Online Learning with Adversarial Delays

Kent Quanrud and Daniel Khashabi

{quanrud2,khashab2}@illinois.edu

Neural Information Processing Systems Foundation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Online Learning

For each round $t = 1, \ldots, T$, we pick a point $x_t \in K$, and an adversary picks a cost function f_t. We incur the **loss** $f_t(x_t)$. The **regret** of our strategy is the difference between our total loss and the total loss of the best fixed point in hindsight:

$$R(T) = \sum_{t=1}^{T} f_t(x_t) - \arg\min_{x \in K} \sum_{t=1}^{T} f_t(x).$$

The goal is to minimize the regret $R(T)$.

Abstract

We study the performance on standard online learning algorithm when the feedback is **delayed by an adversary**. We obtain:

- $O(\sqrt{D})$ regret bounds for `online-gradient-descent`, and
- $O(\sqrt{D})$ regret bounds for `follow-the-perturbed-leader`,

where D is the sum of delays. In both cases the algorithms are left essentially unmodified.

Delayed Feedback Model

- $d_t \in \mathbb{Z}^+$ denotes a non-negative **delay**. Feedback from round t is delivered at the end of round $t + d_t - 1$ and can be used in round $t + d_t$.
- $\mathcal{F}_t = \{u \in [T] : u + d_u - 1 = t\}$ denotes the set of rounds whose feedback appears at the end of round t.
- $D = \sum_{t=1}^{T} d_t$ denotes the sum of all delays. In the standard setting with no delays, $D = T$.

Delays in Learning

The standard models assume that the adversary gives us the loss function f_t *before* we select the next point x_{t+1}. What if the feedback is delayed? For example:

- Online advertising algorithms serve many ads before finding out which ones are clicked.
- Online algorithms planning resource allocation in the cloud cannot wait for one batch job to end before launching the next.
- In finance, online learning algorithms managing portfolios are subject to information and transaction delays from the market.
- Optimization algorithms implemented in distributed or parallel environments suffer communication delays between asynchronous processors.

`online-gradient-descent`

Convex setting:

Convex domain K, convex loss functions $\{f_t : \mathbb{R}^n \to \mathbb{R}\}$

Undelayed algorithm and regret bound: [Zinkevich, 2003]

$$x_{t+1} = \pi_K\left(x_t - \Theta\left(\frac{1}{\sqrt{T}}\right) f_t'(x_t)\right),$$

where π_K projects to nearest point in K.

$$\Rightarrow \sum_{t=1}^{T} f_t(x_t) \leq \arg\min_{x \in K} \sum_{t=1}^{T} f_t(x) + O(\sqrt{T}).$$

Delayed algorithm:

$$x_{t+1} = \pi_K\left(x_t - \Theta\left(\frac{1}{\sqrt{D}}\right) \sum_{s \in \mathcal{F}_t} f_s'(x_s)\right)$$

(!) Same as undelayed algorithm when $\mathcal{F}_t = \{t\}$

Delayed regret bound:

$$\sum_{t=1}^{T} f_t(x_t) \leq \arg\min_{x \in K} \sum_{t=1}^{T} f_t(x) + O(\sqrt{D})$$

(!) Matches undelayed regret bound when $D = T$

`follow-the-perturbed-leader`

Discrete setting:

Discrete domain K, cost vectors $\{c_t \in \mathbb{R}^n\}$

Undelayed algorithm and regret bound: [Kalai and Vempala, 2005]

$$x_{t+1} = \arg\min_{x \in K} c_0 \cdot x + \sum_{s=1}^{t} c_s \cdot x,$$

where $c_0 \sim [0, \Theta(\sqrt{T})]^n$ uniformly at random.

$$\Rightarrow \sum_{t=1}^{T} c_t \cdot x_t \leq \arg\min_{x \in K} \sum_{t=1}^{T} c_t \cdot x + O(\sqrt{T})$$

Delayed algorithm:

$$x_{t+1} = \arg\min_{x \in K} c_0 \cdot x + \sum_{s=1}^{t} \sum_{r \in \mathcal{F}_s} c_r \cdot x$$

where $c_0 \sim [0, \Theta(\sqrt{D})]^n$ uniformly at random.

(!) Same as undelayed algorithm when $\mathcal{F}_t = \{t\}$

Delayed regret bound:

$$\sum_{t=1}^{T} c_t \cdot x_t \leq \arg\min_{x \in K} \sum_{t=1}^{T} c_t \cdot x + O(\sqrt{D})$$

(!) Matches undelayed regret bound when $D = T$

Extensions

- $O(\sqrt{D})$ regret bound for `online-mirror-descent`, a generalization of `online-gradient-descent` and randomized expert selection by exponential weights.
- $O(\sqrt{D})$ regret bound for `follow-the-lazy-leader`, a variation of `follow-the-perturbed-leader` for switching costs.

Selected References

A. Kalai and S. Vempala. Efficient algorithms for online decision problems. **J. Comput. Sys. Sci.**, 71:291–307, 2005. Extended abstract in Proc. 16th Ann. Conf. Comp. Learning Theory *(COLT)*, 2003.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In **Proc. 20th Int. Conf. Mach. Learning** *(ICML)*, pages 928–936, 2003.