



Lecture 1. Online Learning/Optimization

Peng Zhao

School of AI

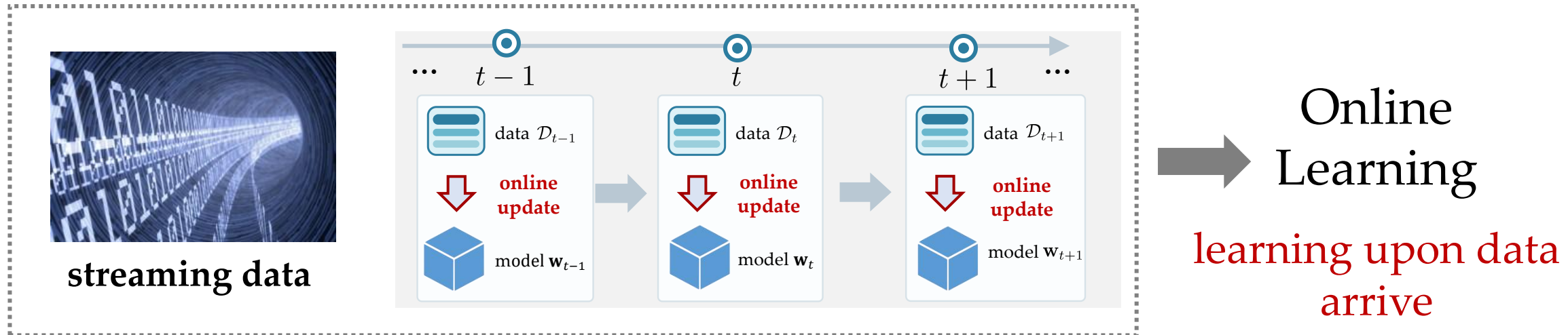
Nanjing University

July 14, 2026

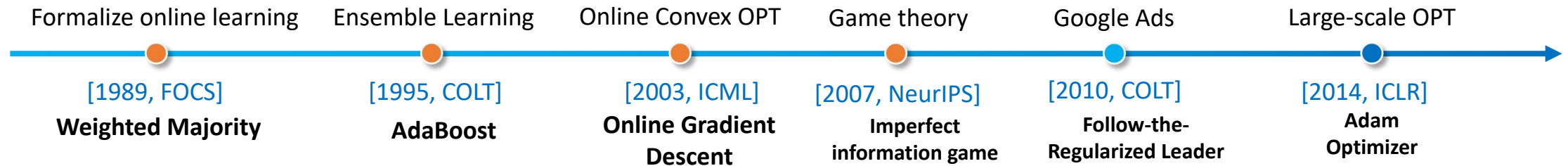


Online Learning/Optimization

- Online learning: for scenarios where data accumulates continuously



- Fundamental role in ML and its connections to many disciplines





Online Learning/Optimization in ML

Best paper award in top ML conferences (like ICML/NeurIPS/COLT)

- Optimal and Adaptive Algorithms for **Online Boosting**. ICML 2015
- Fast **Convergence** of Regularized Learning in **Games**. NIPS 2015
- Multi-scale Exploration of Convex Functions and **Bandit** Convex Optimization. COLT 2016
- **No-Regret** Learning Dynamics for Extensive-Form Correlated **Equilibrium**. NeurIPS 2020
- Bayesian Design Principles for Frequentist **Sequential Learning**. ICML 2023
- Adapting to Game Trees in Zero-Sum Imperfect Information **Games**. ICML 2023
- Optimal Mistake Bounds for Transductive **Online Learning**. NeurIPS 2025 runner up



Online Learning/Optimization in ML

Test of time award in top ML conferences (like ICML/NeurIPS)

- The Weighted Majority Algorithm. FOCS 1989 [*FOCS 2019 30-Years Test of Time*]
- Online Convex Programming and Generalized Infinitesimal Gradient Ascent. ICML 2003 [*ICML 2013 Test of Time*]
- Dual Averaging Method for Regularized Stochastic Learning and Online Optimization. NIPS 2009 [*NeurIPS 2019 Test of Time*]
- Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. ICML 2010 [*ICML 2020 Test of Time*]
- Adam: A Method for Stochastic Optimization [*ICLR 2025 test of time*]

Online Learning/Optimization in ML

- Optimizers for (Deep) Learning

Adaptive subgradient methods for online learning and stochastic optimization.	16868	2011
J Duchi, E Hazan, Y Singer Journal of machine learning research 12 (7)		

Adam: A method for stochastic optimization	256152	2015
D Kingma, J Ba International Conference on Learning Representations		

till 2026.07.13



Test of Time

Adam: A Method for Stochastic Optimization
Diederik P. Kingma, Jimmy Ba
<https://arxiv.org/abs/1412.6980>



This mini-course covers

- **Lecture 1: Online Convex Optimization**

Convex optimization basics, Regret minimization, online gradient descent, convex and strongly convex losses, online-to-batch conversion, and connections to SGD.

- **Lecture 2: Online Mirror Descent**

Mirror maps, Bregman divergence, OMD, FTRL, multiplicative weights, and KL/entropy geometry.

- **Lecture 3: Stochastic Multi-Armed Bandits**

Exploration-exploitation tradeoff, Explore-Then-Commit, UCB, Optimism in the face of uncertainty

- **Lecture 4: Linear Bandits and Contextual Bandits**

Contextual bandit, linear bandits, LinUCB, confidence ellipsoids, self-normalized concentration, and generalized linear/logistic bandits.

- **Lecture 5: From Bandits to Reinforcement Learning**

Finite-horizon stochastic MDPs, UCB Value Iteration, linear function approximation in MDPs, linear MDP, linear mixture MDP, policy optimization, policy gradient.

Course Website

Course Slides will be uploaded to

<https://www.pengzhao-ml.com/course/OnlineLearning2026Summer>

More materials can be found at AOPT courses

- [Advanced Optimization \(For Graduate Students, 2025 Fall\)](#)
- [Advanced Optimization \(For Graduate Students, 2024 Fall\)](#)
- [Advanced Optimization \(For Undergraduate and Graduate Students, 2023 Fall\)](#)
- [Advanced Optimization \(For Undergraduate and Graduate Students, 2022 Fall\)](#)



Outline

- Optimization Basics
- Online Learning
- Online Convex Optimization
- Online-to-Batch Conversion



Part 1. Optimization Basics

- Convex Optimization
- Function Properties
- Gradient Descent
- Convergence Rate

Optimization Problem

- We adopt a minimization language

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}\end{array}$$

- optimization variable $\mathbf{x} \in \mathbb{R}^d$
- objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$: convex and continuously differentiable
- feasible domain $\mathcal{X} \subseteq \mathbb{R}^d$: convex

Performance Measure

- Iterated Optimization

$$\mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \cdots \mathbf{x}_t \rightarrow \mathbf{x}_{t+1} \rightarrow \cdots \mathbf{x}_T$$

To output a sequence $\{\bar{\mathbf{x}}_t\}_{t=1}^T$ such that $\bar{\mathbf{x}}_t$ **approximates** $\mathbf{x}^*$ when t goes larger, where $\{\bar{\mathbf{x}}_t\}_{t=1}^T$ can be *statistics* of the original sequence $\{\mathbf{x}_t\}_{t=1}^T$, and $\mathbf{x}^* \arg \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ denotes the optimal solution.

- Measure

- Function-value level: $f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) \leq \varepsilon(T)$
- Optimizer-value level: $\|\bar{\mathbf{x}}_T - \mathbf{x}^*\| \leq \varepsilon(T)$

and $\varepsilon(T)$ is the *approximation error* and is a function of iterations T .

Performance Measure

- In general, there are two performance measures (essentially same).

Convergence: $f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) \leq \varepsilon(T)$,

- **Qualitatively:** $\varepsilon(T) \rightarrow 0$ when $T \rightarrow \infty$
- **Quantitatively:** $\mathcal{O}(\frac{1}{\sqrt{T}})$ / $\mathcal{O}(\frac{1}{T})$ / $\mathcal{O}(\frac{1}{T^2})$ / $\mathcal{O}(\frac{1}{e^T})$ / ...

Complexity:

- **Definition:** number of iterations required to achieve $f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) \leq \varepsilon$.
- **Quantitatively:** $\mathcal{O}(\frac{1}{\varepsilon^2})$ / $\mathcal{O}(\frac{1}{\varepsilon})$ / $\mathcal{O}(\frac{1}{\sqrt{\varepsilon}})$ / $\mathcal{O}(\log \frac{1}{\varepsilon})$ / ...

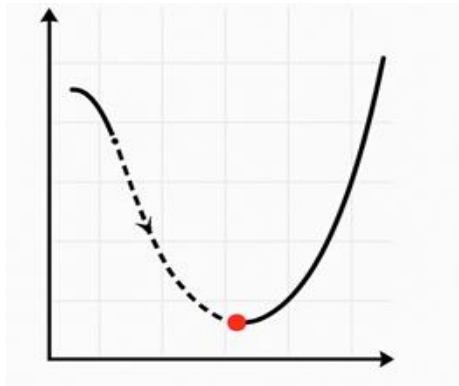
corresponds to $\mathcal{O}(\frac{1}{\sqrt{T}})$ / $\mathcal{O}(\frac{1}{T})$ / $\mathcal{O}(\frac{1}{T^2})$ / $\mathcal{O}(\frac{1}{e^T})$ / ...

Function Property

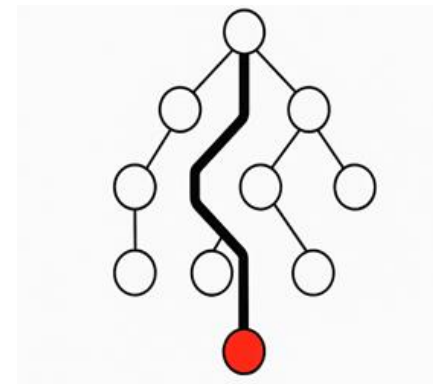
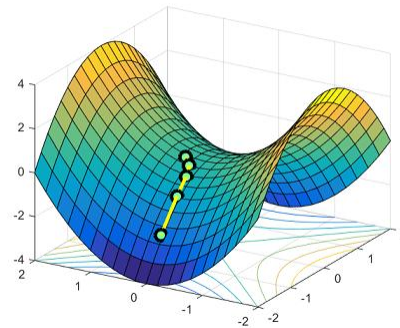
- Function mapping $f : \text{dom } f \subseteq \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathcal{Y} \subseteq \mathbb{R}^m$

Definition 1 (Continuity). A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuous at $\mathbf{x} \in \text{dom } f$ if for all $\epsilon > 0$ there exists a $\delta > 0$ with $\mathbf{y} \in \text{dom } f$, such that

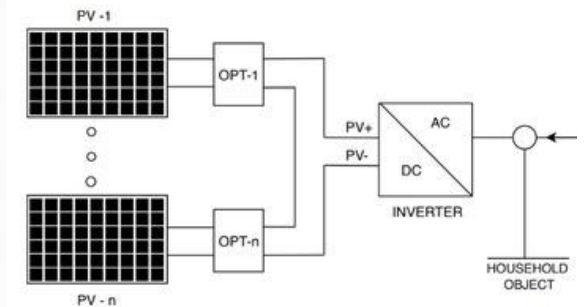
$$\|\mathbf{y} - \mathbf{x}\|_2 \leq \delta \Rightarrow \|f(\mathbf{y}) - f(\mathbf{x})\|_2 \leq \epsilon.$$



Continuous Optimization



Discrete Optimization





Lipschitz Continuity

Definition 1 (Continuity). A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuous at $\mathbf{x} \in \text{dom } f$ if for all $\epsilon > 0$ there exists a $\delta > 0$ with $\mathbf{y} \in \text{dom } f$, such that

$$\|\mathbf{y} - \mathbf{x}\|_2 \leq \delta \Rightarrow \|f(\mathbf{y}) - f(\mathbf{x})\|_2 \leq \epsilon.$$

Definition 2 (Lipschitz Continuity). A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is G -Lipschitz-continuous if for all $\mathbf{x}, \mathbf{y} \in \text{dom } f$,

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq G \|\mathbf{x} - \mathbf{y}\|.$$

Smoothness

Definition 3 (Smoothness). A function f is L -smooth with respect to the $\|\cdot\|$ norm if, for any $\mathbf{x}, \mathbf{y} \in \text{dom } f$,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L\|\mathbf{x} - \mathbf{y}\|.$$

Smoothness is also called *gradient Lipschitz* in many literature.

Smoothness is defined over the primal-dual norms, which become ℓ_2 -norm when specialized to Euclidean space (and then, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2$).

Smoothness

The next lemma is an *equivalent* condition of smoothness.

Lemma 1 (Descent Lemma). *Let f be an L -smooth function over a given convex set $\mathcal{X}$. Then for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|^2.$$

Proof: $f(\mathbf{y}) - f(\mathbf{x}) = \int_0^1 \langle \nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})), \mathbf{y} - \mathbf{x} \rangle dt$ (calculus)

$$\Rightarrow f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle = \int_0^1 \langle \nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) - \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle dt$$

$$\text{(Cauchy-Schwarz)} \leq \int_0^1 \|\nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) - \nabla f(\mathbf{x})\| \|\mathbf{y} - \mathbf{x}\| dt$$

$$\text{(smoothness)} \leq L \|\mathbf{y} - \mathbf{x}\|^2 \int_0^1 t dt \leq \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|^2$$

□

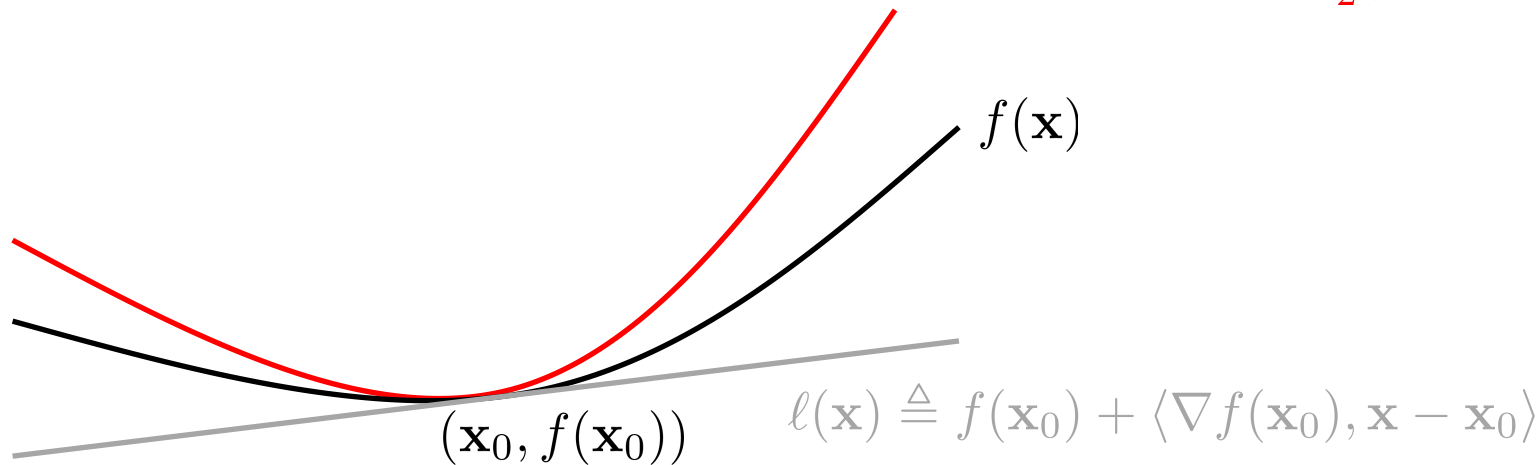
Smoothness

The next lemma is an *equivalent* condition of smoothness.

Lemma 1 (Descent Lemma). *Let f be an L -smooth function over a given convex set $\mathcal{X}$. Then for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|^2.$$

$$h(\mathbf{x}) \triangleq f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}_0\|_2^2$$



In Optimization theory:
smooth vs nonsmooth
“平滑” vs “非平滑”

Strong Convexity

Definition 4 (Strong Convexity). A function f is σ -strongly convex with respect to norm $\|\cdot\|$ if, for any $\mathbf{x}, \mathbf{y} \in \text{dom } f$ and $\lambda \in [0, 1]$,

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}) - \frac{\sigma}{2} \lambda (1 - \lambda) \|\mathbf{x} - \mathbf{y}\|^2.$$

- Clearly, for generally convex functions, $\sigma = 0$.

Examples:

- $f(\mathbf{x}) = \|\mathbf{x}\|_p^2$ is 2-strongly-convex with respect to norm $\|\cdot\|_p$.
- Negative entropy $f(\mathbf{x}) = \sum_{i=1}^d x_i \ln x_i$ over probability distribution (i.e., $x_i \in [0, 1]$ and $\sum_{i=1}^d x_i = 1$) is 1-strongly-convex with respect to norm $\|\cdot\|_1$.

Strong Convexity

Theorem 1. *Let $\mathcal{X}$ be a Euclidean space. Then f is σ -strongly convex with respect to norm $\|\cdot\|$ if and only if the function $f(\cdot) - \frac{\sigma}{2}\|\cdot\|^2$ is convex.*

f is “as least as convex” as a quadratic function.

Theorem 2. *Let f be a proper closed and σ -strongly convex function. Then for any $\mathbf{x} \in \text{dom}(\partial f)$, $\mathbf{y} \in \text{dom}(f)$ and $\mathbf{g} \in \partial f(\mathbf{x})$,*

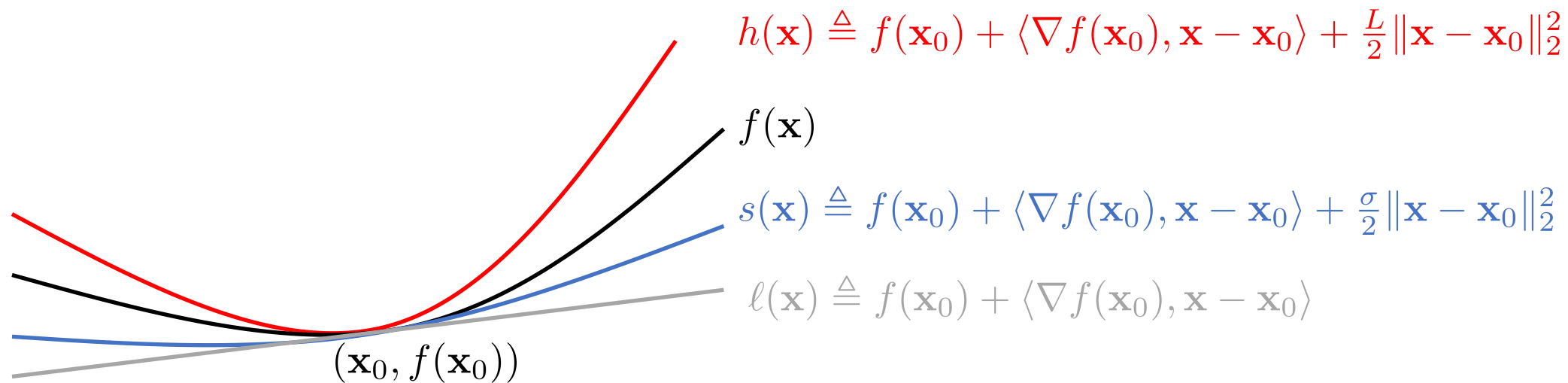
$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle + \frac{\sigma}{2} \|\mathbf{y} - \mathbf{x}\|^2.$$

Strongly Convex and Smooth

If function f is both σ -strongly convex and L -smooth w.r.t. ℓ_2 -norm, then

- $\sigma I \preceq \nabla^2 f(\mathbf{x}) \preceq LI$
- f is γ -well-conditioned with $\gamma = \kappa^{-1}$, where $\kappa \triangleq L/\sigma \geq 1$ is called the *condition number*.

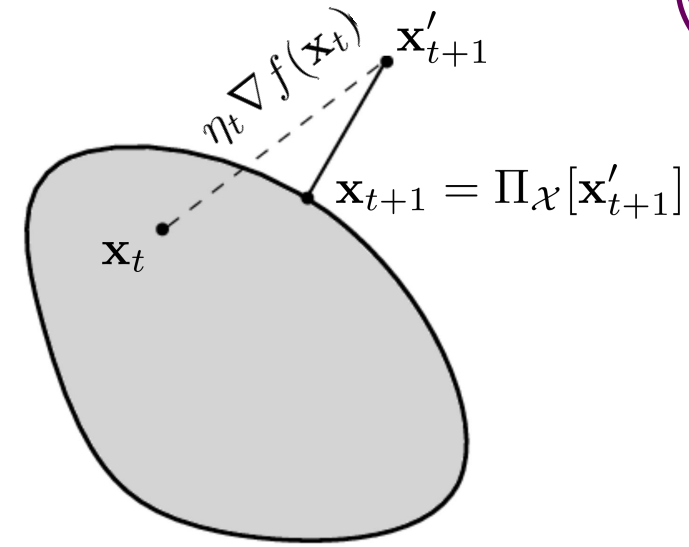
The smaller the condition number κ is, the easier the function is.



Gradient Descent

- GD Template:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)]$$



- $\mathbf{x}_1$ can be an arbitrary point inside the domain.
- $\eta_t > 0$ is the potentially time-varying *step size* (or called *learning rate*).
- Projection $\Pi_{\mathcal{X}}[\mathbf{y}] = \arg \min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$ ensures the feasibility.

Why Gradient Descent?

- For simplicity, we consider the *unconstrained* setting.

- **A General Idea:** *Surrogate Optimization*

We aim to find a sequence of *local upper bounds* $U_1, \dots, U_T$, where the surrogate function $U_t : \mathbb{R}^d \mapsto \mathbb{R}$ may depend on $\mathbf{x}_t$ such that

- (i) $f(\mathbf{x}_t) = U_t(\mathbf{x}_t)$;
- (ii) $f(\mathbf{x}) \leq U_t(\mathbf{x})$ holds for all $\mathbf{x} \in \mathbb{R}^d$;
- (iii) $U_t(\mathbf{x})$ should be simple enough to minimize.

⇒ Then, our proposed algorithm would be $\mathbf{x}_{t+1} = \arg \min_{\mathbf{x}} U_t(\mathbf{x})$

Why Gradient Descent?

- Following the *surrogate optimization* principle, let's invent GD for convex and *smooth* functions.

Proposition 1. Suppose that f is convex and differentiable. Moreover, suppose that f is L -smooth with respect to ℓ_2 -norm. Define the surrogate $U_t : \mathbb{R}^d \mapsto \mathbb{R}$ as

$$U_t(\mathbf{x}) \triangleq f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2.$$

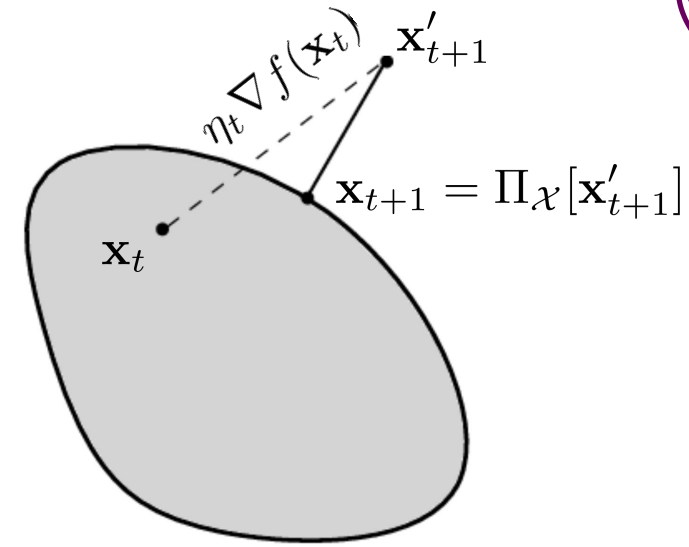
Then, we have

- (i) $f(\mathbf{x}_t) = U_t(\mathbf{x}_t)$;
- (ii) $f(\mathbf{x}) \leq U_t(\mathbf{x})$ holds for all $\mathbf{x} \in \mathbb{R}^d$;
- (iii) $\mathbf{x}_{t+1} = \arg \min_{\mathbf{x}} U_t(\mathbf{x})$ is equivalent to $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{L} \nabla f(\mathbf{x}_t)$.

Gradient Descent

- GD Template:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)]$$



- $\mathbf{x}_1$ can be an arbitrary point inside the domain.
- $\eta_t > 0$ is the potentially time-varying *step size* (or called *learning rate*).
- Projection $\Pi_{\mathcal{X}}[\mathbf{y}] = \arg \min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$ ensures the feasibility.

GD Template

GD template: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)]$

⇒ It only forms a general template, and there are still much complexity yet to specify.

- step size η_t
- output sequence $\bar{\mathbf{x}}_t$ based on $\{\mathbf{x}_s\}_{s=1}^t$
- stochastic case: gradient estimates
- GD is only a template, and many variants exist
- ...

Key requirements: provable guarantees, particularly with finite-sample (non-asymptotic) convergence rates

The First Gradient Descent Lemma

Lemma 2. Suppose that f is proper, closed and convex; the feasible domain $\mathcal{X}$ is nonempty, closed and convex. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by the gradient descent method, $\mathcal{X}^*$ be the optimal set of the optimization problem and f^* be the optimal value. Then for any $\mathbf{x}^* \in \mathcal{X}^*$ and $t \geq 0$,

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 \leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 - 2\eta_t(f(\mathbf{x}_t) - f^*) + \eta_t^2 \|\nabla f(\mathbf{x}_t)\|^2.$$

Proof:

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2 &= \|\Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)] - \mathbf{x}^*\|^2 \quad (\text{GD}) \\ &\leq \|\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) - \mathbf{x}^*\|^2 \quad (\text{Pythagoras Theorem}) \\ &= \|\mathbf{x}_t - \mathbf{x}^*\|^2 - 2\eta_t \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle + \eta_t^2 \|\nabla f(\mathbf{x}_t)\|^2 \\ &\leq \|\mathbf{x}_t - \mathbf{x}^*\|^2 - 2\eta_t(f(\mathbf{x}_t) - f^*) + \eta_t^2 \|\nabla f(\mathbf{x}_t)\|^2 \\ &\quad (\text{convexity: } f(\mathbf{x}_t) - f^* = f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^* \rangle) \quad \square \end{aligned}$$

Convex Function

Theorem 3. Assume the gradient of f is bounded by G , i.e., $\|\nabla f(\cdot)\| \leq G$, and assume the feasible domain $\mathcal{X}$ is bounded and convex with a diameter $D > 0$, that is, $\|\mathbf{x} - \mathbf{y}\|_2 \leq D$ holds for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by GD with step size $\eta_t = \frac{D}{G\sqrt{T}}$. Then

$$f(\bar{\mathbf{x}}_T) - f^* \leq \frac{DG}{\sqrt{T}} = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right),$$

where $\bar{\mathbf{x}}_T \triangleq \arg \min_{\{\mathbf{x}_t\}_{t=1}^T} f(\mathbf{x}_t)$ or $\bar{\mathbf{x}}_T \triangleq \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$.

Convex and Smooth Function

Theorem 4. Suppose the function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is convex and differentiable, and also L -smooth. GD updates by $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)$ with step size $\eta_t = \frac{1}{L}$, and then GD enjoys the following convergence guarantee:

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

Result Overview

Table 1: A summary of convergence rates of GD for different function families, where we use $\kappa \triangleq L/\sigma$ to denote the condition number.

Function Family		Step Size	Output Sequence	Convergence Rate	
G-Lipschitz	convex	$\eta = \frac{D}{G\sqrt{T}}$	$\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$	$\mathcal{O}(1/\sqrt{T})$	<i>lipschitz</i>
	σ -strongly convex	$\eta_t = \frac{2}{\sigma(t+1)}$	$\bar{\mathbf{x}}_T = \sum_{t=1}^T \frac{2t}{T(T+1)} \mathbf{x}_t$	$\mathcal{O}(1/T)$	
L -smooth	convex	$\eta = \frac{1}{L}$	$\bar{\mathbf{x}}_T = \mathbf{x}_T$	$\mathcal{O}(1/T)$	<i>smooth</i>
	σ -strongly convex	$\eta = \frac{2}{\sigma+L}$	$\bar{\mathbf{x}}_T = \mathbf{x}_T$	$\mathcal{O}\left(\exp\left(-\frac{T}{\kappa}\right)\right)$	

For simplicity, we mostly focus on *unconstrained* domain, i.e., $\mathcal{X} = \mathbb{R}^d$.

The smoothness is defined to be with respect to ℓ_2 -norm.

GD and AGD

Table 1: A summary of convergence rates of GD method for smooth optimization.

Algorithm	Function Family	Step Size	Output Sequence	Convergence Rate	Remark
GD	L -smooth and convex	$\eta = \frac{1}{L}$	$\bar{\mathbf{x}}_T \triangleq \mathbf{x}_T$	$\mathcal{O}(1/T)$	suboptimal
	L -smooth and σ -strongly convex	$\eta = \frac{2}{\sigma+L}$	$\bar{\mathbf{x}}_T \triangleq \mathbf{x}_T$	$\mathcal{O}\left(\exp\left(-\frac{T}{\kappa}\right)\right)$	suboptimal
AGD	L -smooth and convex	$\mathbf{x}_{t+1} = \mathbf{y}_t - \frac{1}{L}\nabla f(\mathbf{y}_t), \mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t)$	$\bar{\mathbf{x}}_T \triangleq \mathbf{x}_T$	$\mathcal{O}(1/T^2)$	optimal
	L -smooth and σ -strongly convex	$\mathbf{x}_{t+1} = \mathbf{y}_t - \frac{1}{L}\nabla f(\mathbf{y}_t), \mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}(\mathbf{x}_{t+1} - \mathbf{x}_t)$	$\bar{\mathbf{x}}_T \triangleq \mathbf{x}_T$	$\mathcal{O}\left(\exp\left(-\frac{T}{\sqrt{\kappa}}\right)\right)$	optimal



Part 2. Online Optimization

- Fixed Optimization
- Interactive Optimization
- Online Optimization: Problem and Measure

Optimization

- A typical optimization problem: $\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$

Iterative algorithm (like GD) generates a sequence of iterates $\mathbf{x}_1, \dots, \mathbf{x}_T$, aiming to ensure that $F(\mathbf{x}_T) - F(\mathbf{x}^*)$ is small with respect to T .

for simplicity, consider last-iterate convergence

For each round $t = 1, \dots, T$:

- Algorithm outputs a decision $\mathbf{x}_t \in \mathcal{X}$.
- Algorithm has information about $F(\mathbf{x}_t)$ and $\nabla F(\mathbf{x}_t)$.

⇒ Note that the objective function remains *fixed* throughout the entire optimization process as the iterative algorithm progresses.

Interactive Optimization

- However, in many ML applications, the optimization objective *does not stay fixed*.

For each round $t = 1, \dots, T$:

- Algorithm outputs a decision $\mathbf{x}_t \in \mathcal{X}$.
- A new loss function $f_t : \mathcal{X} \rightarrow \mathbb{R}$ is revealed.
- Algorithm suffers the loss $f_t(\mathbf{x}_t)$.

Optimization is no longer performed against a single, fixed target; instead, it involves interactions between an algorithm and *an evolving objective*.

Example 1: Large-Scale ERM

- Consider the model training. Our goal is to minimize ERM

$$F(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^M \ell(\mathbf{x}; z_i)$$

- Big data: facing millions of samples (M is very large).
- Computing full gradient is almost impossible (like due to limited memory of GPU facilities).



Stochastic Optimization as Interaction

Stochastic optimization using *mini-batch* (by SGD/Adam or others)

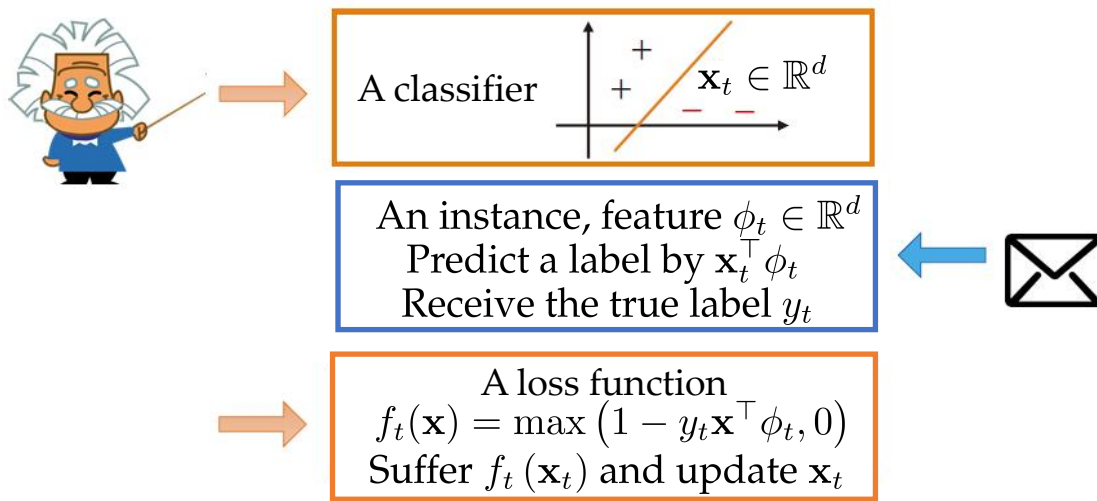
$$f_t(\mathbf{x}) = \frac{1}{|S_t|} \sum_{z_i \in S_t} \ell(\mathbf{x}; z_i)$$

Note that S_t is still sampled from a fixed distribution (over all M samples)

- The optimization objective f_t varies across iterations (due to changing mini-batches).
- This renders the optimization process *interactive* — the algorithm continuously adapts in response to a new batch of samples.

Example 2: Learning with Data Streams

- In applications, data may arrive in a form of stream, which requires the model to make *sequential predictions or even decisions*.
- For example: spam filtering



Sequential Predictions as Interaction

- We can formulate the sequential prediction problem as follows:

At each step t :

- The learner outputs a prediction $\mathbf{x}_t$.
- Then a new sample z_t is revealed.
- The instantaneous loss is $f_t(\mathbf{x}_t) = \ell(\mathbf{x}_t; z_t)$.

Note that the loss function f_t is sampled from (unknown) distribution $\mathcal{D}_\star$ or $\mathcal{D}_t$

- If data are stationary, we can view f_t as sampled from a fixed distribution $\mathcal{D}_\star$.
- Alternatively, if having non-stationarity, distribution $\mathcal{D}_t$ may vary with t .

Online Optimization: distribution-free

- Online optimization: an even general framework without distributional assumption.
- The function f_t may be arbitrary, or even chosen adversarially.
- The general interactive procedures of online optimization:

For each round $t = 1, \dots, T$:

- Learner gives a decision $\mathbf{x}_t \in \mathcal{X}$.
- Learner observes $f_t : \mathcal{X} \rightarrow \mathbb{R}$ and suffers the loss $f_t(\mathbf{x}_t)$.

A Game-theoretic Language

- Online (Interactive) optimization as a *repeated game* between
 - **Player**: essentially the learner, or you can think as the “learning model”
 - **Environments**: an abstraction of all factors evaluating the model.

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

- Player first picks a model $\mathbf{x}_t \in \mathcal{X}$.
- Simultaneously environments pick an online function $f_t : \mathcal{X} \rightarrow \mathbb{R}$.
- Player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.

Fixed vs Interactive Optimization

- Fixed Optimization (Closed-world Learning)
 - The training data are all available on hand
 - The objective is fixed with no changes.
- Online Optimization (Open-ended Learning)
 - This may be due to sampling issue, so connected to stochastic optimization
 - Or since data are in the form of stream, so it is crucial for continual learning
 - Even more complicated: decision this round may also influent the environment, decision-theoretic RL/control



Performance Measure

- The best in hindsight

After T rounds, suppose we look back and imagine that we *did* know all $\{f_t\}_{t=1}^T$.

Then the best single decision in hindsight would be:

$$\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$$

which we obviously could *not* know in advance.

Regret:
$$\text{REG}_T = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$$

*benchmark performance with the
offline model (optimal in hindsight)*

*“If I had known all the outcomes ahead of time, I would have
chosen differently — and I **regret** the extra loss I’ve accumulated.”*

Performance Measure

- In online learning, we define *regret* as measure for $\{\mathbf{x}_t\}_{t=1}^T$:

$$\text{REG}_T = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$$

benchmark performance with the offline model (optimal in hindsight)

We hope the regret be sub-linear dependence with T

$$\frac{\text{REG}_T}{T} \rightarrow 0 \text{ as } T \rightarrow \infty$$

Hannan Consistency

ALT'16

Hannan Consistency in On-Line Learning
in Case of Unbounded Losses Under Partial
Monitoring^{*,**}

Chamy Allenberg¹, Peter Auer², László Györfi³, and György Ottucsák³

worst-case-dynamic regret $\text{REG}_T(\{\mathbf{x}_t^*\}) = \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}_t^*)$ *suffer from overfitting issue*

general dynamic regret $\text{REG}_T(\{\mathbf{u}_t\}) = \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)$ *“online ensemble” framework*



Part 3. Online Convex Optimization

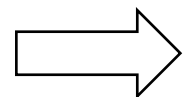
- Online Convex Optimization
- Problem Space
- Online Gradient Descent
- Lower Bound

Is Online Optimization (provably) solvable?

- In general, the online optimization is *too hard to solve*.

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

- Player first picks a model $\mathbf{x}_t \in \mathcal{X}$.
- Simultaneously environments pick an online function $f_t : \mathcal{X} \rightarrow \mathbb{R}$.
- Player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.



A Trackable Case: *Online Convex Optimization*

*requiring feasible domain and online functions to be **convex***

Online Convex Optimization

- Requirements:

- (1) feasible domain is a convex set
- (2) online functions are convex

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

- Player first picks a model $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$.
- Environments pick an online convex function $f_t : \mathcal{X} \rightarrow \mathbb{R}$.
- Player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.



Online Convex Optimization: Hardness

Clearly, similar to offline opt, curvature information matters:

- Convex vs Strongly convex?
- Lipschitz vs Smooth?

There are other issues due to the *interaction* nature:

- **Feedback:** How much information can the learner access from environments?
- **Environments:** How powerful is the environment?

Given that it can choose the loss function!

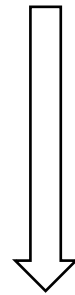
Hardness: Different Feedback

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

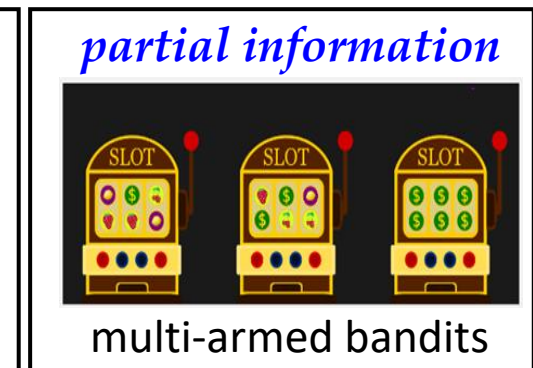
- Player first picks a model $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$.
- Environments pick an online convex function $f_t : \mathcal{X} \rightarrow \mathbb{R}$.
- Player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.

on the feedback information:

- **full information**: observe entire f_t (or at least gradient $\nabla f_t(\mathbf{x}_t)$)
- **partial information (bandits)**: observe function value $f_t(\mathbf{x}_t)$ only



less information



Hardness: Different Environments

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

- Player first picks a model $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$.
- **Environments** pick an online convex function $f_t : \mathcal{X} \rightarrow \mathbb{R}$.
- Player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.

on the difficulty of environments:

- stochastic setting

- adversarial setting { oblivious
adaptive
(non-oblivious)

*less restricted
but harder*

oblivious adversary



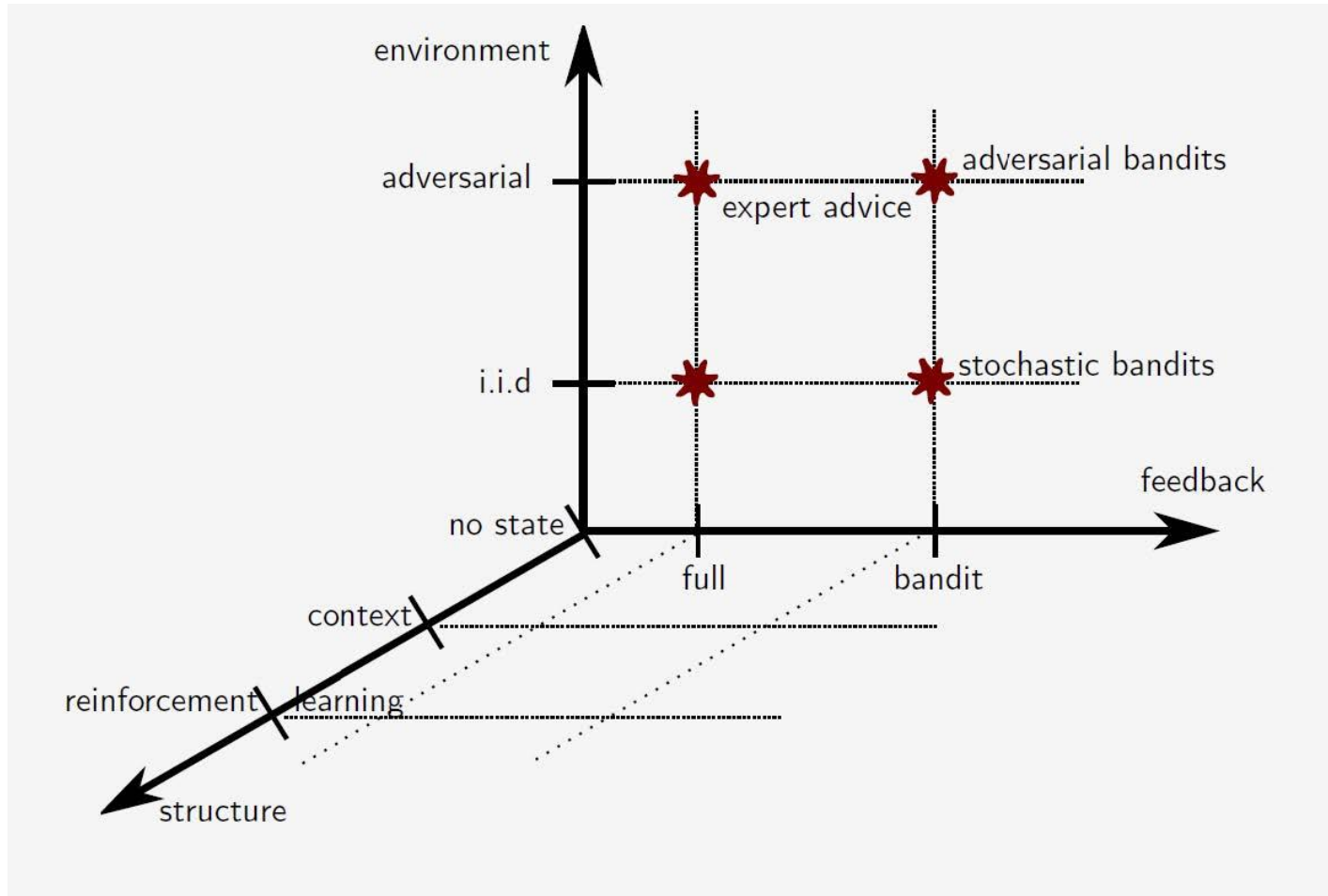
examination

adaptive adversary



interview

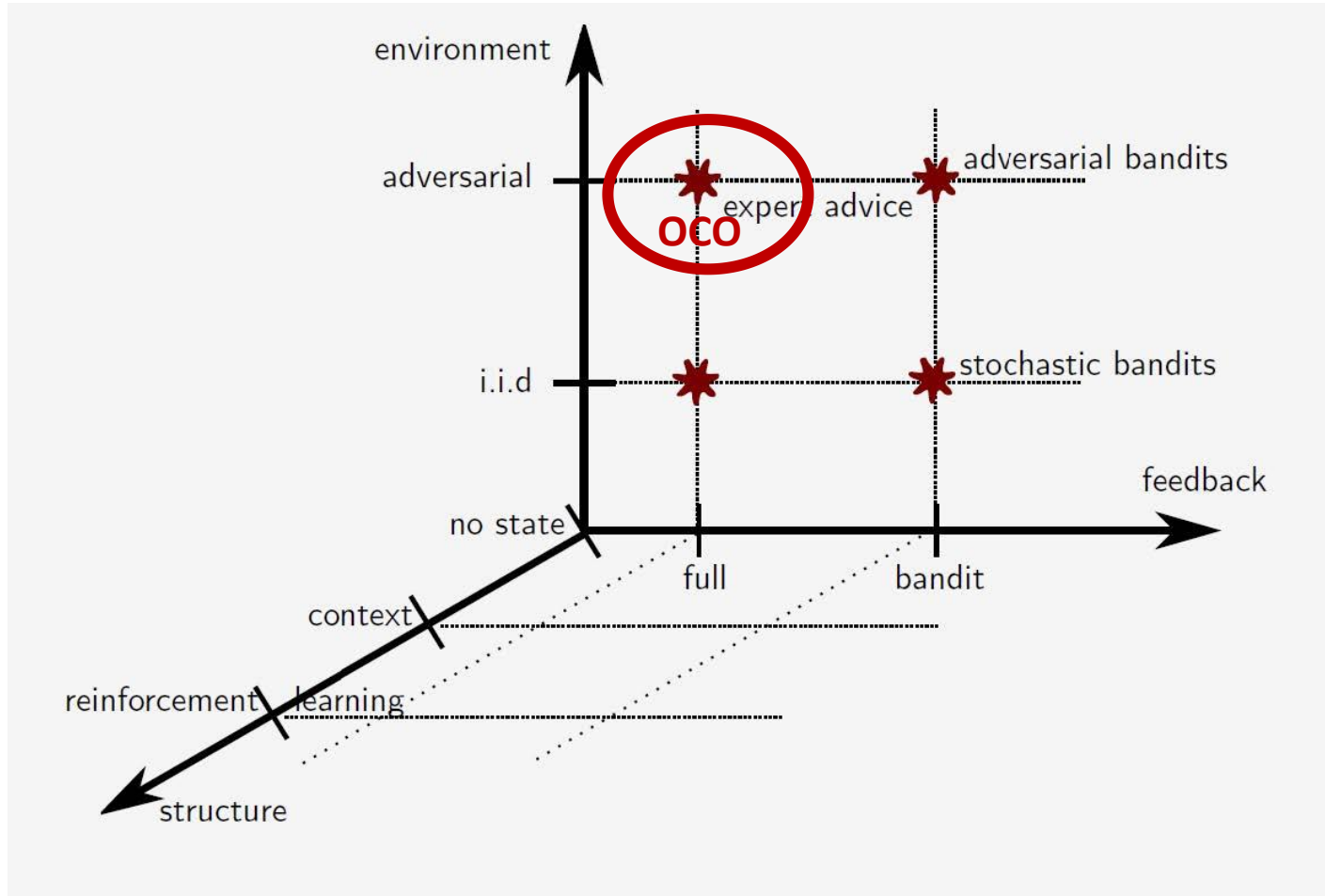
Online Learning: Problem Space



- Full-information setting:
 - Online Convex Optimization
 - Prediction with Expert Advice
 - ...
- Partial-information setting:
 - Multi-Armed Bandits
 - Linear Bandits
 - Parametric Bandits
 - Bandit Convex Optimization
 - ...

Yevgeny Seldin. The Space of Online Learning Problems, ECML-PKDD, Porto, Portugal, 2015.

Online Learning: Problem Space



- Full-information setting:
 - Online Convex Optimization
 - Prediction with Expert Advice
 - ...
- Partial-information setting:
 - Multi-Armed Bandits
 - Linear Bandits
 - Parametric Bandits
 - Bandit Convex Optimization
 - ...

Yevgeny Seldin. The Space of Online Learning Problems, ECML-PKDD, Porto, Portugal, 2015.

OCO: Convex Functions

Definition 5 (Convex Function). A function $f : \mathcal{X} \mapsto \mathbb{R}$ is convex if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, it holds $\forall \alpha \in [0, 1]$, $f((1 - \alpha)\mathbf{x} + \alpha\mathbf{y}) \leq (1 - \alpha)f(\mathbf{x}) + \alpha f(\mathbf{y})$.

Equivalently, if f is differentiable, we have that $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$,

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

The feasible set $\mathcal{X}$ is closed and convex in Euclidean space, and $f_1, \dots, f_T$ are convex functions.

Regret Minimization in OCO

- We focus on the G -Lipschitz functions

Assumption 1 (Bounded Gradient). The norm of the subgradients is upper bounded by G , i.e., $\|\nabla f_t(\mathbf{x})\| \leq G$ for all $\mathbf{x} \in \mathcal{X}$ and $t \in [T]$.

- The following is *domain boundedness*

Assumption 2 (Bounded Domain). The diameter of the feasible domain $\mathcal{X}$ is upper bounded by D , i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}, \|\mathbf{x} - \mathbf{y}\| \leq D$.

Essentially working on Lipschitz (online) optimization with a bounded feasible domain.

On the domain boundedness

Theorem 5.1. Let $\mathcal{V} \subset \mathbb{R}^d$ be any non-empty bounded closed convex subset. Let $D = \sup_{\mathbf{v}, \mathbf{w} \in \mathcal{V}} \|\mathbf{v} - \mathbf{w}\|_2$ be the diameter of $\mathcal{V}$. Let $\mathcal{A}$ be any (possibly randomized) algorithm for OLO on $\mathcal{V}$. Let T be any non-negative integer. Then, there exists a sequence of vectors $\mathbf{g}_1, \dots, \mathbf{g}_T$ with $\|\mathbf{g}_t\|_2 \leq L$ and $\mathbf{u} \in \mathcal{V}$ such that the regret of algorithm $\mathcal{A}$ satisfies

$$\text{Regret}_T(\mathbf{u}) = \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{x}_t \rangle - \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u} \rangle \geq \frac{\sqrt{2}LD\sqrt{T}}{4}.$$

Theorem 5.6. For $T \in \mathbb{N}$ suppose that there is an online convex optimization algorithm that guarantees regret at most ϵ_T against the null competitor on any sequence of T linear and L -Lipschitz losses $\ell_t : \mathbb{R} \rightarrow \mathbb{R}$ for $t = 1, \dots, T$. Let $U > 0$ be such that $1 \leq W\left(\frac{\sqrt{T}UL}{5\epsilon_T}\right) \leq \frac{\sqrt{T}}{2}$, then there exists a sequence of $\mathbf{g}_t$ with $\|\mathbf{g}_t\|_2 \leq L$ and a competitor $\mathbf{u} \in \mathbb{R}^d$ with $\|\mathbf{u}\|_2 = U$, such that

$$\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{u} \rangle \geq R_T(U) := UL\sqrt{T} \left(\sqrt{2W\left(\frac{\sqrt{T}UL}{5\epsilon_T}\right)} - 1 \right) - 2UL + \epsilon_T,$$

where $W : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is the Lambert function.



Recall Lipschitz (offline) optimization

For G -Lipschitz optimization with $\min_{\mathbf{x}} f(\mathbf{x})$, we use GD to optimize it and obtain $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ optimal rate.

Optimal Result with Known T

Theorem 5. Under the same assumptions with Theorem 1, assume the feasible domain $\mathcal{X}$ is bounded and convex with a diameter $D > 0$, that is, $\|\mathbf{x} - \mathbf{y}\|_2 \leq D$ holds for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by GD with step size

$$\eta_t = \frac{D}{G\sqrt{T}}.$$

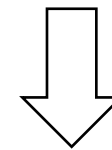
Then

$$f(\bar{\mathbf{x}}_T) - f^* \leq \frac{DG}{\sqrt{T}} = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right),$$

where $\bar{\mathbf{x}}_T \triangleq \arg \min_{\{\mathbf{x}_t\}_{t=1}^T} f(\mathbf{x}_t)$ or $\bar{\mathbf{x}}_T \triangleq \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$.

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)]$$

Can we extend the GD idea to
online optimization?



$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)]$$

OCO: OGD Algorithm

Online Gradient Descent (OGD)

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

1. the player first picks a model $\mathbf{x}_t \in \mathcal{X}$;
2. and simultaneously environments pick a **convex** online function $f_t : \mathcal{X} \rightarrow \mathbb{R}$;
3. the player suffers loss $f_t(\mathbf{x}_t)$, observes the information of f_t and update the model according to $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)]$.

This belongs to the full-information setting, so player can access the gradient $\nabla f_t(\mathbf{x}_t)$.

Actually, only gradient is required, so it's also called **gradient-feedback OCO model**.

Regret Analysis of OGD

Theorem 5 (Regret bound for OGD). *Under Assumption 1 (G-Lipschitz) and Assumption 2 (D-bounded domain), online gradient descent (OGD) with step sizes $\eta_t = \frac{D}{G\sqrt{t}}$ for $t \in [T]$ guarantees:*

$$\text{REG}_T = \sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x}) \leq \frac{3}{2}GD\sqrt{T} = \mathcal{O}(\sqrt{T}).$$

The First Gradient Descent Lemma

Lemma 3. Suppose that f_t is proper, closed and convex; the feasible domain $\mathcal{X}$ is nonempty, closed and convex. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by the gradient descent method. Then for any $\mathbf{u} \in \mathcal{X}$ and $t \geq 0$,

$$\|\mathbf{x}_{t+1} - \mathbf{u}\|^2 \leq \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t(f_t(\mathbf{x}_t) - f_t(\mathbf{u})) + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2.$$

Proof:

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{u}\|^2 &= \|\Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)] - \mathbf{u}\|^2 \quad (\text{GD}) \\ &\leq \|\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t) - \mathbf{u}\|^2 \quad (\text{Pythagoras Theorem}) \\ &= \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2 \\ &\leq \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t(f_t(\mathbf{x}_t) - f_t(\mathbf{u})) + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2 \\ &\quad (\text{convexity: } f_t(\mathbf{x}_t) - f_t(\mathbf{u}) = f_t(\mathbf{x}_t) - f_t(\mathbf{u}) \leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle) \quad \square \end{aligned}$$

Proof for OGD Regret Bound

Proof: We use the first gradient descent lemma to analyze online gradient descent.

Lemma 3. Suppose that f_t is proper, closed and convex; the feasible domain $\mathcal{X}$ is nonempty, closed and convex. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by the gradient descent method. Then for any $\mathbf{u} \in \mathcal{X}$ and $t \geq 0$,

$$\|\mathbf{x}_{t+1} - \mathbf{u}\|^2 \leq \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t(f_t(\mathbf{x}_t) - f_t(\mathbf{u})) + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2.$$

By Lemma 3 and the gradient boundedness, we have

$$2(f_t(\mathbf{x}_t) - f_t(\mathbf{u})) \leq \frac{\|\mathbf{x}_t - \mathbf{u}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{u}\|^2}{\eta_t} + \eta_t G^2$$

Proof for OGD Regret Bound

Proof: By setting $\eta_t = \frac{D}{G\sqrt{t}}$ (with $\frac{1}{\eta_0} := 0$), summing over T :

$$\begin{aligned} 2 \left(\sum_{t=1}^T f_t(\mathbf{x}_t) - f_t(\mathbf{u}) \right) &\leq \sum_{t=1}^T \frac{\|\mathbf{x}_t - \mathbf{u}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{u}\|^2}{\eta_t} + G^2 \sum_{t=1}^T \eta_t && \text{(GD lemma)} \\ &\leq \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + G^2 \sum_{t=1}^T \eta_t && (\|\mathbf{x}_{T+1} - \mathbf{u}\|^2 \geq 0) \\ &\leq D^2 \sum_{t=1}^T \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + G^2 \sum_{t=1}^T \eta_t \\ &\leq D^2 \frac{1}{\eta_T} + G^2 \sum_{t=1}^T \eta_t && (\eta_t = \frac{D}{G\sqrt{t}} \text{ and } \sum_{t=1}^T \frac{1}{\sqrt{t}} \leq 2\sqrt{T}) \\ &\leq 3DG\sqrt{T}. \end{aligned}$$

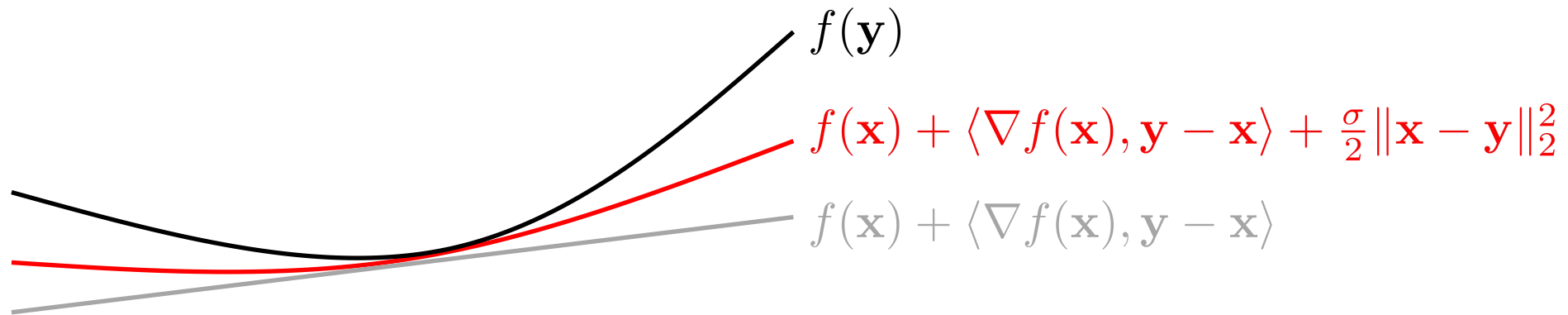
□

OCO: Strongly Convex Functions

Definition 6 (Strong Convexity). A function f is σ -strongly convex if, for any $\mathbf{x}, \mathbf{y} \in \text{dom } f$,

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

or equivalently, $\nabla^2 f(\mathbf{x}) \succeq \alpha I$.





Recall Lipschitz (offline) Optimization

For G -Lipschitz optimization with $\min_{\mathbf{x}} f(\mathbf{x})$, we use GD to optimize it and obtain $\mathcal{O}\left(\frac{1}{T}\right)$ optimal rate.

Strongly Convex and Lipschitz

Theorem 7. Under the same assumptions with Theorem 1, except that f is σ -strongly-convex. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by GD with step size

$$\eta_t = \frac{2}{\sigma(t+1)}.$$

Then (i)

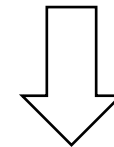
$$f(\bar{\mathbf{x}}_T) - f^* \leq \frac{2G^2}{\sigma(T+1)} = \mathcal{O}\left(\frac{1}{T}\right),$$

where $\bar{\mathbf{x}}_T \triangleq \arg \min_{\{\mathbf{x}_t\}_{t=1}^T} f(\mathbf{x}_t)$ or $\bar{\mathbf{x}}_T \triangleq \sum_{t=1}^T \frac{2t}{T(T+1)} \mathbf{x}_t$.

And (ii)

$$\|\bar{\mathbf{x}}_T - \mathbf{x}^*\| \leq \frac{2G}{\sigma\sqrt{T+1}}.$$

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)]$$



$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)]$$

Again, we extend it to online optimization.

OGD for Strongly Convex Functions

Online Gradient Descent (OGD)

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

- (1) the player first picks a model $\mathbf{x}_t \in \mathcal{X}$;
- (2) and simultaneously environments pick a *strongly convex function* $f_t : \mathcal{X} \rightarrow \mathbb{R}$;
- (3) the player suffers loss $f_t(\mathbf{x}_t)$, observes the information of f_t and update the model according to $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla f_t(\mathbf{x}_t)]$.

OGD for Strongly Convex Functions

Theorem 6 (Regret bound for strongly-convex functions). *Under Assumption 1 (G-Lipschitz), for σ -strongly convex loss functions, online gradient descent with step sizes $\eta_t = \frac{1}{\sigma t}$ achieves the following guarantee*

$$\text{REG}_T \leq \frac{G^2}{2\sigma} (1 + \log T) = \mathcal{O}(\log T).$$

- Strongly convex case compared with convex case: $\mathcal{O}(\log T)$ vs. $\mathcal{O}(\sqrt{T})$
- A caveat is that we now don't need Assumption 2 (bounded domain).

OGD for Strongly Convex Functions

Proof: we start by extending *the first GD lemma* to strongly convex case.

Strongly convex case:

$$\begin{aligned}\|\mathbf{x}_{t+1} - \mathbf{u}\|^2 &\leq \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2 \\ &\leq \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t \left(f_t(\mathbf{x}_t) - f_t(\mathbf{u}) + \frac{\sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|^2 \right) + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2 \\ &\quad \text{(strong convexity: } f_t(\mathbf{x}_t) - f_t(\mathbf{u}) + \frac{\sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|^2 \leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle \text{)} \\ &\leq (1 - \sigma\eta_t) \|\mathbf{x}_t - \mathbf{u}\|^2 - 2\eta_t (f_t(\mathbf{x}_t) - f_t(\mathbf{u})) + \eta_t^2 \|\nabla f_t(\mathbf{x}_t)\|^2 \\ \implies f_t(\mathbf{x}_t) - f_t(\mathbf{u}) &\leq \frac{\eta_t^{-1} - \sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|^2 - \frac{\eta_t^{-1}}{2} \|\mathbf{x}_{t+1} - \mathbf{u}\|^2 + \frac{\eta_t G^2}{2} \quad \text{(rearranging)}\end{aligned}$$

OGD for Strongly Convex Functions

Proof:
$$f_t(\mathbf{x}_t) - f_t(\mathbf{u}) \leq \frac{\eta_t^{-1} - \sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|^2 - \frac{\eta_t^{-1}}{2} \|\mathbf{x}_{t+1} - \mathbf{u}\|^2 + \frac{\eta_t G^2}{2}$$

Summing from $t = 1$ to T , setting $\eta_t = \frac{1}{\sigma t}$ (define $\frac{1}{\eta_0} := 0$):

$$\begin{aligned} 2 \sum_{t=1}^T (f_t(\mathbf{x}_t) - f_t(\mathbf{u})) &\leq \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma \right) + G^2 \sum_{t=1}^T \eta_t \quad \left(\frac{1}{\eta_0} := 0 \right) \\ &= 0 + G^2 \sum_{t=1}^T \frac{1}{\sigma t} \quad \left(\frac{1}{\eta_0} \triangleq 0, \|\mathbf{x}_{T+1} - \mathbf{u}\|^2 \geq 0, \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma = 0 \right) \\ &\leq \frac{G^2}{\sigma} (1 + \log T). \end{aligned}$$

□

Comparisons

Convex

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

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

$$\text{REG}_T \leq \frac{3}{2} G D \sqrt{T}$$

Strongly Convex

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

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

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

Lower Bounds

- A natural question: whether previous regret can be improved?
- *minimax bound*: smallest possible worst-case regret of any algorithm

$$\min_{\mathcal{A}} \max_{\ell_1, \dots, \ell_T} \text{REG}_T$$

Theorem 7 (Lower Bound for OCO). *Any algorithm for online convex optimization incurs $\Omega(DG\sqrt{T})$ regret in the worst case. This is true even if the cost functions are generated from a fixed stationary distribution.*

Proof Sketch. Construct a *hard* environment:

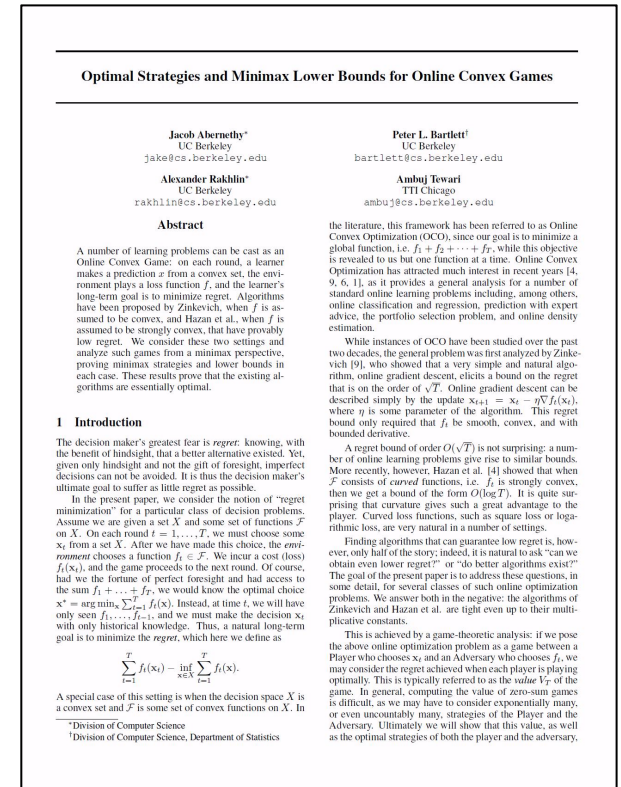
- Binary classification, loss functions in each iteration are chosen at random
- The hardness comes from the compression of a sequence of T -round random bits

Comparison



	Algo.	Step size	Upper Bound	Lower Bound
Convex	OGD	$\frac{D}{G\sqrt{t}}$	$\mathcal{O}(\sqrt{T})$	$\Omega(\sqrt{T})$
σ -Strongly Convex	OGD	$\frac{1}{\sigma t}$	$\mathcal{O}\left(\frac{\log T}{\sigma}\right)$	$\Omega\left(\frac{\log T}{\sigma}\right)$

Lower bound for strongly convex functions is more non-trivial.



Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In COLT, 2008.



Part 4. Online-to-Batch Conversion

- Convex Functions
- Strongly Convex Functions

What and Why?

- Online-to-Batch Conversion: solve *offline optimization* (namely, batch learning) using *online optimization/learning* technique.
- Why?
 - Offline optimization does not mean a fixed objective! In particular, *stochastic* (offline) optimization actually involves mini-batch sampling, which involves interactions and naturally requires online opt.
 - While online optimization adds a bit more complexity, it can render certain sophisticated tricks ineffective, thereby exposing the essential challenges and avoiding reliance on "hacked" solutions.

Offline Optimization

- Consider *offline optimization* $\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$ with stochastic opt method

Computational oracle: only access *noisy* gradient oracle, namely, $\mathbf{g}(\mathbf{x})$, such that

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] = \nabla F(\mathbf{x}), \text{ and } \mathbb{E}[\|\mathbf{g}(\mathbf{x})\|^2] \leq G^2$$

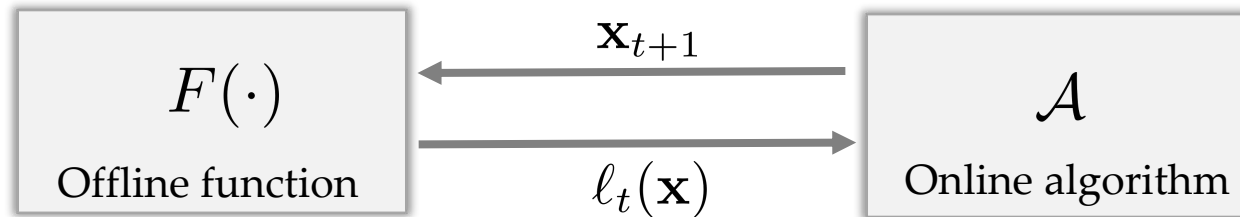
for some $G > 0$.

Example (large-scale opt.). Given dataset $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$, ERM optimizes

$$\min_{h \in \mathcal{H}} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i) \quad \Rightarrow \quad \begin{array}{l} \text{full gradient computation requires a pass of } \textit{all data} \\ \text{stochastic method only uses a } \textit{mini batch} \text{ at each round} \end{array}$$

Online-to-Batch Conversion

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



Algorithm 2 Online-to-Batch Conversion for Convex Functions

Input: noisy gradient oracle $\mathbf{g}(\cdot)$, online learning algorithm $\mathcal{A}$

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Obtain noisy gradient $\mathbf{g}(\mathbf{x}_t)$
 - 3: Pass loss function $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$ to online learning algorithm $\mathcal{A}$
 - 4: Receive next point $\mathbf{x}_{t+1}$ from $\mathcal{A}$
 - 5: **end for**
 - 6: **return** $\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$
-

Online-to-Batch Conversion

Theorem 8 (O2B Conversion for Convex Functions). Assume $F(\mathbf{x})$ is convex. Let $\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$. Define the online loss function $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$, then Algorithm 2 achieves the following guarantee:

$$\mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{x}^*) \leq \frac{\mathbb{E}[\text{REG}_T^{\mathcal{A}}(\mathbf{x}^*)]}{T}$$

where $\text{REG}_T^{\mathcal{A}}(\mathbf{u}) = \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}))$ be the regret of OL algorithm $\mathcal{A}$ over loss $\{\ell_t(\mathbf{x})\}_t^T$.

Proof: Using Jensen's inequality, we have

$$\begin{aligned} \mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{u}) &\stackrel{\text{(Jensen)}}{\leq} \frac{1}{T} \sum_{t=1}^T \mathbb{E}[F(\mathbf{x}_t)] - F(\mathbf{u}) \leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\langle \nabla F(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle] \quad (\text{Convexity}) \\ &= \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\langle \mathbf{g}(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle] = \frac{\mathbb{E}[\text{REG}_T^{\mathcal{A}}(\mathbf{u})]}{T}. \quad \square \end{aligned}$$

Stochastic Optimization

- Optimization Goal

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$$

Computational oracle: only access *noisy* gradient oracle, namely, $\mathbf{g}(\mathbf{x})$, such that

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] = \nabla F(\mathbf{x}), \text{ and } \mathbb{E}[\|\mathbf{g}(\mathbf{x})\|^2] \leq G^2.$$

By deploying OGD as $\mathcal{A}$ in Algorithm 2, we obtain a practical algorithm for convex functions, which updates as follows:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \mathbf{g}(\mathbf{x}_t)]$$

Algorithm 2 Online-to-Batch Conversion for Convex Functions

Input: noisy gradient oracle $\mathbf{g}(\cdot)$, online learning algorithm $\mathcal{A}$

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Obtain noisy gradient $\mathbf{g}(\mathbf{x}_t)$
 - 3: Pass loss function $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$ to online learning algorithm $\mathcal{A}$
 - 4: Receive next point $\mathbf{x}_{t+1}$ from $\mathcal{A}$
 - 5: **end for**
 - 6: **return** $\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$
-

Plugging the $\mathcal{O}(\sqrt{T})$ regret bound of OGD into Theorem 6, we can achieve $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ rate.

Understand SGD from Online Learning

Algorithm 3 Stochastic Gradient Descent

Input: noisy gradient oracle $\mathbf{g}(\cdot)$, step sizes $\{\eta_t\}$

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Obtain noisy gradient $\mathbf{g}(\mathbf{x}_t)$
 - 3: Update the model $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \mathbf{g}(\mathbf{x}_t)]$
 - 4: **end for**
 - 5: **return** $\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$
-

$$\begin{aligned} \text{OGD: } \mathbf{x}_{t+1} &= \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \nabla \ell_t(\mathbf{x}_t)] \\ &= \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \mathbf{g}(\mathbf{x}_t)] \end{aligned}$$

SGD is **equivalent to** deploying OGD in Algorithm 2 over functions $\{\ell_t(\mathbf{x})\}_{t=1}^T$.

Theorem 9 (Convergence of SGD). Assume $F(\mathbf{x})$ is convex, and let $\mathbf{x}^* \in \arg \min_{\mathbf{x}} F(\mathbf{x})$. Algorithm 3 achieves the following guarantee:

$$\mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{x}^*) \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right).$$

Stochastic Optimization

- Optimization Goal

$$\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$$

Computational oracle: only access *noisy* gradient oracle, namely, $\mathbf{g}(\mathbf{x})$, such that

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] = \nabla F(\mathbf{x}), \text{ and } \mathbb{E}[\|\mathbf{g}(\mathbf{x})\|^2] \leq G^2.$$

By deploying OGD as $\mathcal{A}$ in Algorithm 2, we obtain a practical algorithm for convex functions, which updates as follows:

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}} [\mathbf{x}_t - \eta_t \mathbf{g}(\mathbf{x}_t)]$$

Algorithm 2 Online-to-Batch Conversion for Convex Functions

Input: noisy gradient oracle $\mathbf{g}(\cdot)$, online learning algorithm $\mathcal{A}$

```
1: for  $t = 1, \dots, T$  do
2:   Obtain noisy gradient  $\mathbf{g}(\mathbf{x}_t)$ 
3:   Pass loss function  $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$  to online learning algorithm  $\mathcal{A}$ 
4:   Receive next point  $\mathbf{x}_{t+1}$  from  $\mathcal{A}$ 
5: end for
6: return  $\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$ 
```

Plugging the $\mathcal{O}(\sqrt{T})$ regret bound of OGD into Theorem 6, we can achieve $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ rate.

⇒ Note that function $\ell_t(\mathbf{x}) \triangleq \mathbf{g}(\mathbf{x}_t)^\top \mathbf{x}$ actually *depends* on the decision $\mathbf{x}_t$, which reveals that OGD regret can hold even against *adaptive adversary*.

Stochastic Optimization as Interaction

- As mentioned, stochastic opt can be deemed as an *interaction* between learning algo with the entire dataset (due to sampling)

Algorithm 2 Online-to-Batch Conversion for Convex Functions

Input: noisy gradient oracle $\mathbf{g}(\cdot)$, online learning algorithm $\mathcal{A}$

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Obtain noisy gradient $\mathbf{g}(\mathbf{x}_t)$
 - 3: Pass loss function $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$ to online learning algorithm $\mathcal{A}$
 - 4: Receive next point $\mathbf{x}_{t+1}$ from $\mathcal{A}$
 - 5: **end for**
 - 6: **return** $\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$
-

So, this method: essentially an online optimization method working on the *interactive loss* $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$ based on this round sampling.



O2B Conversion for Strongly Convex

- For σ -strongly convex functions, we can revise the O2B conversion and select OGD as $\mathcal{A}$ to achieve an $\mathcal{O}\left(\frac{\log T}{T}\right)$ convergence rate.
- But the key challenge is how to retain strong convexity in the online optimization, note that the vanilla conversion sets $\ell_t = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle$, which is a **linear** function.
- Surrogate loss technique: for σ -strongly convex function $F(\cdot)$

$$\begin{aligned} F(\mathbf{x}_t) - F(\mathbf{u}) &\leq \langle \nabla F(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle - \frac{\sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|_2^2 \\ &\stackrel{\mathbb{E}}{=} \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}) \end{aligned}$$

$$\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle + \frac{\sigma}{2} \|\mathbf{x}_t - \mathbf{x}\|_2^2$$

O2B Conversion for Strongly Convex

- For σ -strongly convex functions, we can revise the O2B conversion and select OGD as $\mathcal{A}$ to achieve an $\mathcal{O}\left(\frac{\log T}{T}\right)$ convergence rate.

Algorithm 4 Online-to-Batch Conversion for Convex Functions

Require: noisy gradient oracle $\mathbf{g}(\cdot)$, online learning algorithm $\mathcal{A}$

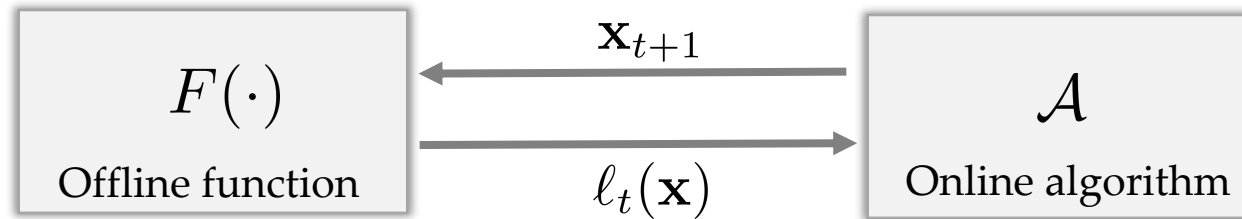
- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Obtain noisy gradient $\mathbf{g}(\mathbf{x}_t)$
 - 3: Pass loss function $\ell_t(\mathbf{x}) = \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle + \frac{\sigma}{2} \|\mathbf{x}_t - \mathbf{x}\|^2$ to online learning algorithm $\mathcal{A}$
 - 4: Receive next point $\mathbf{x}_{t+1}$ from $\mathcal{A}$
 - 5: **end for**
 - 6: **return** $\bar{\mathbf{x}}_T = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t$
-

retain strong convexity

However, this rate exhibits a gap to the $\Omega(1/T)$ lower bound for stochastic optimization over strongly convex functions.

Weighted O2B Conversion

- To achieve the optimal rate for strongly convex optimization, we introduce the *weighted* Online-to-Batch Conversion.



Algorithm 5 Weighted O2B Conversion for Strongly Convex Functions

Input: noisy gradient oracle $\mathbf{g}(\cdot)$, online learning algorithm $\mathcal{A}$, and **weights** $\{\alpha_t\}_{t=1}^T$

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Obtain noisy gradient $\mathbf{g}(\mathbf{x}_t)$
 - 3: Pass loss function $\ell_t(\mathbf{x}) = \langle \alpha_t \mathbf{g}(\mathbf{x}_t), \mathbf{x} \rangle + \frac{\alpha_t \sigma}{2} \|\mathbf{x}_t - \mathbf{x}\|^2$ to online learning algorithm $\mathcal{A}$
 - 4: Receive next point $\mathbf{x}_{t+1}$ from $\mathcal{A}$
 - 5: **end for**
 - 6: **return** $\bar{\mathbf{x}}_T = \frac{1}{\sum_{t=1}^T \alpha_t} \sum_{t=1}^T \alpha_t \mathbf{x}_t$
-

Weighted O2B Conversion

Theorem 10 (Weighted O2B Conversion for Strongly Convex Functions). Assume $F(\mathbf{x})$ is σ -strongly convex. Let $\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x})$. Define the surrogate loss function $\ell_t(\mathbf{x}) = \langle \alpha_t \mathbf{g}_t, \mathbf{x} \rangle + \frac{\alpha_t \sigma}{2} \|\mathbf{x}_t - \mathbf{x}\|^2$, and Algorithm 4 achieves the following guarantee:

$$\mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{x}^*) \leq \frac{\mathbb{E} [\text{REG}_T^{\mathcal{A}}(\mathbf{u})]}{\sum_{t=1}^T \alpha_t},$$

where $\text{REG}_T^{\mathcal{A}}(\mathbf{u}) = \sum_{t=1}^T (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{u}))$ be the regret of OL algorithm $\mathcal{A}$ over loss $\{\ell_t(\mathbf{x})\}_t^T$.

Proof:

$$\begin{aligned} \mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{u}) &\leq \frac{1}{\sum_{t=1}^T \alpha_t} \sum_{t=1}^T \alpha_t (\mathbb{E}[F(\mathbf{x}_t)] - F(\mathbf{u})) \\ &\leq \frac{1}{\sum_{t=1}^T \alpha_t} \sum_{t=1}^T \mathbb{E} \left[\alpha_t \langle \mathbf{g}(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u} \rangle - \frac{\alpha_t \sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|^2 \right] = \frac{\mathbb{E} [\text{REG}_T^{\mathcal{A}}(\mathbf{u})]}{\sum_{t=1}^T \alpha_t}. \quad \square \end{aligned}$$

O2B for Strongly Convex Functions

Theorem 11 (O2B for Strongly Convex Functions). Assume $F(\mathbf{x})$ is σ -strongly-convex, and let $\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x})$. By setting $\mathcal{A}$ as OGD with learning rate $\eta_t = \frac{1}{\sigma \sum_{i=1}^t \alpha_i}$ and $\alpha_t = t$, Algorithm 4 achieves the following guarantee:

$$\mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{x}^*) \leq \mathcal{O}\left(\frac{1}{T}\right).$$

Recall that in Theorem 8, we have

$$\mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{x}^*) \leq \frac{\mathbb{E}[\text{REG}_T^{\mathcal{A}}(\mathbf{u})]}{\sum_{t=1}^T \alpha_t}.$$

First, as $\alpha_t = t$, the denominator becomes $\frac{T(T+1)}{2}$.

Next, we focus on $\mathbb{E}[\text{REG}_T^{\mathcal{A}}(\mathbf{u})]$, where $\ell_t(\mathbf{x})$ is defined by $\ell_t(\mathbf{x}) = \langle \alpha_t \mathbf{g}_t, \mathbf{x} \rangle + \frac{\alpha_t \sigma}{2} \|\mathbf{x}_t - \mathbf{x}\|^2$.

O2B for Strongly Convex Functions



$$G_t = \alpha_t G$$

Proof:

$$\mathbb{E} [\text{REG}_T^A] = \mathbb{E} \left[\sum_{t=1}^T (f_t(\mathbf{x}_t) - f_t(\mathbf{u})) \right] \leq \mathbb{E} \left[\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}\|^2 \left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} - \frac{\sigma}{2} \right) \right] + \mathbb{E} \left[\sum_{t=1}^T \frac{G_t^2 \eta_t}{2} \right]$$

Choosing $\eta_t = \frac{1}{\sigma \sum_{s=1}^t \alpha_s}$, we have

$$\begin{aligned} \mathbb{E} [\text{REG}_T^A(\mathbf{x}^*)] &\leq \sum_{t=1}^T \frac{G_t^2}{2\sigma \sum_{i=1}^t \alpha_i} \\ &= \sum_{t=1}^T \frac{G^2 t^2}{2\sigma \sum_{i=1}^t t} = \frac{G^2}{2\sigma} \sum_{t=1}^T \frac{t}{t-1} = \mathcal{O}(T). \end{aligned}$$

Plugging it back to Theorem 8, we obtain

$$\mathbb{E}[F(\bar{\mathbf{x}}_T)] - F(\mathbf{x}^*) \leq \mathcal{O} \left(\frac{1}{T} \right).$$

□

OCO with Strongly Convex Functions

Proof: $f_t(\mathbf{x}_t) - f_t(\mathbf{u}) \leq \frac{\eta_t^{-1} - \sigma}{2} \|\mathbf{x}_t - \mathbf{u}\|^2 - \frac{\eta_t^{-1}}{2} \|\mathbf{x}_{t+1} - \mathbf{u}\|^2 + \frac{\eta_t G^2}{2}$

Summing from $t = 1$ to T , setting $\eta_t = \frac{1}{\sigma t}$ (define $\frac{1}{\eta_0} := 0$):

$$\begin{aligned} 2 \sum_{t=1}^T (f_t(\mathbf{x}_t) - f_t(\mathbf{u})) &\leq \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}\|^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma \right) + G^2 \sum_{t=1}^T \eta_t \quad \left(\frac{1}{\eta_0} := 0 \right) \\ &= 0 + G^2 \sum_{t=1}^T \frac{1}{\sigma t} \quad \left(\frac{1}{\eta_0} \triangleq 0, \|\mathbf{x}_{T+1} - \mathbf{u}\|^2 \geq 0, \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma = 0 \right) \\ &\leq \frac{G^2}{\sigma} (1 + \log T). \quad \square \end{aligned}$$



Summary of G-Lipschitz functions

- Offline Optimization: $\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$ with gradient oracle $\nabla F(\cdot)$
convergence rate Convex: $\mathcal{O}(\frac{1}{\sqrt{T}})$; Strongly Convex $\mathcal{O}(\frac{1}{T})$.
- Online Optimization: $\{f_t(\mathbf{x})\}_{t=1}^T$
regret bound Convex: $\mathcal{O}(\sqrt{T})$; Strongly Convex $\mathcal{O}(\log T)$.
- Stochastic Optimization: $\min_{\mathbf{x} \in \mathcal{X}} F(\mathbf{x})$ with noisy gradient oracle $\mathbf{g}(\cdot)$
convergence rate Convex: $\mathcal{O}(\frac{1}{\sqrt{T}})$; Strongly Convex $\mathcal{O}(\frac{1}{T})$.
by online-to-batch conversion

More bits of OGD

- Note that function $\ell_t(\mathbf{x}) \triangleq \mathbf{g}(\mathbf{x}_t)^\top \mathbf{x}$ *depends* on the decision $\mathbf{x}_t$, which actually reveals that OGD regret can hold even against *adaptive adversary*.

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

- (1) the player first picks a model $\mathbf{x}_t \in \mathcal{X}$;
- (2) and *simultaneously* environments pick an online function $f_t : \mathcal{X} \rightarrow \mathbb{R}$;
- (3) the player suffers loss $f_t(\mathbf{x}_t)$, observes some information about f_t and updates the model.

oblivious adversary



examination

adaptive adversary



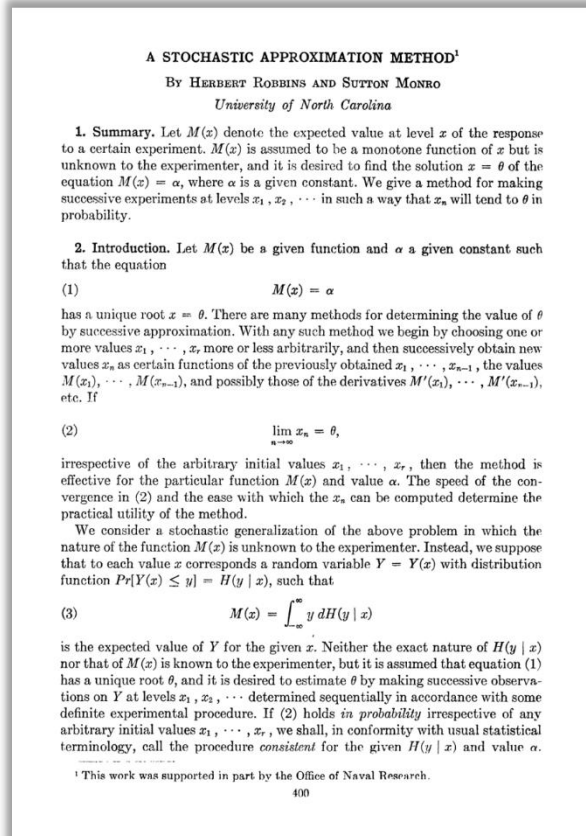
interview

The “simultaneously” requirement can be sometimes not necessary!

OGD for full-info OCO can handle the case when online functions *depend on $\mathbf{x}_t$* !

History: SGD

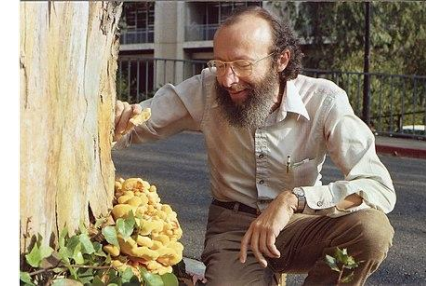
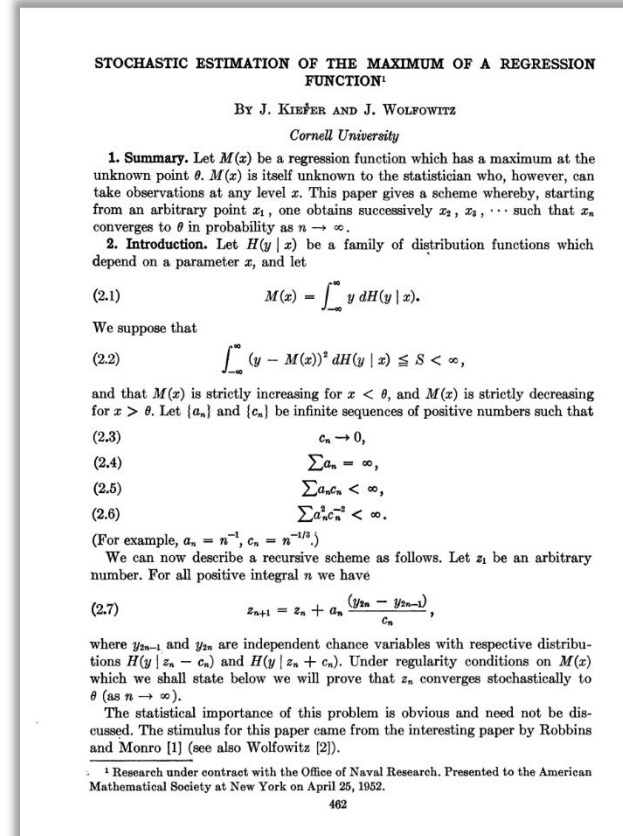
Robbins-Monro Method



Herbert Ellis Robbins
(1915 - 2001)

A Stochastic Approximation Method.
Herbert Robbins, Sutton Monro
Ann. Math. Statist. 22(3): 400-407 (September, 1951).

Kiefer-Wolfowitz Method



Jack Kiefer
(1924 - 1981)



Jacob Wolfowitz
(1910 - 1981)

Stochastic Estimation of the Maximum of a Regression Function
Jack Kiefer, Jacob Wolfowitz
Ann. Math. Statist. 23(3): 462-466 (September, 1952)

History: SGD



Herbert Ellis Robbins
(1915 - 2001)

Statistical Science
1986, Vol. 1, No. 2, 276-284

The Contributions of Herbert Robbins to Mathematical Statistics

Tze Leung Lai and David Siegmund

Herbert Robbins was born on January 12, 1915, in New Castle, Pennsylvania. In 1931 he entered Harvard College at the age of 16. Although his interests until then had been predominantly literary, he found himself increasingly attracted to mathematics under the influence of Marston Morse, who during many long conversations conveyed a vivid sense of the intellectual challenge of creative work in that field (cf. Page, 1984, p. 7). He received the A.B. summa cum laude in 1935, and the Ph.D. in 1938, also from Harvard. His thesis, in the field of combinatorial topology and written under the supervision of Hassler Whitney, was published in 1941 [3]. (Numbers in brackets refer to Robbins' bibliography at the end of this article.)

After graduation, Robbins worked for a year at the Institute for Advanced Study at Princeton as Marston Morse's assistant. He then spent the next three years at New York University as instructor in mathematics. He became nationally known in 1941 as the coauthor,

North Carolina at Chapel Hill. Having read [7] and [10], and greatly impressed by Robbins' mathematical skills, Hotelling offered him the position of associate professor to teach measure theory and probability to the graduate students in the new department. Robbins accepted the position and spent the next six years at Chapel Hill. During this relatively short period Robbins not only studied and developed an increasingly deep interest in statistics, but he also made a number of profound contributions to his new field: complete convergence [12], compound decision theory [25], stochastic approximation [26], and the sequential design of experiments [28], to name a few.

After a Guggenheim Fellowship at the Institute for Advanced Study during 1952-1953, Robbins moved from Chapel Hill to Columbia University as professor and chairman of the Department of Mathematical Statistics. Since 1953, with the exception of the three years 1965-1968 spent at Minnesota, Purdue, Berkeley, and Michigan, he has been at Columbia, where he

History: Two-Player Zero-Sum Games

Theory of repeated games



James Hannan
(1922–2010)



David Blackwell
(1919–2010)

Learning to play a game (1956)

Play a game repeatedly against a possibly suboptimal opponent

Zero-sum 2-person games played more than once

	1	2	...	M
1	$\ell(1,1)$	$\ell(1,2)$	...	
2	$\ell(2,1)$	$\ell(2,2)$	...	
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
N				

$N \times M$ known loss matrix

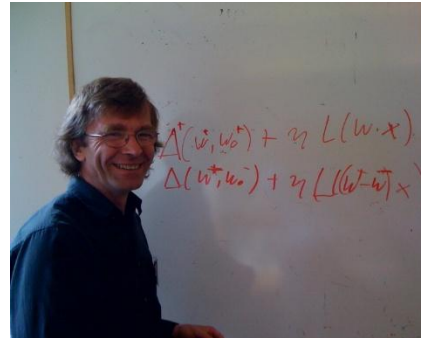
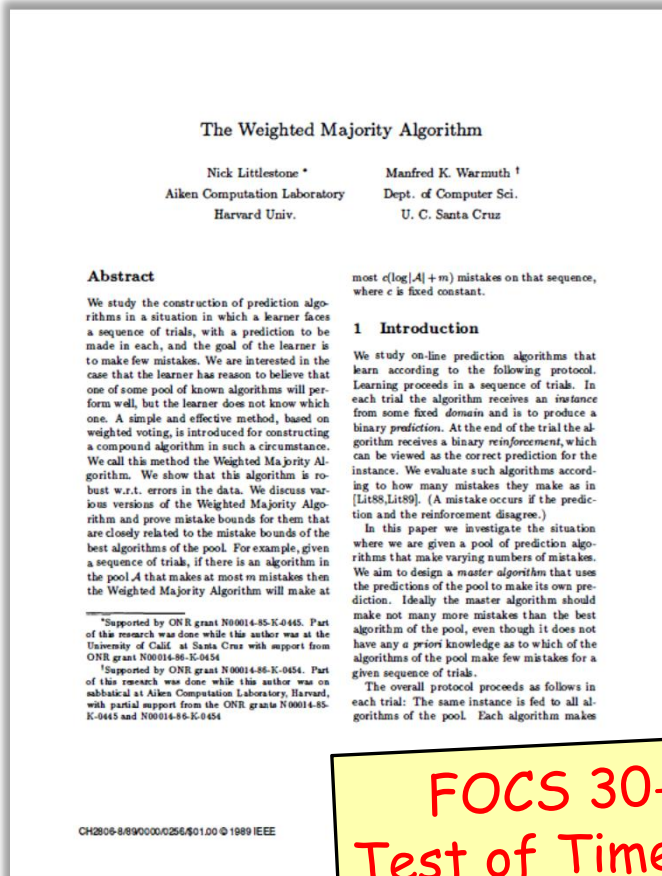
- Row player (**player**) has N actions
- Column player (**opponent**) has M actions

For each game round $t = 1, 2, \dots$

- Player chooses action i_t and opponent chooses action y_t
- The player suffers loss $\ell(i_t, y_t)$ (= gain of opponent)

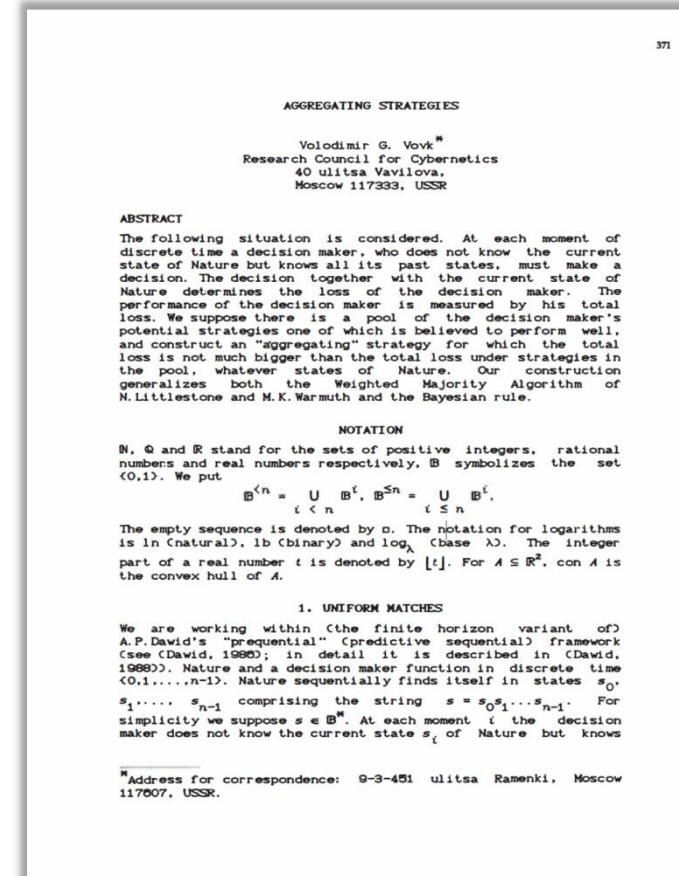
Player can learn from opponent's history of past choices $y_1, \dots, y_{t-1}$

History: Prediction with Expert Advice



Manfred Warmuth
UC Santa Cruz

**FOCS 30-year
Test of Time Award!**



Volodimir G. Vovk
Royal Holloway,
University of London

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

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

Summary



ONLINE OPTIMIZATION

Interaction optimization

Game-theoretic language

ONLINE-TO-BATCH CONVERSION

Online-to-batch conversion

Weighted O2B conversion

SGD, stochastic optimization

ONLINE CONVEX OPTIMIZATION

Problem formulation

Regret measure

Convex functions: OGD

Strongly convex functions: OGD

Q & A

Thanks!