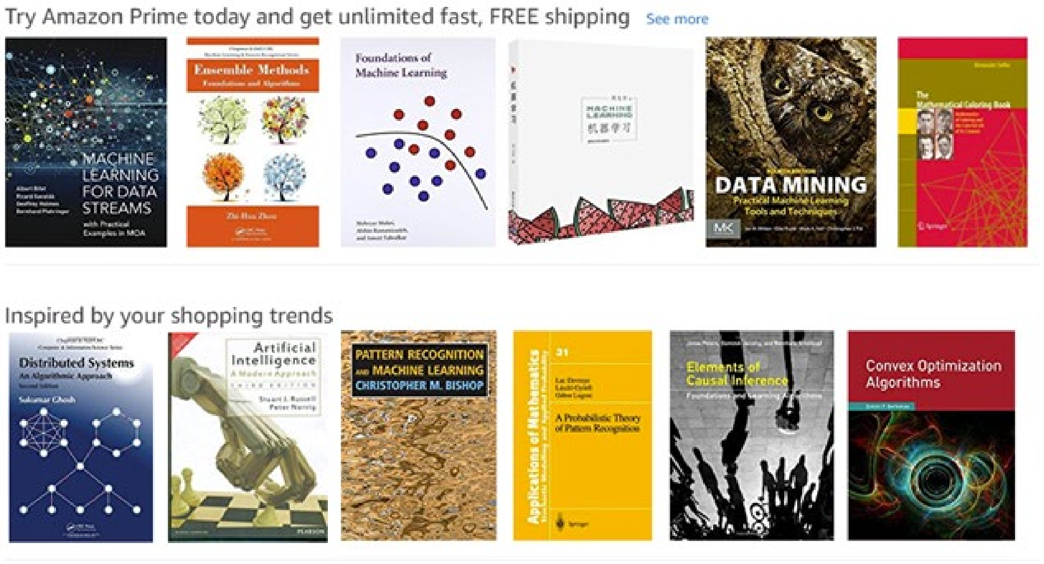
  the learner sends a single point $\mathbf{w}_t \in \mathcal{W}$, and then receives the ***function value*** $f_t(\mathbf{w}_t)$ only

  [Flaxman et al., SODA 2005; Bubeck et al., STOC 2017]

- BCO with two-point feedback

  the learner sends two points $\mathbf{w}_t^1, \mathbf{w}_t^2 \in \mathcal{W}$, and then receives their ***function values***, namely, $f_t(\mathbf{w}_t^1)$ and $f_t(\mathbf{w}_t^2)$, only

  [Agarwal et al., COLT 2010; Shamir, JMLR 2017]



online recommendation

A Gentle Start

Online Gradient Descent (OGD)

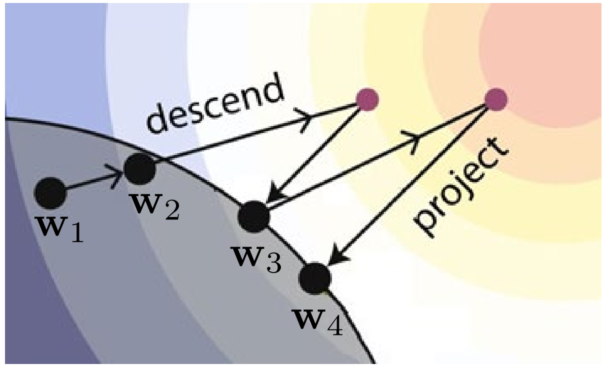
for $t = 1$ to T **do**

Play model $\mathbf{w}_t$ and suffer loss $f_t(\mathbf{w}_t)$

Update the model

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)]$$

end for



https://www.nature.com/articles/s41534-017-0043-1

Challenge: with only bandit feedback, the learner *cannot evaluate the gradient*

FKM estimator [Flaxman et al., SODA'05]

construct $\mathbf{w}_t$ using the perturbation technique

$$\mathbf{w}_t \triangleq \widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t$$

$\mathbf{s}_t$ is random vector sampled from ball $\mathbb{B} = \{\mathbf{v} \mid \|\mathbf{v}\| \leq 1\}$

⇨ $$\mathbb{E}\left[\frac{d}{\delta} f_t(\mathbf{w}_t) \cdot \mathbf{s}_t\right] = \nabla \widehat{f}_t(\widetilde{\mathbf{w}}_t)$$

[proved by Stokes equation]

with $\widehat{f}_t(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{v} \in \mathbb{B}}[f_t(\mathbf{w} + \delta \mathbf{v})]$ being smoothed function.

⇨ define $\mathbf{g}_t \triangleq \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$ as gradient estimator

A Gentle Start

Online Gradient Descent (OGD)

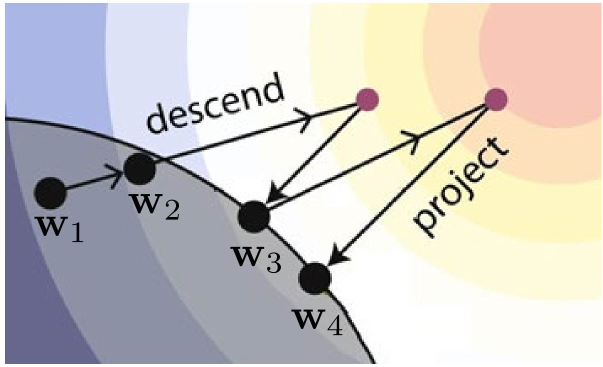
for $t = 1$ to T **do**

Play model $\mathbf{w}_t$ and suffer loss $f_t(\mathbf{w}_t)$

Update the model

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)]$$

end for



https://www.nature.com/articles/s41534-017-0043-1

Challenge: with only bandit feedback, the learner *cannot evaluate the gradient*

FKM estimator [Flaxman et al., SODA'05]

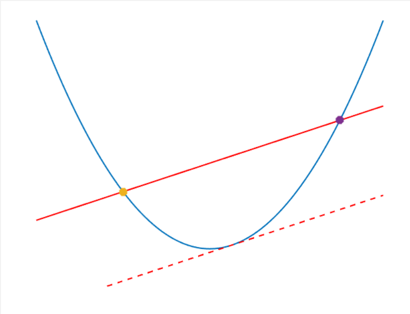
construct $\mathbf{w}_t$ using the perturbation technique

$$\mathbf{w}_t \triangleq \widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t$$

$\mathbf{s}_t$ is random vector sampled from ball $\mathbb{B} = \{\mathbf{v} \mid \|\mathbf{v}\| \leq 1\}$

Consider the 1-dim case ($d = 1$).

$$\mathbb{E}_{\mathbf{s} \in \mathbb{S}} \left[\frac{d}{\delta} f_t(\widetilde{\mathbf{w}} + \delta \mathbf{s}) \cdot \mathbf{s} \right] = \frac{1}{2\delta} f_t(\widetilde{w} + \delta) - \frac{1}{2\delta} f_t(\widetilde{w} - \delta)$$



Base Algorithm : BGD

- Gradient estimator: $\mathbf{g}_t = \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$
- Perform Online Gradient Descent using this gradient estimator.

Bandit Gradient Descent (BGD)

for $t = 1$ to T **do**

Select a unit vector $\mathbf{s}_t$ uniformly at random

Submit $\mathbf{w}_t = \widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t$

Receive $f_t(\mathbf{w}_t)$ as the feedback

Construct the gradient estimator by $\mathbf{g}_t = \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$

$\widetilde{\mathbf{w}}_{t+1} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_t - \eta \mathbf{g}_t]$

end for

$\mathbb{E}[\mathbf{g}_t] = \nabla \widehat{f}_t(\widetilde{\mathbf{w}}_t)$

$\widehat{f}_t(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{v} \in \mathbb{B}}[f_t(\mathbf{w} + \delta \mathbf{v})]$

Base Algorithm: Dynamic Regret

Theorem 1. Under certain standard assumptions, for any perturbation parameter $\delta > 0$, step size $\eta > 0$, and shrinkage parameter $\alpha = \delta/r$, the expected dynamic regret of $\text{BGD}(T, \delta, \alpha, \eta)$ for the one-point feedback model satisfies

$$\mathbb{E}\left[\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T)\right] \leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2} + \left(3L + \frac{LR}{r}\right)\delta T$$
$$= \mathcal{O}\left(\frac{1 + P_T}{\eta} + \frac{\eta T}{\delta^2} + \delta T\right),$$

where $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ measures the non-stationarity level.

Optimal parameter setting is

- step size $\eta_* = \left(\frac{7R^2 + RP_T}{T}\right)^{\frac{3}{4}}$ $\Longrightarrow$ $\mathcal{O}\left(T^{3/4}(1 + P_T)^{1/4}\right)$
- perturbation parameter $\delta_* = \eta_*^{\frac{1}{3}}$

Base Algorithm: Dynamic Regret

Theorem 1. Under certain standard assumptions, for any perturbation parameter $\delta > 0$, step size $\eta > 0$, and shrinkage parameter $\alpha = \delta/r$, the expected dynamic regret of $\mathrm{BGD}(T, \delta, \alpha, \eta)$ for the one-point feedback model satisfies

$$\mathbb{E}\left[\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T)\right] \leq \frac{7R^2 + RP_T}{4\eta} + \frac{\eta d^2 C^2 T}{2\delta^2} + \left(3L + \frac{LR}{r}\right)\delta T$$
$$= \mathcal{O}\left(\frac{1 + P_T}{\eta} + \frac{\eta T}{\delta^2} + \delta T\right),$$

where $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ measures the non-stationarity level.

Optimal parameter setting is

- step size $\eta_* = \left(\frac{7R^2 + RP_T}{T}\right)^{\frac{3}{4}}$
- perturbation parameter $\delta_* = \eta_*^{\frac{1}{3}}$

Comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T$ can be arbitrary, we cannot know non-stationarity P_T in advance, ***so how to tune the step size ?***

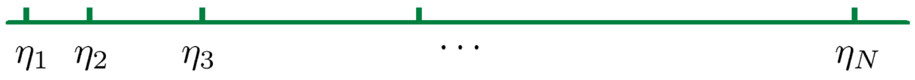
Online Ensemble for BCO

Deploying a proper online ensemble to deal with the issue of ***unknown non-stationarity***, so that we can ***optimally tune step size***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{w}_{t+1,i}$$

- **Multiple candidates:** to cover uncertainty

  diversity consideration: cover all the possible range using as fewer as possible discretization items



$$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots, N \right\}$$

with $N = \lceil \log_2(1 + 2T/7) \rceil + 1 = \mathcal{O}(\log T)$.

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\text{BGD}(\eta_i): \quad \widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t^{\eta_i}]$$

$$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$$

- **Meta algorithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

  increase weight on base-learners with better performance

$$\text{Hedge:} \quad p_{t+1,i} \propto p_{t,i} \exp(-\varepsilon f_t(\mathbf{w}_{t,i}))$$

Online Ensemble for BCO

Deploying a proper online ensemble to deal with the issue of ***unknown non-stationarity***, so that we can ***optimally tune step size***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{w}_{t+1,i}$$

- **Multiple candidates:** to cover uncertainty

  diversity consideration: cover all the possible range using as fewer as possible discretization items

  η_1 η_2 η_3 $\cdots$

  $$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots \right\}$$

  with $N = \lceil \log_2(1 + 2T/7) \rceil + 1 = \mathcal{O}(\log T)$.

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

  BGD(η_i): $\widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t^{\eta_i}]$

  $\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$

- **…orithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

  weight on base-learners with better performance

  Hedge: $p_{t+1,i} \propto p_{t,i}\exp(-\varepsilon f_t(\mathbf{w}_{t,i}))$

***bandit** feedback makes it hard to initiate **multiple** base learners*

Multiple base learners in BCO

- A closer look at dynamic regret analysis

$$\widetilde{\mathbf{w}}_{t+1} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_t - \eta \mathbf{g}_t], \ \mathbb{E}[\mathbf{g}_t] = \nabla \widehat{f}_t(\widetilde{\mathbf{w}}_t).$$

smoothed function $\widehat{f}_t(\mathbf{w}) = \mathbb{E}_{\mathbf{v}\in\mathbb{B}}[f_t(\mathbf{w} + \delta\mathbf{v})]$

rescaled comparator $\mathbf{v}_t = (1-\alpha)\mathbf{u}_t$

$$\sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)$$

$$= \underbrace{\sum_{t=1}^T \widehat{f}_t(\widetilde{\mathbf{w}}_t) - \sum_{t=1}^T \widehat{f}_t(\mathbf{v}_t)}_{\texttt{term (a)}} + \underbrace{\sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T \widehat{f}_t(\widetilde{\mathbf{w}}_t)}_{\texttt{term (b)}} + \underbrace{\sum_{t=1}^T \widehat{f}_t(\mathbf{v}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)}_{\texttt{term (c)}}$$

term (a): *depends on* P_T

term (b): $\leq 2L\delta T$

term (c): $\leq (L\delta + L\alpha R)T$

crucial term, related to non-stationarity measure P_T

not involve the unknown non-stationarity measure P_T

(approximation error due to the perturbation operation)

Multiple base learners in BCO

- Key idea: ***surrogate optimization***

Proposition 1. For any $t \in [T]$, the following holds true:

$$\mathbb{E}[\widehat{f}_t(\widetilde{\mathbf{w}}_t) - \widehat{f}_t(\mathbf{v}_t)] \leq \mathbb{E}[\langle \mathbf{g}_t, \widetilde{\mathbf{w}}_t - \mathbf{v}_t \rangle],$$

where $\mathbf{g}_t = \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$ is the one-point gradient estimator.

- Construct the surrogate loss $\ell_t(\mathbf{w}) \triangleq \langle \mathbf{g}_t, \mathbf{w} \rangle$

  which is a linearized loss parametrized by the gradient estimator $\mathbf{g}_t$.

Feed this surrogate loss to online ensemble to maintain multiple base learners!

Surrogate Loss

- Construct the surrogate loss $\ell_t(\mathbf{w}) \triangleq \langle \mathbf{g}_t, \mathbf{w} \rangle$ and feed it to online ensemble.

Theorem 2. The constructed surrogate loss satifies the following properties:

(i) $\mathbb{E}\big[\widehat{f}_t(\widetilde{\mathbf{w}}_t) - \widehat{f}_t(\mathbf{v})\big] \leq \mathbb{E}\left[\ell_t(\widetilde{\mathbf{w}}_t) - \ell_t(\mathbf{v})\right]$ holds for any $\mathbf{v} \in \mathcal{W}$.

(ii) $\nabla \ell_t(\mathbf{w}) = \mathbf{g}_t$ holds for any $\mathbf{w} \in \mathcal{W}$.

- Property (i) implies that it suffices to optimize ***dynamic regret of surrogate loss***.
- Property (ii) implies that it is feasible to ***deploy multiple base learners*** to perform BGD over the ***surrogate loss***.

All the gradients $\nabla \ell_t(\widetilde{\mathbf{w}}_t^1) = \nabla \ell_t(\widetilde{\mathbf{w}}_t^2) = \cdots = \nabla \ell_t(\widetilde{\mathbf{w}}_t^N) = \mathbf{g}_t$, so they can be obtained by querying the function value of f_t ***only once***.

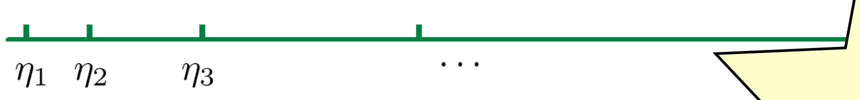
Online Ensemble for BCO

Deploying a proper online ensemble to deal with the issue of ***unknown non-stationarity***, so that we can ***optimally tune step size***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i} \mathbf{w}_{t+1,i}$$

- **Multiple candidates:** to cover uncertainty

  diversity consideration: cover all the possible range using as fewer as possible discretization items



$$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots \right\}$$

with $N = \lceil \log_2(1 + 2T/7) \rceil + 1 = \mathcal{O}(\log T)$.

bandit *feedback makes it hard to initiate* ***multiple*** *base learners*

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\text{BGD}(\eta_i): \quad \widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t^{\eta_i}]$$

$$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$$

- **...orithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

  ...weight on base-learners with better performance

$$\text{Hedge:} \quad p_{t+1,i} \propto p_{t,i} \exp(-\varepsilon f_t(\mathbf{w}_{t,i}))$$

Online Ensemble for BCO

Deploying a proper online ensemble to deal with the issue of ***unknown non-stationarity***, so that we can ***optimally tune step size***.

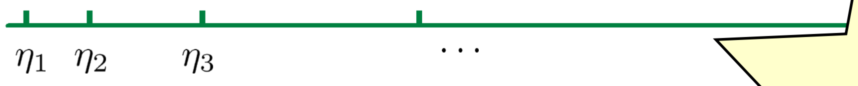
$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{w}_{t+1,i}$$

$$\mathbf{g}_t = \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$$

$$\ell_t(\mathbf{w}) = \langle \mathbf{g}_t, \mathbf{w} \rangle$$

- **Multiple candidates:** to cover uncertainty

diversity consideration: cover all the possible range using as fewer as possible discretization items



$$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots \right\}$$

with $N = \lceil \log_2(1 + 2T/7) \rceil + 1 = \mathcal{O}(\log T)$.

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

BGD(η_i): $\widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t]$

$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$

- **...orithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

...weight on base-learners with better performance

Hedge: $p_{t+1,i} \propto p_{t,i} \exp(-\varepsilon \ell_t(\mathbf{w}_{t,i}))$

surrogate loss
makes online ensemble possible in bandit!

Dynamic Regret

Theorem 3. *Under certain standard assumptions, with a proper setting of the pool of candidate step sizes $\mathcal{H}$ and the learning rate ϵ for the meta-algorithm, our* PBGD *algorithm enjoys the following expected dynamic regret guarantees.*

- *For the one-point feedback model,* $\mathbb{E}[\text{D-Regret}_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)] \leq \mathcal{O}(T^{\frac{3}{4}}(1 + P_T)^{\frac{1}{2}})$.
- *For the two-point feedback model,* $\mathbb{E}[\text{D-Regret}_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)] \leq \mathcal{O}(T^{\frac{1}{2}}(1 + P_T)^{\frac{1}{2}})$.

We further establish the lower bound to demonstrate the hardness of the problem: an $\Omega(\sqrt{TP_T})$ regret is unavoidable for bandit feedback models.

⇨ *Our algorithm is minimax optimal for two-point BCO model; while it remains open how to close the gap in one-point BCO.*

Online Ensemble for BCO

Deploying a proper online ensemble to deal with the issue of ***unknown non-stationarity***, so that we can ***optimally tune step size***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i} \mathbf{w}_{t+1,i}$$

*Proper **surrogate loss** is essential for deploying online ensemble to bandit online problems.*

- **Multiple candidates:** to cover uncertainty

  diversity consideration: cover all the possible range using as fewer as possible discretization items

  $\eta_1 \quad \eta_2 \quad \eta_3 \quad \cdots \quad \eta_N$

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots, N \right\}$$

$$\text{with } N = \lceil \log_2(1 + 2T/7) \rceil + 1 = \mathcal{O}(\log T).$$

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\text{BGD}(\eta_i): \quad \widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t]$$

$$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$$

- **Meta algorithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

  increase weight on base-learners with better performance

$$\text{Hedge:} \quad p_{t+1,i} \propto p_{t,i} \exp(-\varepsilon \ell_t(\mathbf{w}_{t,i}))$$

Deploying Online Ensemble

We will showcase that properly deploying online ensemble can effectively resolve several important online learning problem.

- Dynamic Regret of Bandit Convex Optimization

- Problem-dependent Dynamic Regret

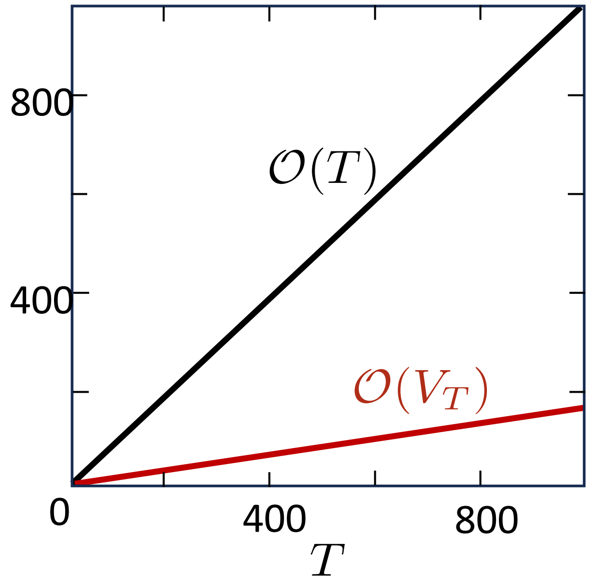
Beyond the worst-case analysis

- Previously, we have achieved minimax results like $\mathcal{O}(\sqrt{T(1+P_T)})$.

- More ambitious: achieving ***problem-dependent*** guarantees
  - ▶ become tighter than worst-case results for benign problems
  - ▶ safeguard the same minimax rate in the worst case

gradient variation

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



It is also essential due to profound connections with many other areas such as online games, stochastic optimization, etc.

Exploiting historical information

- How to exploit the niceness of the environments?

focusing on the gradient feedback for simplicity

Optimistic Online Gradient Descent [Rakhlin and Sridharan, 2013]

$$\widehat{\mathbf{w}}_{t+1} = \Pi_{\mathcal{W}}\left[\widehat{\mathbf{w}}_t - \eta \nabla f_t\left(\mathbf{w}_t\right)\right]$$
$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left[\widehat{\mathbf{w}}_{t+1} - \eta M_{t+1}\right].$$

where $\{M_1, M_2, \ldots, M_T\}$ is the ***hint sequence*** encoding prior knowledge of future.

- If the environment is benign, which means it is "predictable", and thus we can provide the $\{M_t\}_{t=1}^T$ sequence by exploiting historical information.
- A two-step update fashion, and it will degenerate as the standard OGD when there is no external hint (simply setting $M_t = \mathbf{0}$).

Base Algorithm Analysis

- Optimistic OGD can serve as the base learner for problem-dependent dynamic regret minimization.

$$\widehat{\mathbf{w}}_{t+1} = \Pi_{\mathcal{W}}\left[\widehat{\mathbf{w}}_t - \eta \nabla f_t(\mathbf{w}_t)\right]$$
$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left[\widehat{\mathbf{w}}_{t+1} - \eta M_{t+1}\right].$$

Theorem 4. *Under certain standard assumptions, the dynamic regret of optimistic OGD over comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is bounded as*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq GD + \frac{1}{2\eta}\underbrace{(D^2 + 2DP_T)}_{\text{non-stationarity}} + \eta \underbrace{\sum_{t=2}^{T} \|\nabla f_t(\mathbf{w}_t) - M_t\|^2}_{\text{adaptivity}} \underbrace{- \frac{1}{\eta}\sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2}_{\text{negative term}}$$
$$= \mathcal{O}\left(\frac{1+P_T}{\eta} + \eta A_T\right),$$

crucial for gradient variation

where $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ measures non-stationarity and $A_T = \sum_{t=2}^{T} \|\nabla f_t(\mathbf{w}_t) - M_t\|^2$ reflects adaptivity.

Online Ensemble for Adaptive Bounds

- An online ensemble to balance between ***non-stationarity*** and ***adaptivity***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i} \mathbf{w}_{t+1,i}$$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \mathcal{O}\left(\frac{1+P_T}{\eta} + \eta A_T\right)$$

▶ **Multiple candidates:** to cover uncertainty

diversity consideration: cover all the possible range using as fewer as possible discretization items

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \frac{D}{2GT} \mid i = 1, \ldots, N \right\}$$

with $N = \lceil \log_2(GT/(8D^2L^2)) \rceil + 1 = \mathcal{O}(\log T)$.

▶ **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\widehat{\mathbf{w}}_{t+1,i} = \Pi_{\mathcal{W}} \left[\widehat{\mathbf{w}}_{t,i} - \eta_i \nabla f_t(\mathbf{w}_t) \right]$$
$$\mathbf{w}_{t+1,i} = \Pi_{\mathcal{W}} \left[\widehat{\mathbf{w}}_{t+1,i} - \eta_i M_{t+1} \right].$$

▶ **Meta algorithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

also include the "hint" in the performance evaluation

Hedge: $p_{t+1,i} \propto \exp\left(-\varepsilon(L_{t,i} + m_{t+1,i})\right), \ \forall i \in [N]$.

$$L_{t,i} \triangleq \sum_{s=1}^{t} \ell_s(\mathbf{w}_{s,i}) = \sum_{s=1}^{t} \langle \nabla f_s(\mathbf{w}_s), \mathbf{w}_{s,i} \rangle, \ m_{t+1,i} \triangleq \langle M_{t+1}, \mathbf{w}_{t,i} \rangle.$$

Online Ensemble for Adaptive Bounds

- An online ensemble to balance between ***non-stationarity*** and ***adaptivity***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i} \mathbf{w}_{t+1,i}$$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \le \mathcal{O}\left(\frac{1+P_T}{\eta} + \eta A_T\right) = \mathcal{O}\left(\sqrt{A_T(1+P_T)}\right)$$

▶ **Multiple candidates:** to cover uncertainty

diversity consideration: cover all the possible range using as fewer as possible discretization items

$$\mathcal{H} = \left\{\eta_i = 2^{i-1}\frac{D}{2GT} \mid i = 1, \ldots, N\right\}$$

$$\text{with } N = \lceil \log_2(GT/(8D^2L^2)) \rceil + 1 = \mathcal{O}(\log T).$$

▶ **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\widehat{\mathbf{w}}_{t+1,i} = \Pi_{\mathcal{W}}\left[\widehat{\mathbf{w}}_{t,i} - \eta_i \nabla f_t(\mathbf{w}_t)\right]$$
$$\mathbf{w}_{t+1,i} = \Pi_{\mathcal{W}}\left[\widehat{\mathbf{w}}_{t+1,i} - \eta_i M_{t+1}\right].$$

▶ **Meta algorithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

also include the "hint" in the performance evaluation

$$\text{Hedge: } p_{t+1,i} \propto \exp\left(-\varepsilon(L_{t,i} + m_{t+1,i})\right), \ \forall i \in [N].$$

$$L_{t,i} \triangleq \sum_{s=1}^{t} \ell_s(\mathbf{w}_{s,i}) = \sum_{s=1}^{t} \langle \nabla f_s(\mathbf{w}_s), \mathbf{w}_{s,i} \rangle, \ m_{t+1,i} \triangleq \langle M_{t+1}, \mathbf{w}_{t,i} \rangle.$$

Gradient-Variation Dynamic Regret

- From adaptive bound to ***gradient-variation*** regret bound

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \mathcal{O}\left(\sqrt{A_T(1+P_T)}\right)$$

non-stationarity $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$

adaptivity $A_T = \sum_{t=2}^{T} \|\nabla f_t(\mathbf{w}_t) - M_t\|^2$

gradient variation $V_T \triangleq \sum_{t=2}^{T} \sup_{\mathbf{w}\in\mathcal{W}} \|\nabla f_{t-1}(\mathbf{w}) - \nabla f_t(\mathbf{w})\|_2^2$ ***problem-dependent***

⇨ setting $M_{t+1} = \nabla f_t(\mathbf{w}_t)$ as the last-round gradient

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \mathcal{O}\left(\sqrt{(1+P_T)\cdot \sum_{t=2}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|^2}\right)$$

only "data-dependent"

need to analyze $\|\mathbf{w}_t - \mathbf{w}_{t-1}\|^2$ (*stability* of the dynamics)

Stability Analysis

- Stability of the meta-base online ensemble

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{w}_{t+1,i} \quad \Longrightarrow \quad \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \leq \underbrace{2D^2 \left\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\right\|_1^2}_{\text{meta stability}} + \underbrace{2\sum_{i=1}^{N} p_{t,i} \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2}_{\text{weighted combine of base stability}}$$

meta stability   *weighted combine of base stability*

- Decompose the overall dynamic regret into the meta-base two levels:

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) = \underbrace{\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_{t,i})}_{\texttt{meta-regret}} + \underbrace{\sum_{t=1}^{T} f_t(\mathbf{w}_{t,i}) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)}_{\texttt{base-regret}}$$

- $\texttt{meta-regret} \leq \mathcal{O}\left(\varepsilon V_T + \varepsilon \sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 + \frac{1 + P_T}{\varepsilon} - \frac{1}{\varepsilon}\sum_{t=2}^{T} \left\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\right\|_1^2\right)$ *negative term for self-cancellation*
- $\texttt{base-regret} \leq \mathcal{O}\left(\eta_i V_T + \eta_i \sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 + \frac{1}{\eta_i} - \frac{1}{\eta_i}\sum_{t=2}^{T} \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2\right)$ *only for a particular base learner, not sufficient for cancellation*

Stability Analysis

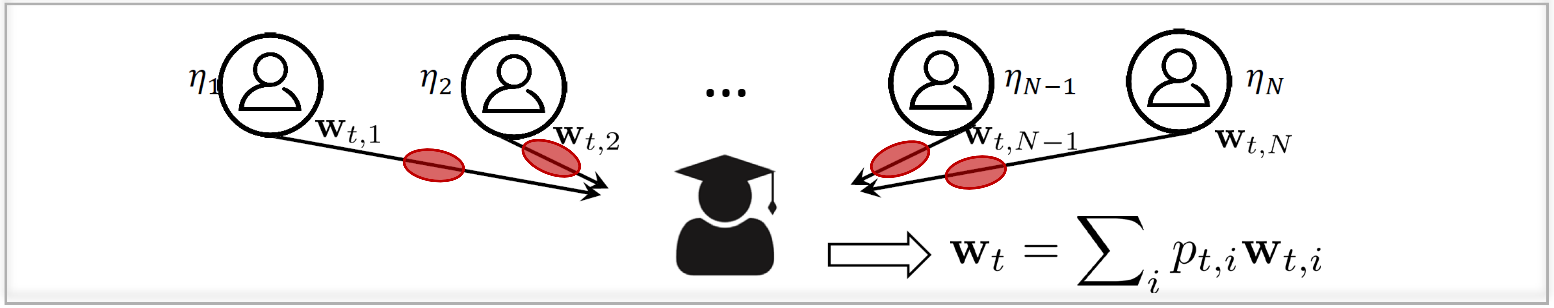
- Stability of the meta-base online ensemble

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{w}_{t+1,i} \Longrightarrow \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \leq \underbrace{2D^2 \left\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\right\|_1^2}_{\text{meta stability}} + \underbrace{2\sum_{i=1}^{N} p_{t,i} \left\|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\right\|_2^2}_{\text{weighted combine of base stability}}$$

meta stability *weighted combine of base stability*

- **Stablization**: meta algorithm $p_{t+1,i} \propto \exp\left(-\varepsilon(L_{t,i} + m_{t+1,i})\right)$ with
  - surrogate loss $\boldsymbol{\ell}_t \in \mathbb{R}^N$ with $\ell_{t,i} = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_{t,i}\rangle + \lambda \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2$;
  - hint prediction $\boldsymbol{m}_{t+1} \in \mathbb{R}^N$ with $m_{t+1,i} = \langle M_{t+1}, \mathbf{w}_{t+1,i}\rangle + \lambda \|\mathbf{w}_{t+1,i} - \mathbf{w}_{t,i}\|_2^2$.

***correction**: penalizing instable base learners*



Collaborative Online Ensemble

- Dynamic regret of the modified algorithm (***with corrections***):
  - $\texttt{meta-regret} \leq \mathcal{O}\left(\frac{1+P_T}{\varepsilon} + \varepsilon V_T + \varepsilon \sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 - \frac{1}{\varepsilon}\sum_{t=2}^{T} \|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1^2 + \frac{1}{\eta_i}\sum_{t=2}^{T} \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2 - \sum_{t=2}^{T}\sum_{i=1}^{N} \frac{1}{\eta_i} p_{t,i} \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2\right)$

    these two terms are due to correction

  - $\texttt{base-regret} \leq \mathcal{O}\left(\eta_i V_T + \frac{1}{\eta_i} + \eta_i \sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 - \frac{1}{\eta_i}\sum_{t=2}^{T} \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2\right)$

Collaborative Online Ensemble

$$\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \le 2D^2 \|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1^2 + 2\sum_{i=1}^{N} p_{t,i}\|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2$$

meta stability *weighted combine of base stability*

- Dynamic regret of the ... *ns*):

  - $\texttt{meta-regret} \le \mathcal{O}\Bigg(\frac{1+P_T}{\varepsilon} + \varepsilon V_T + \varepsilon\sum_{t=2}^{T}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 - \frac{1}{\varepsilon}\sum_{t=2}^{T}\|\boldsymbol{p}_t - \boldsymbol{p}_{t-1}\|_1^2 + \frac{1}{\eta_i}\sum_{t=2}^{T}\|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2 - \sum_{t=2}^{T}\sum_{i=1}^{N}\frac{1}{\eta_i}p_{t,i}\|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2\Bigg)$

    these two terms are due to correction

  - $\texttt{base-regret} \le \mathcal{O}\Bigg(\eta_i V_T + \frac{1}{\eta_i} + \eta_i\sum_{t=2}^{T}\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 - \frac{1}{\eta_i}\sum_{t=2}^{T}\|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2\Bigg)$

with suitable parameter configurations

$$\text{D-Regret}_T \le \mathcal{O}\left(\sqrt{V_T(1+P_T)}\right)$$

Collaborations **between meta and base learners:**
simultaneously exploiting

- ✷ *negative terms in the regret analysis*
- ✷ *correction terms in the algorithm design*

Online Ensemble for Gradient Variation

- An online ensemble to balance between non-stationarity and adaptivity.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i} \mathbf{w}_{t+1,i}$$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \mathcal{O}\left(\sqrt{V_T(1+P_T)}\right)$$

▶ **Multiple candidates:** to cover uncertainty

diversity consideration: cover all the possible range using as fewer as possible discretization items

$\eta_1 \quad \eta_2 \quad \eta_3 \quad \cdots \quad \eta_N$

$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \frac{D}{2G \ldots} \right.$

with $N = \lceil \log_2(GT/(8D^2 \ldots$

correction terms *enable **collaborations** between meta and base levels*

▶ **Base-learners:** each updated using $\eta_i \in \mathcal{H}$

$$\widehat{\mathbf{w}}_{t+1,i} = \Pi_{\mathcal{W}} \left[\widehat{\mathbf{w}}_{t,i} - \eta_i \nabla f_t(\mathbf{w}_t) \right]$$
$$\mathbf{w}_{t+1,i} = \Pi_{\mathcal{W}} \left[\widehat{\mathbf{w}}_{t+1,i} - \eta_i \nabla f_t(\mathbf{w}_t) \right].$$

▶ **Meta-algorithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

Hedge: $p_{t+1,i} \propto \exp\left(-\varepsilon(L_{t,i} + m_{t+1,i})\right), \ \forall i \in [N].$

- surrogate loss $\ell_{t,i} = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_{t,i} \rangle + \lambda \|\mathbf{w}_{t,i} - \mathbf{w}_{t-1,i}\|_2^2$;
- hint prediction $m_{t+1,i} = \langle M_{t+1}, \mathbf{w}_{t+1,i} \rangle + \lambda \|\mathbf{w}_{t+1,i} - \mathbf{w}_{t,i}\|_2^2$.

Summary of Our Results

- Full-information online learning

  gradient information is available to the learner

  [Zhang et al., NeurIPS'18; Zhao et al., NeurIPS'20; Zhao et al., NeurIPS'22; Zhao et al., JMLR'23]

- Partial-information online learning

  gradient information cannot be observed, only function value is available

  [Zhao et al., JMLR'21; Luo et al., COLT'22; Yan et al., JMLR'23]

- Decision-dependent online learning

  current decision will affect the future (incl. gradient & function value)

  [Zhao et al., ICML'22; Zhao et al., AISTAST'23; Li et al., NeurIPS'23]

Outline

- Problem Setup
- Non-stationary Online Learning
- Universal Online Learning
- Conclusion

OCO: classic methods

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

Function type	**Algorithm**	**Regret**
convex	Online Gradient Descent with $\eta_t \approx \frac{1}{\sqrt{t}}$	$\mathcal{O}(\sqrt{T})$
λ-strongly convex	Online Gradient Descent with $\eta_t = \frac{1}{\lambda t}$	$\mathcal{O}(\log T)$
α-exp-concave	Online Newton Step with α	$\mathcal{O}(d \log T)$

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

- **High-Level:** adaptive to ***unknown function curvatures***
- **Low-Level:** adaptive to ***unknown niceness of environments***

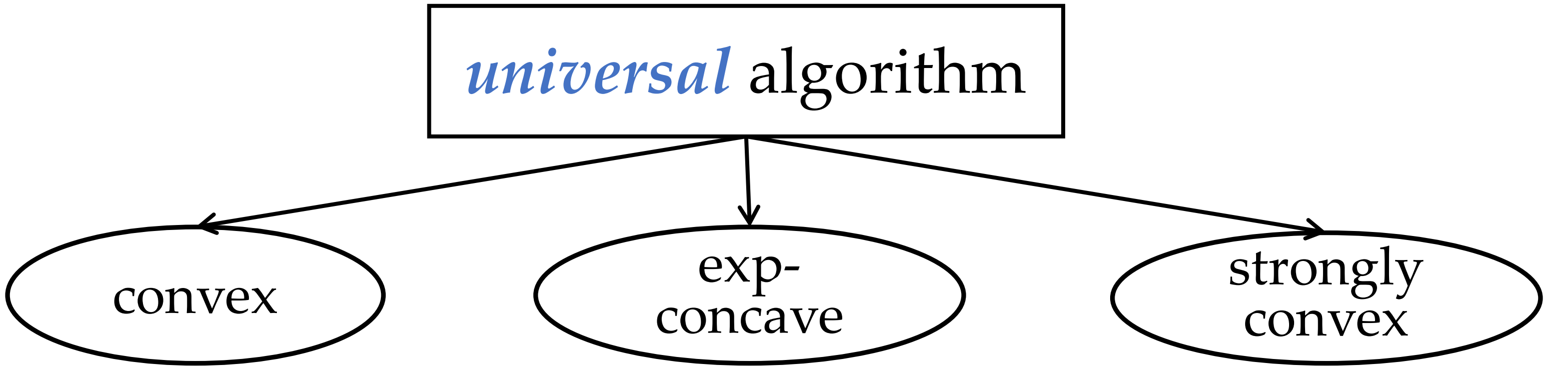
OCO: high-level adaptivity

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

  Universal method aims to develop a single algorithm for different families:

  (i) agnostic to the specific function curvature;

  (ii) while achieving the same regret as if they were known.



⇨ An algorithm achieves $\mathcal{O}(\sqrt{T})$, $\mathcal{O}(d \log T)$, and $\mathcal{O}(\log T)$ regret for convex/exp-concave/str. convex functions, respectively.

OCO: low-level adaptivity

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

Problem-dependent method aims to develop more adaptive bounds:

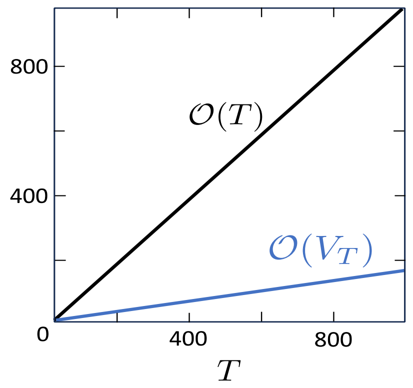
(i) regret guarantee can be substantially improved for easy environments ;

(ii) while can simultaneously safeguard the worst-case minimax rate.

Gradient variation:

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

measure the cumulative variations in gradients



⇨ Improved regret of $\mathcal{O}(\sqrt{V_T})$, $\mathcal{O}(d \log V_T)$, and $\mathcal{O}(\log V_T)$ can be attained for convex/exp-concave/str. convex functions, respectively (using different algorithms).

Guiding Question

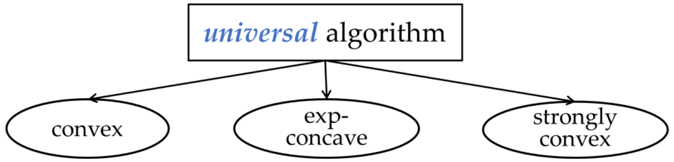
Is it possible to design an algorithm with two-level adaptivity?

i.e., universal to function curvature, and adaptive to gradient variations

OCO: high-level adaptivity

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

  Universal method aims to develop a single algorithm for different families:
  (i) agnostic to the specific function curvature;
  (ii) while achieving the same regret as if they were known.



⇨ An algorithm achieves $\mathcal{O}(\sqrt{T})$, $\mathcal{O}(d \log T)$, and $\mathcal{O}(\log T)$ regret for convex/exp-concave/str. convex functions, respectively.

Peng Zhao (Nanjing University) 1

OCO: low-level adaptivity

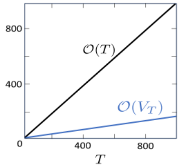
- **Low-Level:** adaptive to ***unknown niceness of environments***

  Problem-dependent method aims to develop more adaptive bounds:
  (i) regret guarantee can be substantially improved for easy environments;
  (ii) while can simultaneously safeguard the worst-case minimax rate.

Gradient variation:

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

measure the cumulative variations in gradients



⇨ Improved regret of $\mathcal{O}(\sqrt{V_T})$, $\mathcal{O}(d \log V_T)$, and $\mathcal{O}(\log V_T)$ can be attained for convex/exp-concave/str. convex functions, respectively (using different algorithms).

Peng Zhao (Nanjing University) 11

Main Result

- We provide an affirmative answer by providing the following result.

Theorem 1 (Yan-Z-Zhou; NeurIPS 2023). *Under standard assumptions, our algorithm ensures that*

- *it achieves* $\mathcal{O}(\log V_T)$ *regret for strongly convex functions;*
- *it achieves* $\mathcal{O}(d \log V_T)$ *regret for exp-concave functions;*
- *it achieves* $\widehat{\mathcal{O}}(\sqrt{V_T})$ *regret for convex functions.*

Here, $V_T = \sum_{t=2}^{T} \sup_{\mathbf{x} \in \mathcal{X}} \|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|^2$ *is gradient variation and* $\widehat{\mathcal{O}}(\cdot)$ *omits* $\log V_T$ *factors.*

A single algorithm with simultaneously near-optimal gradient-variation 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 minimax rate*
    - V_T denotes the variation in gradients that can be much smaller than $\mathcal{O}(T)$.
    - Gradient-variation regret bounds $\mathcal{O}(\log V_T)$, $\mathcal{O}(d \log V_T)$, and $\mathcal{O}(\sqrt{V_T})$ can recover the minimax rate of $\mathcal{O}(\log T)$, $\mathcal{O}(d \log T)$, and $\mathcal{O}(\sqrt{T})$.
  - *Implications in **Games** & **Stochastic Optimization***
    - Gradient variation bounds are essential for obtaining fast rates in games.
    - Gradient variation can bridge stochastic and adversarial optimization.

Implications: Games

- **Gradient Variation in *Games***: [Syrgkanis et al., NIPS'15]

Example: 

y-player decision $\mathbf{y}_t = (1/2\ 1/2\ 0)^\top$

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

Implications: Games

- **Gradient Variation in *Games***: [Syrgkanis et al., NIPS'15]

Online Game Protocol

The environments decide a payoff matrix A

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

- x-player submits $\mathbf{x}_t \in \Delta_d$ and y-player submits $\mathbf{y}_t \in \Delta_d$
- the x-player suffers loss $\mathbf{x}_t^\top A \mathbf{y}_t$ and receives gradient $A\mathbf{y}_t$, the y-player receives reward $\mathbf{x}_t^\top A \mathbf{y}_t$ and receives gradient $A\mathbf{x}_t$

Gradient-variation online learning plays an important role in games.

Implications: Games

Deploying *gradient-variation algorithm* (e.g., online mirror descent with last-round gradient) attains:

$$f_t^x(\mathbf{x}) \triangleq \mathbf{x}^\top A \mathbf{y}_t$$

$$f_{t-1}^x(\mathbf{x}) \triangleq \mathbf{x}^\top A \mathbf{y}_{t-1}$$

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

gradient variation *negative stability*

Deploying *gradient-variation algorithm* (e.g., online mirror descent with last-round gradient) attains:

$$f_t^y(\mathbf{y}) \triangleq \mathbf{x}_t^\top A \mathbf{y}$$

$$f_{t-1}^y(\mathbf{y}) \triangleq \mathbf{x}_{t-1}^\top A \mathbf{y}$$

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

gradient variation *negative stability*

***Regret summation** is usually related to some global performance measures in games, such as Nash equilibrium regret and duality gap.*

Implications: Games

Deploying *gradient-variation algorithm* (e.g., online mirror descent with last-round gradient) attains:

$$f_t^x(\mathbf{x}) \triangleq \mathbf{x}^\top A\mathbf{y}_t$$

$$f_{t-1}^x(\mathbf{x}) \triangleq \mathbf{x}^\top A\mathbf{y}_{t-1}$$

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

Deploying *gradient-variation algorithm* (e.g., online mirror descent with last-round gradient) attains:

$$f_t^y(\mathbf{y}) \triangleq \mathbf{x}_t^\top A\mathbf{y}$$

$$f_{t-1}^y(\mathbf{y}) \triangleq \mathbf{x}_{t-1}^\top A\mathbf{y}$$

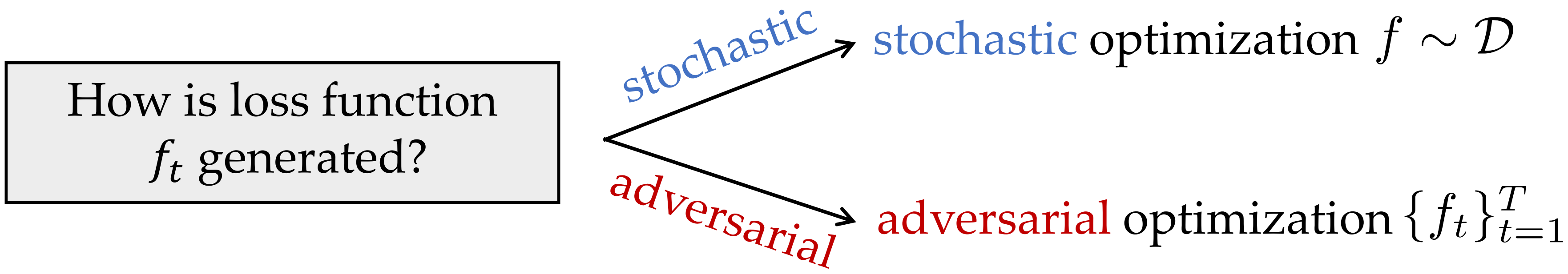
$$\mathrm{Reg}_T^y \lesssim 1 + \underbrace{\sum_{t=2}^T \|\mathbf{x}_t^\top A - \mathbf{x}_{t-1}^\top A\|_\infty^2}_{\text{gradient variation}} \underbrace{- \sum_{t=2}^T \|\mathbf{y}_t - \mathbf{y}_{t-1}\|_1^2}_{\text{negative stability}}$$

$\Rightarrow \mathrm{Reg}_T^x + \mathrm{Reg}_T^y \le \mathcal{O}(1)$ *which is essential for the* $\mathcal{O}\left(\frac{1}{T}\right)$ *fast rate in games.*

Implications: Stochastic Opt.

- **Gradient Variation in *Stochastic/Adversarial Optimization* :**

[Sachs et al., NeurIPS'22]



➢ The studies on these two fields are previously ***separate***.

➢ Recent works reveal the essential role of ***gradient variation***, which provides an important interpolation between stochastic and adversarial optimization.

Implications: Stochastic Opt.

- **SEA (Stochastically Extended Adversarial) model** [Sachs et al., NeurIPS'22]

  Setup: at round $t \in [T]$, SEA optimizes $\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$

  f_t is the *randomized function* sampled from underlying distribution $\mathcal{D}_t$: $f_t \sim \mathcal{D}_t$

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

Two crucial complexity measures:

$$\sigma_{1:T}^2 \triangleq \sum_{t=1}^{T} \max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{f_t\sim\mathcal{D}_t}[\|\nabla f_t(\mathbf{x}) - \nabla F_t(\mathbf{x})\|^2], \quad \Sigma_{1:T}^2 \triangleq \mathbb{E}\left[\sum_{t=2}^{T} \sup_{\mathbf{x}\in\mathcal{X}} \|\nabla F_t(\mathbf{x}) - \nabla F_{t-1}(\mathbf{x})\|^2\right]$$

stochastic change *adversarial change*

Implications: Stochastic Opt.

- **SEA (Stochastically Extended Adversarial) model** [Sachs et al., NeurIPS'22]

Setup: at round $t \in [T]$, SEA optimizes $\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x})$

f_t is the *randomized function* sampled from underlying distribution $\mathcal{D}_t$: $f_t \sim \mathcal{D}_t$

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

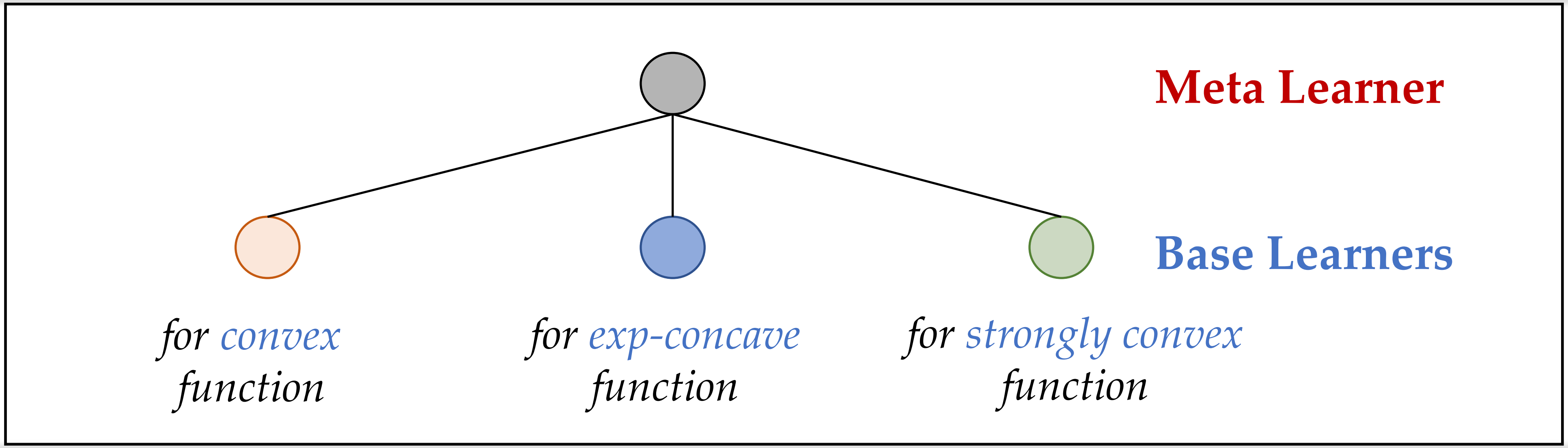
⇨ SEA model can be solved by deploying gradient-variation algorithm over the randomized function $\{f_t\}_{t=1}^T$.

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

Approximately $V_T \approx \sigma_{1:T}^2 + \Sigma_{1:T}^2$. For stochastic optimization, $\sigma_{1:T}^2 = \sigma^2 T$ and $\Sigma_{1:T}^2 = 0$. For adversarial optimization, $\sigma_{1:T}^2 = 0$ and $\Sigma_{1:T}^2 = V_T$.

Approach

- **Basic idea: Online Ensemble** $\mathbf{x}_t = \sum_{i=1}^{N} p_{t,i}\mathbf{x}_{t,i}$
  - $\boldsymbol{p}_t = [p_{t,1}, \ldots, p_{t,N}]^\top$ is the meta weight;
  - $\{\mathbf{x}_{t,i}\}_{t=1}^{T}$ is the base decisions of the i-th base learners, $i \in [N]$.



also used in non-stationary online learning (for dynamic/adaptive regret minimization)

Approach

- **Regret decomposition:** how to control meta-regret in two layers

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

meta regret *base regret*

- **Key idea:** exploiting the ***second-order regret bound*** on the meta level

[Zhang et al., ICML'22]

$$\sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \sum_{t=1}^{T} \ell_{t,i} \leq \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} r_{t,i}^2}\right)$$

(second-order bound, e.g., Adapt-ML-Prod) [Gaillard et al, COLT'14]

⇒ $\ell_{t,i} \triangleq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_{t,i} \rangle$

$r_{t,i} \triangleq \langle p_t, \ell_t \rangle - \ell_{t,i}$

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

Approach

- **Regret decomposition:** how to control meta-regret in two layers

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

- **Key idea:** exploiting the ***second-order regret bound*** on the meta level

$\Rightarrow$

$$\ell_{t,i} \triangleq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_{t,i}\rangle \qquad r_{t,i} \triangleq \langle p_t, \ell_t\rangle - \ell_{t,i}$$

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

e.g., **exp-concave**

$\Rightarrow$

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

Approach

- **Regret decomposition:** how to control meta-regret in two layers

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

- **Key idea:** exploiting the ***second-order regret bound*** on the meta level

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

$$r_{t,i} \triangleq \langle p_t, \ell_t\rangle - \ell_{t,i}$$

e.g., **strongly convex**

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

Approach

- **Regret decomposition:** how to control meta-regret in two layers

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

- **Key idea:** exploiting the ***second-order regret bound*** on the meta level

$\Rightarrow$

$$\ell_{t,i} \triangleq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_{t,i}\rangle$$
$$r_{t,i} \triangleq \langle p_t, \ell_t\rangle - \ell_{t,i}$$

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

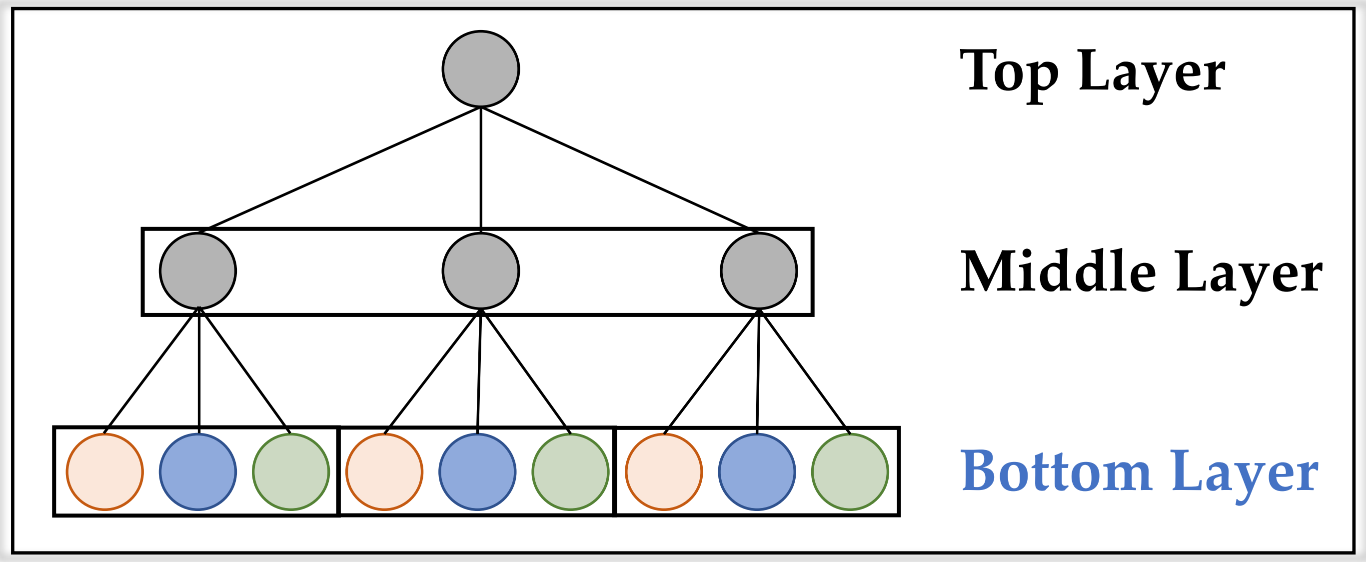
e.g., **convex**

$\Rightarrow$

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

Achieving Two-Level Adaptivity

- **Multi-layer Online Ensemble**



- ➢ **Top layer & Middle layer:**

  a ***two-layer meta learner***

- ➢ **Bottom layer:**

  basic online ensemble idea

Why three layers? (mostly due to the technical reasons)

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

Key Ingredients

- **Ingredient I:** novel ***optimism*** to reuse historical gradients ***universally***

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

Recall meta regret: $\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_{t,i^\star})$

we optimize the linearized regret: $\sum_{t=1}^T r_{t,i^\star} \triangleq \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_{t,i^\star} \rangle$

Optimistic-Adapt-ML-Prod: $$\sum_{t=1}^T r_{t,i^\star} \leq \mathcal{O}\left(\sqrt{\sum_{t=1}^T (r_{t,i^\star} - m_{t,i^\star})^2}\right)$$

[Wei et al., NIPS'16]

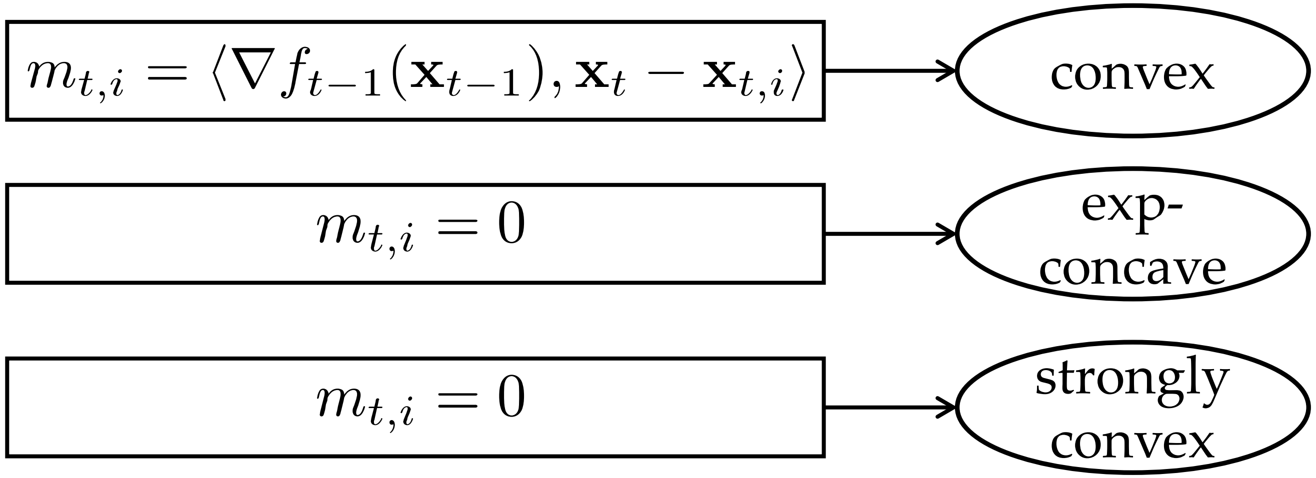
optimism

Key Ingredients

- **Ingredient I:** novel ***optimism*** to reuse historical gradients ***universally***

Goal: to ensure an $\mathcal{O}(1)$ meta regret for ***exp-concave/strongly convex*** functions, and $\mathcal{O}(\sqrt{V_T})$ meta regret for ***convex*** functions.

Challenge: can only use ***separate*** parameters to act as the optimism



different parameters for different functions (not universal)

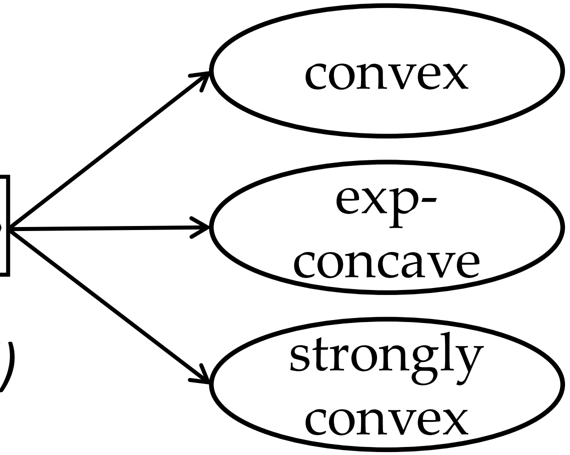
Key Ingredients

- **Ingredient I:** novel ***optimism*** to reuse historical gradients ***universally***

Our solution:

universal parameter

$$m_{t,i} = r_{t-1,i} = \langle \nabla f_{t-1}(\mathbf{x}_{t-1}), \mathbf{x}_{t-1} - \mathbf{x}_{t-1,i} \rangle$$



one parameter for different functions (universal)

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

algorithm stability

Key Ingredients

- **Ingredient II:** ***collaboration*** in multiple layers to handle the ***stability***

Goal: to ensure the stability $\sum_{t=2}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2$ can be handled by the negative regret within the dynamics of online ensemble.

Two layers:

[Zhao et al, 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

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

Key Ingredients

- **Ingredient II:** ***collaboration*** **in multiple layers to handle the** ***stability***

Goal: to ensure the stability $\sum_{t=2}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2$ can be handled by the negative regret within the dynamics of online ensemble.

Three layers:

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

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

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

Top Layer

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

Middle Layer

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

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.

Algorithm

Algorithm 1 Universal OCO with Gradient-variation Guarantees

Input: Curvature coefficient pool $\mathcal{H}$, MsMwC-Mid number K, base learner number N

Initialize: Top layer: $\mathcal{A}^{\text{top}}$ — MsMwC-Top with $\eta_k = (C_0 \cdot 2^k)^{-1}$ and $\widehat{q}_{1,k} = \eta_k^2 / \sum_{k=1}^K \eta_k^2$
Middle layer: $\{\mathcal{A}_k^{\text{mid}}\}_{k \in [K]}$ — MsMwC-Mid with step size $2\eta_k$ and $\widehat{p}_{1,k,i} = 1/N$
Bottom layer: $\{\mathcal{B}_{k,i}\}_{k \in [K], i \in [N]}$ — base learners as specified in Section 2

for $t = 1$ **to** T **do**
  Receive $\mathbf{x}_{t,k,i}$ from $\mathcal{B}_{k,i}$, obtain $\mathbf{x}_{t,k} = \sum_i p_{t,k,i} \mathbf{x}_{t,k,i}$ and submit $\mathbf{x}_t = \sum_k q_{t,k} \mathbf{x}_{t,k}$
  Suffer $f_t(\mathbf{x}_t)$ and observe the gradient information $\nabla f_t(\cdot)$
  Construct $(\boldsymbol{\ell}_t, \boldsymbol{m}_t)$ (3.3) for $\mathcal{A}^{\text{top}}$ and $(\boldsymbol{\ell}_{t,k}, \boldsymbol{m}_{t,k})$ (3.4) for $\mathcal{A}_k^{\text{mid}}$
  $\mathcal{A}^{\text{top}}$ updates to $\boldsymbol{q}_{t+1}$ and $\mathcal{A}_k^{\text{mid}}$ updates to $\boldsymbol{p}_{t+1,k}$
  Multi-Gradient Feedback Model:
    Send gradient $\nabla f_t(\cdot)$ to $\mathcal{B}_{k,i}$ for update ▷ $\mathcal{O}(\log^2 T)$ gradient queries
  One-Gradient Feedback Model:
    Construct surrogates $h_{t,i}^{\text{sc}}(\cdot)$, $h_{t,i}^{\text{exp}}(\cdot)$, $h_{t,i}^{\text{c}}(\mathbf{x})$ using only $\nabla f_t(\mathbf{x}_t)$
    Send the surrogate functions to $\mathcal{B}_{k,i}$ for update ▷ Only *one* gradient query
end for

Main Result

- The first ***universal*** algorithm with near-optimal ***gradient-variation regret***.

Theorem 1 (Yan-Z-Zhou; NeurIPS 2023). *Under standard assumptions, our algorithm ensures that*

- *it achieves* $\mathcal{O}(\log V_T)$ *regret for strongly convex functions;*
- *it achieves* $\mathcal{O}(d \log V_T)$ *regret for exp-concave functions;*
- *it achieves* $\widehat{\mathcal{O}}(\sqrt{V_T})$ *regret for convex functions.*

Here, $V_T = \sum_{t=2}^{T} \sup_{\mathbf{x} \in \mathcal{X}} \|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|^2$ *is gradient variation and* $\widehat{\mathcal{O}}(\cdot)$ *omits* $\log V_T$ *factors.*

Immediate implications to game theory and SEA model.

Result for SEA

- **Stochastically Extended Adversarial (SEA)** [Sachs et al., NeurIPS'22]

Interpolation between stochastic and adversarial online convex optimization

Two crucial complexity measures:

$$\sigma_{1:T}^2 \triangleq \sum_{t=1}^{T} \max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{f_t\sim\mathcal{D}_t}[\|\nabla f_t(\mathbf{x}) - \nabla F_t(\mathbf{x})\|^2], \quad \Sigma_{1:T}^2 \triangleq \mathbb{E}\left[\sum_{t=2}^{T} \sup_{\mathbf{x}\in\mathcal{X}} \|\nabla F_t(\mathbf{x}) - \nabla F_{t-1}(\mathbf{x})\|^2\right]$$

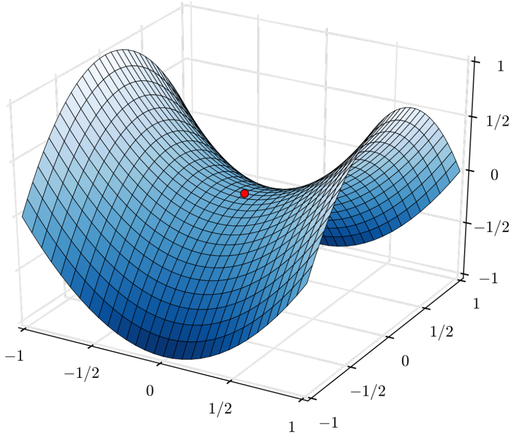
stochastic change *adversarial change*

Theorem 2. *Under standard assumptions, our algorithm obtains* $\mathcal{O}((\sigma_{\max}^2 + \Sigma_{\max}^2)\log(\sigma_{1:T}^2 + \Sigma_{1:T}^2))$ *regret for strongly convex functions,* $\mathcal{O}(d\log(\sigma_{1:T}^2 + \Sigma_{1:T}^2))$ *regret for exp-concave functions and* $\widehat{\mathcal{O}}(\sqrt{(\sigma_{1:T}^2 + \Sigma_{1:T}^2)})$ *regret for convex functions.*

Result for Games

- **Min-Max Optimization**

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y})$$



Consider two aspects:

(i) curvatures: f is bilinear/strongly convex-concave

(ii) honest: all players run the same algo; dishonest: otherwise (some may disobey)

Theorem 3. *Under standard assumptions, for bilinear and strongly convex-concave games, our algorithm enjoys $\mathcal{O}(1)$ regret summation in the honest case, $\widehat{\mathcal{O}}(\sqrt{T})$ and $\mathcal{O}(\log T)$ bounds respectively in the dishonest case.*

Conclusion

- **Online Ensemble**: an effective theoretical framework (base learners; meta learners; schedule) to handle ***uncertainty*** in online environments
- **Non-stationary online learning**: online ensemble for dynamic regret
  - bandit convex optimization: ***surrogate loss*** is essential to exploit limited feedback
  - problem-dependent guarantee: incorporating ***hint prediction***, enable ***collaboration*** between meta and base layers (via negative terms and corrections)
- **Universal online learning**: online ensemble adaptive to curvatures
  - gradient-variation universal regret: ***multi-layer corrections***, ***unifying optimism***
  - applications to SEA model, games, etc.
- Many todo: efficiency/real-time response? non-convexity? continual learning? …

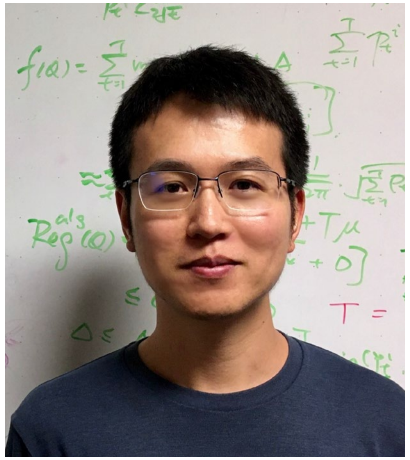
Thanks!

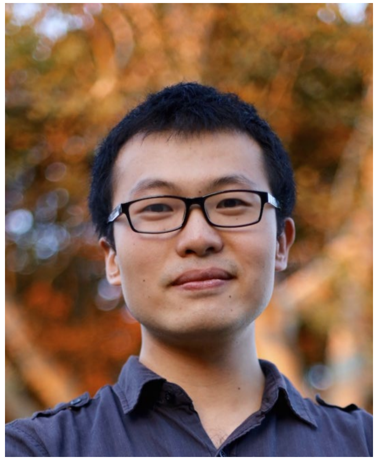
Collaborators

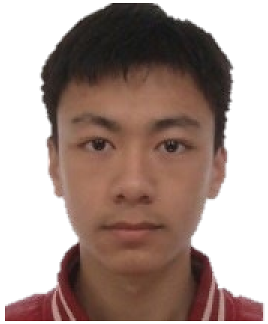

Zhi-Hua Zhou (NJU)


Lijun Zhang (NJU)

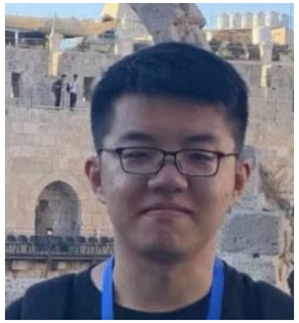

M. Sugiyama (U. Tokyo)


Haipeng Luo (USC)


Yu-Xiang Wang (UCSB)


Yu-Jie Zhang


Mengxiao Zhang


Guanghui Wang


Yu-Hu Yan


Long-Fei Li


Yong Bai


Yan-Feng Xie

References

- Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and Non-stationarity: Problem-dependent Dynamic Regret for Online Convex Optimization. Arxiv: 2112.14368, 2021. **(major ref for non-stationary online learning)**
- Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. Universal Online Learning with Gradient Variations: A Multi-layer Online Ensemble Approach. **NeurIPS 2023**. **(major ref for universal online learning)**
- Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit Convex Optimization in Non-stationary Environments. **JMLR 2021**.
- Peng Zhao, Yu-Hu Yan, Yu-Xiang Wang, Zhi-Hua Zhou. Non-stationary Online Learning with Memory and Non-stochastic Control. **JMLR 2023**.

References

- Peng Zhao, Yan-Feng Xie, Lijun Zhang, and Zhi-Hua Zhou. Efficient Methods for Non-stationary Online Learning. **NeurIPS 2022**.
- Haipeng Luo, Mengxiao Zhang, Peng Zhao, and Zhi-Hua Zhou. Corralling a Larger Band of Bandits: A Case Study on Switching Regret for Linear Bandits. **COLT 2022**.
- Lijun Zhang, Guanghui Wang, Jinfeng Yi, and Tianbao Yang. A simple yet universal strategy for online convex optimization. **ICML 2022**.
- Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Impossible tuning made possible: A new expert algorithm and its applications. COLT 2021.

References

- Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E. Schapire. Fast convergence of regularized learning in games. **NIPS 2015**.
- Pierre Gaillard, Gilles Stoltz, and Tim van Erven. A second-order bound with excess losses. **COLT 2014**.
- Sarah Sachs, Hédi Hadiji, Tim van Erven, and Cristóbal Guzmán. Between stochastic and adversarial online convex optimization: Improved regret bounds via smoothness. **NeurIPS 2022**.
- Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Tracking the best expert in non-stationary stochastic environments. **NIPS 2016**.