Group-wise oracle-efficient algorithms for online multi-group learning

Samuel Deng [1] Daniel Hsu [1] Jingwen Liu [1]

[1]Department of Computer Science, Columbia University

Problem Setup

Cast of characters. Consider a context space $\mathcal{X}$ and an action space $\mathcal{Y} = \{-1, 1\}$...

- "Benchmark" hypothesis class $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$ comprised of functions $h : \mathcal{X} \to \{-1, 1\}$.
- Collection of groups $\mathcal{G} \subseteq 2^{\mathcal{X}}$ comprised of functions $g : \mathcal{X} \to \{0, 1\}$ denoting membership for some subset of $\mathcal{X}$.
- Arbitrary bounded loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$.

NOTE: In this poster, we focus on binary actions, but this generalizes to discrete action spaces.

Online learning game. For rounds $t = 1, 2, 3, \ldots, T$:

1. Nature chooses $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and reveals x_t.
2. Learner chooses an action $\hat{y}_t \in \mathcal{Y}$.
3. Nature reveals $y_t \in \mathcal{Y}$.
4. Learner incurs loss $\ell(\hat{y}_t, y_t) \in [0, 1]$.

Motivation. Traditional online learning is concerned with *aggregate* $o(T)$ regret over the T rounds.

$$\mathrm{Reg}_T(\mathcal{H}) := \sum_{t=1}^{T} \ell(\hat{y}_t, y_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(h(x_t), y_t).$$

"Individual-level" regret guarantees are too strong to be feasible:

$$\mathrm{Reg}_T(\mathcal{H}, \{x\}) := \sum_{t=1}^{T} \mathbf{1}\{x_t = x\}\, \ell(\hat{y}_t, y_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbf{1}\{x_t = x\}\, \ell(h(x_t), y_t).$$

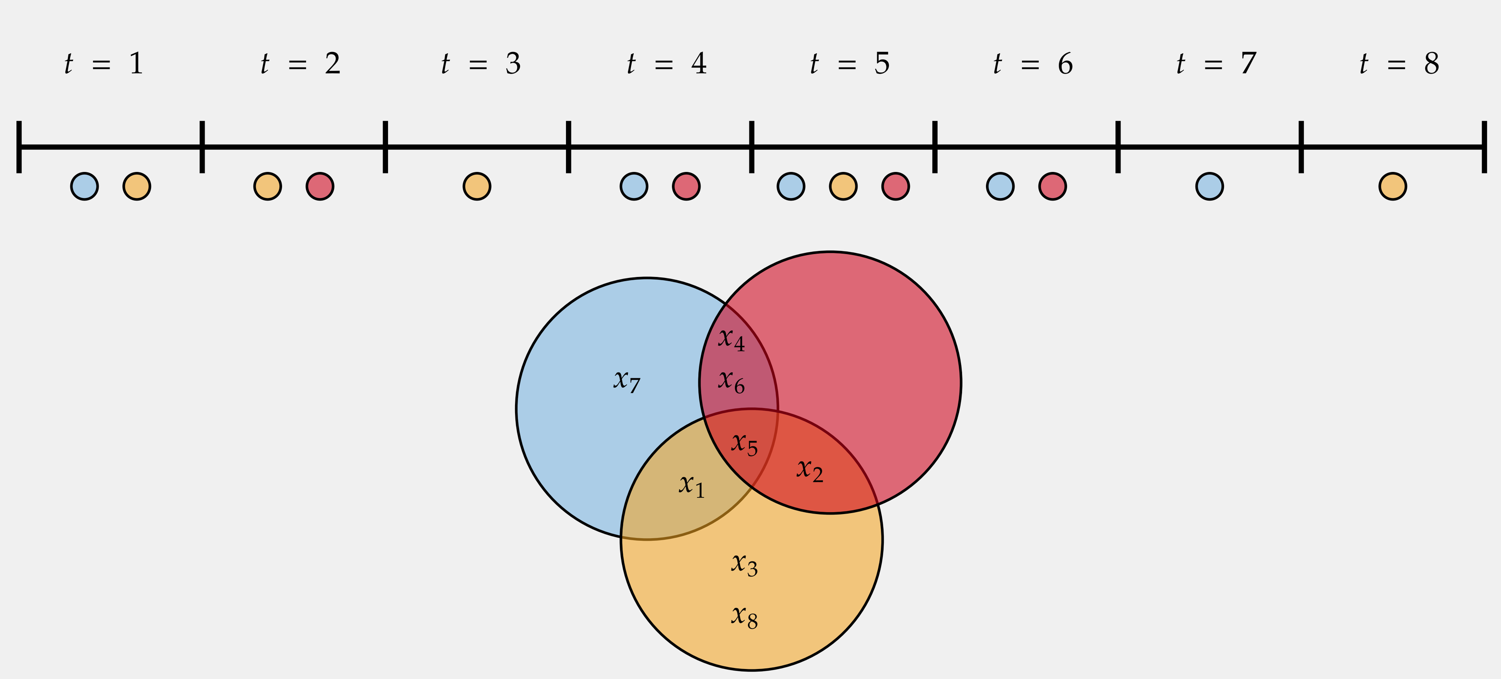
Online multi-group learning

A middle-ground between on-average and individual-level guarantees: consider a rich (possibly exponetnially large/infinite) collection of subsets of the input space, $\mathcal{G} \subseteq \mathcal{X}$, and consider:

$$\mathrm{Reg}_T(\mathcal{H}, g) := \sum_{t=1}^{T} g(x_t)\ell(\hat{y}_t, y_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} g(x_t)\ell(h(x_t), y_t).$$

Goal. Ensure that $\mathrm{Reg}_T(\mathcal{H}, g) = o(T)$ for *all* groups $g \in \mathcal{G}$ simultaneously.

Why is this interesting? The best hypothesis for different groups may differ. The groups may intersect in arbitrary ways, precluding running a separate no-regret algorithm on each $g \in \mathcal{G}$.



Infinite or Large Collections of Groups

Existing results for online multi-group learning assume finiteness/enumerability of either $\mathcal{H}$ or $\mathcal{G}$:

- *Blum & Lykouris (2020):* $\mathrm{Reg}_T(\mathcal{H}, g) = o(T)$ for all $g \in \mathcal{G}$ for *finite* $\mathcal{H}$ and $\mathcal{G}$.
- *Acharya et al. (2023):* $\mathrm{Reg}_T(\mathcal{H}, g) = o(T)$ for all $g \in \mathcal{G}$ with *finite* $\mathcal{G}$ and oracle for (infinite) $\mathcal{H}$.

Our main question. Can we ensure $\mathrm{Reg}_T(\mathcal{H}, g) = o(T)$ for all $g \in \mathcal{G}$ that is oracle-efficient in *both* $\mathcal{H}$ and $\mathcal{G}$? Can we deal with cases in which $\mathcal{G}$ is too large to possibly enumerate?

Assumptions

Assumption 0: Access to oracle. For $\alpha \geq 0$ and a sequence of m loss functions $\ell_i : (\{0,1\} \times \{-1,1\}) \times (\{-1,1\} \times \{-1,1\}) \to [-1,1]$ and weights $w_1, \ldots, w_m \in \mathbb{R}$, an *$\alpha$-approximate $(\mathcal{G}, \mathcal{H})$-optimization oracle* $\mathrm{OPT}^{\alpha}_{(\mathcal{G},\mathcal{H})}$ outputs a pair $(\tilde{g}, \tilde{h}) \in \mathcal{G} \times \mathcal{H}$ satisfying:

$$\sum_{i=1}^{m} w_i \ell_i((\tilde{g}(x_i), \tilde{h}(x_i)), (y_i, y_i')) \geq \sup_{(g^*, h^*) \in \mathcal{G} \times \mathcal{H}} \sum_{i=1}^{m} w_i \ell_i((g^*(x_i), h^*(x_i)), (y_i, y_i')) - \alpha.$$

Assumption 1: Smoothed adversary. Let $\mathcal{B}$ be a base measure on $\mathcal{X}$. A *σ-smooth distribution* μ on $\mathcal{X}$ is absolutely continuous with respect to $\mathcal{B}$ and satisfies

$$\operatorname{ess\,sup} \frac{d\mu}{d\mathcal{B}} \leq \frac{1}{\sigma}.$$

At each round $t \in [T]$, Nature fixes a σ-smooth distribution μ_t and samples $x_t \sim \mu_t$, still choosing $y_t \in \mathcal{Y}$ adversarially.

Assumption 2: Existence of good perturbation matrix. Let $\gamma > 0$. For finite $\mathcal{G}$ and $\mathcal{H}$, there exists a matrix $\Gamma \in [-1, 1]^{|\mathcal{G}||\mathcal{H}| \times N}$, such that:

1. *γ-approximable.* For all $(g, h) \in \mathcal{G} \times \mathcal{H}$ and $(x, y', y) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$, there exists $s \in \mathbb{R}^N$ with $\|s\|_1 \leq \gamma$ such that
$$\langle \Gamma^{(g,h)} - \Gamma^{(g',h')}, s \rangle \geq \tilde{\ell}_x((g, h), (y', y)) - \tilde{\ell}_x((g', h'), (y', y)) \quad \text{for all } (g', h') \in \mathcal{G} \times \mathcal{H}.$$
2. *γ-implementable.* For each column $j \in [N]$, there exists a dataset S_j with $|S_j| \leq M$ such that, for all pairs of rows $(g, h), (g', h') \in \mathcal{G} \times \mathcal{H}$,
$$\Gamma^{((g,h),j)} - \Gamma^{((g',h'),j)} = \sum_{(w,(x,y,y')) \in S_j} w \left(\tilde{\ell}_x((g, h), (y', y)) - \tilde{\ell}_x((g', h'), (y', y)) \right)$$

Main Theorems

Theorem (Smoothed Setting). Under **Assumption 1** and $\mathcal{H}$ and $\mathcal{G}$ with VC dimension $d < \infty$, with $M = \mathrm{poly}(T), n = \mathrm{poly}(T/\sigma)$, and $\eta = \mathrm{poly}(T/\sigma)$, Algorithm 1 achieves, for each $g \in \mathcal{G}$:

$$\mathbb{E}[\mathrm{Reg}_T(\mathcal{H}, g)] \leq O\left(\sqrt{\frac{dT \log T}{\sigma}} + \alpha T \right).$$

Theorem (Existence of approximable and implementable perturbations). Under a $\Gamma \in [-1, 1]^{|\mathcal{G}||\mathcal{H}| \times N}$ with $\gamma > 0$ in **Assumption 2** and finite $\mathcal{H}$ and $\mathcal{G}$, there exists an algorithm that achieves, for each $g \in \mathcal{G}$:

$$\mathbb{E}[\mathrm{Reg}_T(\mathcal{H}, g)] \leq O\left(\sqrt{T_g} \max\left\{ \gamma, \log |\mathcal{H}||\mathcal{G}|, \sqrt{N \log |\mathcal{H}||\mathcal{G}|} \right\} + \alpha T \right)$$

Algorithm (for the smoothed setting)

Main idea. For any $x \in \mathcal{X}$, the *single-round regret* of the Learner on group g to the hypothesis h is

$$\tilde{\ell}_x((g, h), (y', y)) := g(x) \left(\ell(y', y) - \ell(h(x), y) \right).$$

The algorithm is a sequential game between two competing players:

- **$(\mathcal{G}, \mathcal{H})$-player.** Employs $(\mathcal{G}, \mathcal{H})$-optimization oracle and follow-the-perturbed-leader (FTPL) style algorithm to play a distribution over $\mathcal{G} \times \mathcal{H}$ that maximizes single-round regret of the $\mathcal{H}$-player. Maintains an *implicit* distribution on $\mathcal{G} \times \mathcal{H}$ through FTPL.
- **$\mathcal{H}$-player.** Receives (an approximation of a) distribution over $\mathcal{G} \times \mathcal{H}$ from $(\mathcal{G}, \mathcal{H})$-player and solves an LP to choose $\hat{y}_t$ randomly.

The perturbations for the $(\mathcal{G}, \mathcal{H})$-player in Algorithm 1 are:

$$\pi^{\mathrm{bin}}_{t,n}(g, h, \eta) := \sum_{j=1}^{n} \frac{\eta \gamma_{t,j} g(z_{t,j}) h(z_{t,j})}{\sqrt{n}}, \quad \text{where } z_{t,j} \sim \mathcal{B} \text{ and } \gamma_{t,j} \sim N(0, 1)$$

Algorithm 1 Algorithm for Group-wise Oracle Efficiency (for smoothed online learning)

Input: Perturbation strength $\eta > 0$; perturbation count $n \in \mathbb{N}$; number of oracle calls $M \in \mathbb{N}$.

1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: Receive a context $x_t \sim \mu_t$ from Nature.
3: **for** $i = 1, 2, 3, \ldots, M$ **do**
4: **$(\mathcal{G}, \mathcal{H})$-player:** Draw n hallucinated examples as in Equation (4) to construct $\pi^{\mathrm{bin}}_{t,n}$.
5: **$(\mathcal{G}, \mathcal{H})$-player:** Using the entire history $\{(\hat{y}_s, y_s)\}_{s=1}^{t-1}$ so far, call $\mathrm{OPT}^{\alpha}_{(\mathcal{G},\mathcal{H})}$ to obtain $(\tilde{g}^{(i)}_t, \tilde{h}^{(i)}_t) \in \mathcal{G} \times \mathcal{H}$ satisfying:

$$\sum_{s=1}^{t-1} \tilde{\ell}_{x_s}((\tilde{g}^{(i)}_t, \tilde{h}^{(i)}_t), (\hat{y}_s, y_s)) + \pi^{\mathrm{bin}}_{t,n}(\tilde{g}^{(i)}_t, \tilde{h}^{(i)}_t, \eta)$$
$$\geq \sup_{(g^*, h^*) \in \mathcal{G} \times \mathcal{H}} \sum_{s=1}^{t-1} \tilde{\ell}_{x_s}((g^*, h^*), (\hat{y}_s, y_s)) + \pi^{\mathrm{bin}}_{t,n}(g^*, h^*, \eta) - \alpha \quad (5)$$

6: **end for**
7: **$\mathcal{H}$-player:** Call $\mathrm{OPT}_{\mathcal{H}}$ twice on the singleton datasets $\{(x_t, 1)\}$ and $\{(x_t, -1)\}$, with the 0-1 loss, obtaining:

$$h_1' \in \arg\min_{h^* \in \mathcal{H}} \mathbf{1}\{h^*(x_t) \neq 1\}, \quad h_{-1}' \in \arg\min_{h^* \in \mathcal{H}} \mathbf{1}\{h^*(x_t) \neq -1\}.$$

8: **$\mathcal{H}$-player:** Solve the linear program

$$\min_{p, \lambda \in \mathbb{R}} \quad \lambda$$
$$\text{subj. to} \quad \sum_{i=1}^{M} p \tilde{\ell}_{x_t}((\tilde{g}^{(i)}_t, \tilde{h}^{(i)}_t), (h_1'(x_t), y)) + (1 - p) \tilde{\ell}_{x_t}((\tilde{g}^{(i)}_t, \tilde{h}^{(i)}_t), (h_{-1}'(x_t), y)) \leq \lambda$$
$$\forall y \in \{-1, 1\}$$
$$0 \leq p \leq 1.$$

9: Sample $b \sim \mathrm{Ber}(p)$ where $b \in \{-1, 1\}$, let $h_t = h_b'$.
10: Learner commits to the action $\hat{y}_t = h_t(x_t)$; Nature reveals y_t.
11: Learner incurs the loss $\ell(\hat{y}_t, y_t)$.
12: **end for**

samdeng@cs.columbia.edu, djhsu@cs.columbia.edu, jingwenliu@cs.columbia.edu