Smoothed Online Learning is as Easy as Statistical Learning

Adam Block

Yuval Dagan

Noah Golowich

(Me)

Alexander Rakhlin

Independent & Concurrent Work

Oracle-Efficient Online Learning for Beyond Worst-Case Adversaries

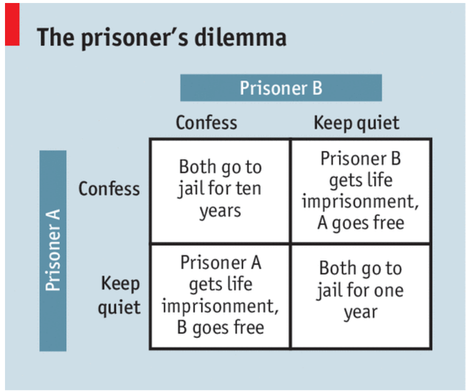
by Nika Haghtalab, Yanjun Han, Abhishek Shetty, Kunhe Yang

[HHSY, ‘22]

Online learning: motivation

Online learning is a fundamental model throughout learning theory; applications in many areas, such as:

- Sequential decision making (reinforcement learning)
- Equilibria computation in games
- Private learning
- Online versions of problems in related areas (auction design, learning of quantum states, etc.)



The prisoner's dilemma

		Prisoner B: Confess	Prisoner B: Keep quiet
Prisoner A	Confess	Both go to jail for ten years	Prisoner B gets life imprisonment, A goes free
Prisoner A	Keep quiet	Prisoner A gets life imprisonment, B goes free	Both go to jail for one year

Economist.com

Adversarial setting of online learning

- Fix set X and a hypothesis class H of hypotheses $h: X \to [0,1]$
- Given a loss function $\ell: [0,1] \times [0,1] \to [0,1]$; i.e., $\ell(\hat{y}, y) \in [0,1]$
- Over T rounds:

This talk: focus on **proper** learning algorithms, i.e., require $h_t \in H$ for all t

Learner	Adversary
Chooses a (possibly random) hypothesis $h_1: X \to [0,1]$	Chooses $(x_1, y_1) \in X \times [0,1]$
Learner suffers loss $\ell(h_1(x_1), y_1)$	
Chooses a (possibly random) hypothesis $h_2: X \to [0,1]$	Chooses $(x_2, y_2) \in X \times [0,1]$
Learner suffers loss $\ell(h_2(x_2), y_2)$	
$\vdots$	$\vdots$

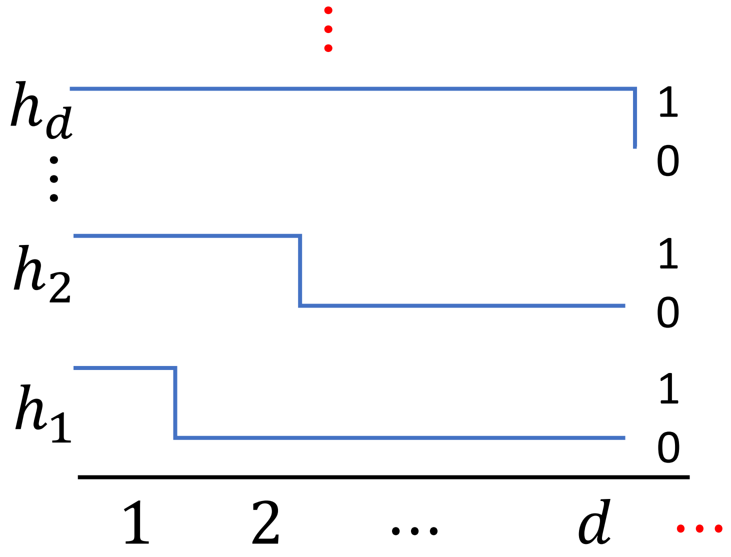
Goal: minimize **expected regret**: $\mathbb{E}[\mathrm{Reg}_T] = \mathbb{E}\left[\sum_{t=1}^{T} \ell(h_t(x_t), y_t) - \inf_{h \in H} \sum_{t=1}^{T} \ell(h(x_t), y_t)\right]$

Lower bounds for adversarial online learning

- Suppose $X = \mathbb{N}$ and consider class of thresholds:

$$H_{\text{thres}} = \{x \mapsto \mathbb{I}[x \leq w] : w \in \mathbb{N}\}$$

- (Unfortunate) fact: for any learner, adversary can choose examples (x_t, y_t) so that $\mathbb{E}[\text{Reg}_T] \geq T/2$



- **Standard "fix":** in case of thresholds, truncate to $X = \{1, 2, \ldots, d\}$
  - In online adversarial setting: can show $\mathbb{E}[\text{Reg}_T] \leq O(\sqrt{T \cdot \log d})$

- Contrast with "*offline (i.e., statistical) setting*": $(x_t, y_t) \sim \mu$ i.i.d. for some distribution μ; for thresholds:
  - Then can get error rates scaling as $O(\sqrt{T})$ – no dependence on d!

Minimax rates for binary classes

- Generalizing from thresholds: consider a class H of **binary** hypotheses, i.e., $h : X \to \{0,1\}$

Theorem [BPS,'09], [ABDMNY,'21]: The optimal *online* learning regret bound for any learner against an adversary is

$$\mathbb{E}[\mathrm{Reg}_T] = \Theta(\sqrt{\mathrm{Ldim}(H) \cdot T})$$

- $\mathrm{Ldim}(H)$ is **Littlestone dimension** of the class H (won't define here)
- Contrast with the *offline* (statistical) setting, where statistical rates scale with **VC dimension** of H
- In general, $\mathrm{Ldim}(H) \geq \mathrm{VCdim}(H)$:
  - E.g., for thresholds on $\{1, 2, \dots, d\}$: $\mathrm{Ldim} = \log d$, $\mathrm{VCdim} = 1$

Minimax rates for general classes

- Consider a class H of **real-valued** hypotheses, i.e., $h : X \to [0,1]$

Theorem [BDR, '21]: Under mild assumptions, the optimal online learning regret in the real-valued case is $\mathbb{E}[\mathrm{Reg}_T] = \Theta(\sqrt{T} \cdot \int_0^1 \sqrt{\mathrm{sfat}_\alpha(H)}\, d\alpha\,)$

- $\mathrm{sfat}_\alpha(H)$ is **sequential fat-shattering dimension at scale α** of the class H (won't define here)
- Contrast with the *offline* (statistical) setting, where statistical rates scale with **fat-shattering dimension at scale α (denoted fat_α)** of H
- In general, $\mathrm{sfat}_\alpha(H) \geq \mathrm{fat}_\alpha(H)$:
  - E.g., for thresholds on $\{1, 2, \ldots, d\}$: $\mathrm{sfat}_\alpha(H) = \log d$, $\mathrm{fat}_\alpha(H) = 1$ for all $\alpha \in (0,1)$

Beyond worst-case adversaries

Question [RST'11], [HRS'20], [HRS'21]: Can we avoid any dependence on Littlestone dimension (in binary case) by placing some assumption on the adversary?

- The "most mild" type of adversary is i.i.d. adversary: $(x_t, y_t) \sim \mu$ for some fixed & known μ
- Under such i.i.d. adversary: for binary classes, optimal regret is $O(\sqrt{\mathrm{VCdim}(H) \cdot T})$
- So: under appropriate assumptions, want regret scaling with VC dimension!

More generally: for real-valued classes, want to avoid dependence on sequential fat-shattering dimension, and just get scaling with fat-shattering dimension.

Smoothed adversarial setting of online learning

- Fix set X, hypothesis class H of hypotheses $h: X \to [0,1]$, loss $\ell(\hat{y}, y) \in [0,1]$
- Fix a (known) distribution μ on X: *only assume that we can sample from μ*

Definition [HRS'20], [HRS'21]: Given $\mu \in \Delta(X)$ and $\sigma \in (0,1]$, define

$$\mathrm{SMOOTH}_\sigma(\mu) \coloneqq \left\{P \in \Delta(X): \frac{P(E)}{\mu(E)} \leq \frac{1}{\sigma} \text{ for all } E \subset X\right\}$$

Learner	Adversary
Chooses a (possibly random) hypothesis $h_1: X \to \{0,1\}$	Chooses $p_1 \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_1 \sim p_1$, y_1 adversarially
Learner suffers loss $\ell(h_1(x_1), y_1)$	
Chooses a (possibly random) hypothesis $h_2: X \to \{0,1\}$	Chooses $p_2 \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_2 \sim p_2$, y_2 adversarially
Learner suffers loss $\ell(h_2(x_2), y_2)$	
⋮	⋮

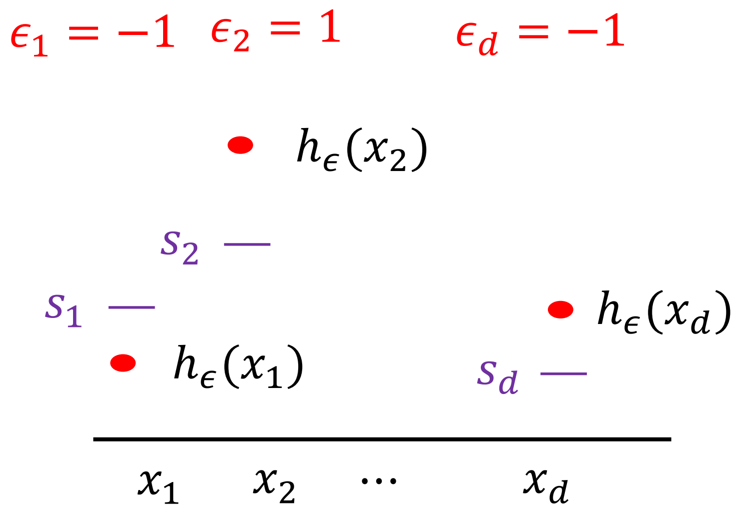
Overview of our contributions

1. Tight regret upper bound of learning a real-valued class in smoothed online setting
   - Extends result of [HRS, '21] treating binary-valued setting
2. **Oracle-efficient** upper bound for learning a real-valued class in smoothed online setting
   - Dependence on smoothness parameter σ is exponentially worse than above upper bound.
3. Lower bound showing that regret of oracle-efficient algorithm cannot be significantly improved
   - Establishes computational-statistical gap for smoothed online learning

Review: VC dimension, fat-shattering dimension

- Recall: given set X and class H of hypotheses $h : X \to [0,1]$
- Say H is **shattered** by points $x_1, \dots, x_d \in X$ at scale α if there are $s_1, \dots, s_d \in [0,1]$ so that for any choice of $\epsilon = (\epsilon_1, \dots, \epsilon_d)^d \in \{-1,1\}^d$, there is $h_\epsilon \in H$ so that

$$\forall i \in [d], \qquad \epsilon_i \cdot (h_\epsilon(x_i) - s_i) \geq \frac{\alpha}{2}$$



Definition (fat-shattering dimension): For $\alpha \in [0,1]$, $\text{fat}_\alpha(H)$ is the largest number of points H can shatter at scale α.

- **VC dimension** defined as: $\text{VC}(H) = \lim_{\alpha \to 0} \text{fat}_\alpha(H)$
  - Note: if H is binary-valued, $\text{fat}_\alpha(H) = \text{fat}_{\alpha'}(H)$ for all $\alpha, \alpha' \in (0,1)$
- **Fact**: a class H is learnable in i.i.d. setting iff $\text{fat}_\alpha(H) < \infty$ for all α

Learner	Adversary
Chooses $h_t \in H$	Chooses $p_t \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_t \sim p_t$, y_t adversarially

Minimax regret for online smoothed learning

- Prior work [HRS, '21] for binary classes H:

$$\mathbb{E}[\mathrm{Reg}_T] \lesssim \sqrt{T \cdot \mathrm{VCdim}(H) \cdot \log(1/\sigma)}$$

- Above is (nearly) tight [HRS, '21]

Theorem [ours]: Fix some $p \leq 2$. Consider any real-valued class H so that $\mathrm{fat}_\alpha(H) \leq d \cdot \alpha^{-p}$ for all $\alpha > 0$. Then there is some algorithm with:

$$\mathbb{E}[\mathrm{Reg}_T] \lesssim \sqrt{Td} \cdot \log(1/\sigma)$$

- **Note**: we get rates for $p > 2$ as well: scaling with T is $T^{1-1/p}$ (optimal rate even in adversarial setting)

Learner	Adversary
Chooses $h_t \in H$	Chooses $p_t \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_t \sim p_t$, y_t adversarially

Proof overview

Lemma 1 (coupling; slight generalization of [HRS,'21]; informal):
Fix $T, k \in \mathbb{N}$. For any adaptive σ-smooth adversary producing $x_t \sim p_t$, there is a coupling between $(x_1, \dots, x_T)$ and random variables $Z_t^j \in X, t \in [T], j \in [k]$ so that:

1. Marginal of $(x_1, \dots, x_T)$ is according to the smooth adversary;
2. Marginal of $\{Z_t^j\}$ is i.i.d. from μ;
3. With probability $1 - T(1-\sigma)^k$, $x_t \in \{Z_t^1, \dots, Z_t^k\}$ for all t

- Take $k \sim \log(T)/\sigma$ to make failure probability in last line negligible.
- High level takeaway: "effectively reduce" domain size to $T \cdot k \sim T/\sigma$

Learner	Adversary
Chooses $h_t \in H$	Chooses $p_t \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_t \sim p_t$, y_t adversarially

Proof overview, cont.

Lemma 1 (coupling; slight generalization of [HRS,'21]; informal): There is a coupling between $(x_1, \ldots, x_T)$ and random variables $Z_t^j \in X$, $t \in [T], j \in [k]$ so that:

1. Marginal of $(x_1, \ldots, x_T)$ is according to the smooth adversary;
2. Marginal of Z_t^j is i.i.d. from μ;
3. With probability $1 - T(1-\sigma)^k$, $x_t \in \{Z_t^1, \ldots, Z_t^k\}$ for all t

Lemma 2: There are constants $c, C > 0$ so that for any function class H on X, we have

$$\mathbf{sfat}_\alpha(H) \le \mathbf{fat}_{c\alpha}(H) \cdot \log^{1.01} \frac{C \cdot |X|}{\mathrm{fat}_{c\alpha}(H) \cdot \alpha}$$

- Lemma 1 implies domain is "effectively" small;
- Lemma 2 implies that online learning is no harder than offline learning when domain is small

Overview of our contributions

1. Tight regret upper bound of learning a real-valued class in smoothed online setting
    - Extends result of [HRS, ‘21] treating binary-valued setting
2. **Oracle-efficient** upper bound for learning a real-valued class in smoothed online setting
    - Dependence on smoothness parameter σ is exponentially worse than above upper bound.
3. Lower bound showing that regret of oracle-efficient algorithm cannot be significantly improved
    - Establishes computational-statistical gap for smoothed online learning

Oracle-efficiency in online learning

- Standard way of getting explicit online learning algorithms: construct an ϵ-cover of H, use Hedge on the cover
  - Issue: cover is exponentially large (e.g., in VCdim), so inefficient!
- Our approach: assume access to an **empirical risk minimization oracle:**

Definition (ERM oracle): An ERM oracle takes as input:

- Sequence $(x_1, y_1), \ldots, (x_m, y_m) \in X \times [0,1]$ of data points
- Sequence $w_1, \ldots, w_m \in \mathbb{R}$ of weights
- Sequence $\ell_1, \ldots, \ell_m : [0,1] \times [0,1] \to [0,1]$ of (convex) loss functions;

ERM oracle outputs

$$\hat{h} = \mathrm{argmin}_{h \in H} \sum_{i=1}^{m} w_i \cdot \ell_i(h(x_i), y_i)$$

Oracle-efficient algorithms: prior work

- Generic way of using ERM oracle: **follow-the-perturbed-leader (FTPL)** [KV,'05], [Hannan,'57]
- At each round t, given past sequence $(x_1, y_1), \dots, (x_{t-1}, y_{t-1})$ choose

$$h_t := \mathrm{argmin}_{h \in H} \sum_{s=1}^{t-1} \ell(h(x_s), y_s) + \omega(h)$$

Noise process (random mapping from hypotheses to reals)

- Originally [KV,'05]: $\omega(h)$ are independent for each h (inefficient!)
- Follow-ups (e.g., [DHLSSV,'17]): efficient algs. for special cases
- Lower bound in general [HK,'16]: ***need computation $\Omega(\sqrt{|H|})$ for worst-case adversary (even with ERM oracle)***

Using smoothness to get oracle efficiency

Learner	Adversary
Chooses $h_t \in H$	Chooses $p_t \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_t \sim p_t$, y_t adversarially

- Fix hyperparameters n, η
- Learner's procedure at each round t:
  1. Draw $Z_1, \dots, Z_n \sim \mu$ i.i.d
  2. Draw $\gamma_1, \dots, \gamma_n \sim N(0,1)$ i.i.d. standard Gaussians
  3. Choose $h_t \coloneqq \mathrm{argmin}_{h \in H} \sum_{s=1}^{t-1} \ell(h(x_s), y_s) + \underbrace{\eta \cdot \sum_{i=1}^{n} \gamma_i \cdot h(Z_i)}_{\omega(h)}$

Theorem [ours]: Fix some $p \leq 2$. Consider any real-valued class H so that $\mathrm{fat}_\alpha(H) \leq \alpha^{-p}$ for all $\alpha > 0$. Then above algorithm has

$$\mathbb{E}[\mathrm{Reg}_T] \lesssim T^{2/3} \cdot \sigma^{-1/3}$$

Further results for our FTPL algorithm

Theorem [ours]: Fix some $p \le 2$. Consider any real-valued class H so that $\text{fat}_\alpha(H) \le \alpha^{-p}$ for all $\alpha > 0$. Then our algorithm has

$$\mathbb{E}[\text{Reg}_T] \lesssim T^{2/3} \cdot \sigma^{-1/3}$$

- **Note**: For $p > 2$, we get regret scaling as $T^{1-\frac{1}{3(p-1)}}$
- Get optimal $\sqrt{T}$ scaling for binary classes:

Theorem [ours]: Consider any binary-valued class H. Then above algorithm has

$$\mathbb{E}[\text{Reg}_T] \lesssim \sqrt{T \cdot \text{VCdim}(H)/\sigma}$$

- Comparison with [HHSY, '22]: they get better smoothness scaling ($\sigma^{-1/4}$ as opposed to our $\sigma^{-1/2}$) for binary classes, but don't get any rates for nonparametric real-valued classes (i.e., when $\text{fat}_\alpha(H) \gg \log 1/\alpha$)

Learner	Adversary
Chooses $h_t \in H$	Chooses $p_t \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_t \sim p_t$, y_t adversarially

Proof overview

- **Step 1:** use standard technique to reduce to **non-adaptive adversary**: i.e., adversary chooses sequence $(x_1, y_1), \ldots, (x_T, y_T)$ without seeing algorithm's predictions
  - p_t still has to satisfy smoothness
  - *Why do this?* Can use a single draw of random process $\omega(\cdot)$ for all t

- **Step 2:** apply "Be the leader" lemma to get:

$$\mathbb{E}[\mathrm{Reg}_T] \leq \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \ell(h_t(x_t), y_t) - \ell(h_{t+1}(x_t), y_t)\right]}_{\text{Stability term}} + \mathrm{Complexity}(H)$$

Bound stability term by showing an upper bound on $\mathbb{P}[\|h_t - h_{t+1}\|_\mu \leq \alpha]$, for all $\alpha > 0$

Overview of our contributions

1. Tight regret upper bound of learning a real-valued class in smoothed online setting
   - Extends result of [HRS, ‘21] treating binary-valued setting
2. **Oracle-efficient** upper bound for learning a real-valued class in smoothed online setting
   - Dependence on smoothness parameter σ is exponentially worse than above upper bound.
3. Lower bound showing that regret of oracle-efficient algorithm cannot be significantly improved
   - Establishes computational-statistical gap for smoothed online learning

Statistical-computational gap

Learner	Adversary
Chooses $\boldsymbol{h_t} \in \boldsymbol{H}$	Chooses $p_t \in \mathrm{SMOOTH}_\sigma(\mu)$, draws $x_t \sim p_t$, y_t adversarially

- Consider binary classes H with $\mathrm{VCdim}(H) = d$:
  - There is an algorithm with $\mathbb{E}[\mathrm{Reg}_T] \lesssim \sqrt{T \cdot \mathrm{VCdim}(H) \cdot \log(1/\sigma)}$
  - Our oracle efficient algorithm has $\mathbb{E}[\mathrm{Reg}_T] \lesssim \sqrt{T \cdot \mathrm{VCdim}(H)/\sigma}$

Does oracle efficiency require having $\sigma^{-\Omega(1)}$ regret? **Yes!**

- **ERM oracle model** [HK, '16]: calling ERM oracle takes $O(1)$ time, as does listing each (x_i, y_i) in the dataset on which ERM oracle is called

Theorem [ours]: Fix any $T \in \mathbb{N}$ and $\sigma \in (0,1)$. No randomized **proper** algorithm can guarantee regret $o(T)$ against a σ-smooth adversary against classes H satisfying $|H| \leq 1/\sigma$ in time $o(1/\sqrt{\sigma})$ in ERM oracle model

- [HHSY, '22]: proved similar result to the above

Additional results

- We also exhibit an oracle-efficient *improper* algorithm that achieves better (optimal) regret dependence on T than our proper algorithm: if $\mathrm{fat}_\alpha(H) \leq \alpha^{-2}$ for all $\alpha > 0$, our improper algorithm has

$$\mathbb{E}[\mathrm{Reg}_T] \lesssim \sqrt{T/\sigma}$$

- Compare to $T^{2/3}$ scaling for proper algorithm
- Similar result in [HHSY, '22]

Future work

1. Oracle-efficient proper regret bound with optimal scaling on T?
2. Is there a stronger notion of smoothness that can get regret scaling with $\mathrm{poly}\log 1/\sigma$ for an oracle-efficient algorithm?
3. Can we get around the $\sigma^{-\Omega(1)}$ computational lower bound by using an **improper** learning algorithm?
4. Can we get fast (i.e., $o(\sqrt{T})$) rates for “nicer” loss functions? (e.g., square loss)
   - Of course, want scaling with the non-sequential fat-shattering dimension

Thank you for listening!