Lipschitz Adaptivity with Multiple Learning Rates in Online Learning

Zakaria Mhammedi Wouter M. Koolen Tim van Erven

Australian National University DATA 61 CWI Centrum Wiskunde & Informatica Universiteit Leiden

Contribution

In online convex optimization and experts problems, **second-order regret bounds** imply **adaptive algorithms**. Current methods **require knowledge** of the **Lipschitz constant**; We efficiently **learn** it.

Abstract

To get good performance in Online Convex Optimization (OCO) you need to **select and tune your algorithm** based on lots of technical stuff. The MetaGrad and Squint algorithms (OCO/experts) promise to overcome this difficulty by maintaining **multiple learning rates**.

Guarantees: MetaGrad and Squint are robust to worst-case losses and exploit stochastic data (Bernstein). MetaGrad automatically adapts to curvature (strong-convexity, exp-concavity).

Limitation: MetaGrad and Squint require prior knowledge of a bound on the gradients/losses; they fail otherwise.

OCO and Experts Settings

For online convex optimization:

1: **for** $t = 1, 2, \dots, T$ **do**
2: Learner plays $\hat{\boldsymbol{u}}_t$ in convex body $\mathcal{U} \subset \mathbb{R}^d$
3: Environment reveals convex loss function $\ell_t : \mathcal{U} \to \mathbb{R}$
4: Learner incurs loss $\ell_t(\hat{\boldsymbol{u}}_t)$, observes gradient $\boldsymbol{g}_t = \nabla \ell_t(\hat{\boldsymbol{u}}_t)$
5: **end for**

Measure **regret** w.r.t. $\boldsymbol{u} \in \mathcal{U}$: $\text{Regret}_T^{\boldsymbol{u}} = \sum_{t=1}^T \ell_t(\hat{\boldsymbol{u}}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u})$.

The experts setting is the special case with *probability simplex* domain $\mathcal{U} = \triangle_K$ and *linear losses* $\ell_t(\boldsymbol{u}) := \langle \boldsymbol{u}, \boldsymbol{l}_t \rangle$, where $\boldsymbol{l}_t \in \mathbb{R}^K$.

State-of-the-Art Second-Order Bounds

Bounds (1) and (2) are achieved respectively by MetaGrad [Van Erven and Koolen, 2016] and Squint [Koolen and Van Erven, 2015]

$$\text{OCO:}\quad O\left(\sqrt{V_T^{\boldsymbol{u}} d \log T}\right) \forall \boldsymbol{u}, \qquad V_T^{\boldsymbol{u}} = \sum_{t=1}^T \langle \hat{\boldsymbol{u}}_t - \boldsymbol{u}, \boldsymbol{g}_t \rangle^2, \quad (1)$$

$$\text{Experts:}\quad O\left(\sqrt{\mathop{\mathbb{E}}_{\rho(k)}[V_T^k] \operatorname{KL}(\rho\|\pi)}\right) \forall \rho, \qquad V_T^k = \sum_{t=1}^T \langle \hat{\boldsymbol{u}}_t - \boldsymbol{e}_k, \boldsymbol{l}_t \rangle^2, \quad (2)$$

under the (standard) assumption that $\|\boldsymbol{g}_t\|_2 \le 1$ and $\|\boldsymbol{l}_t\|_\infty \le 1$, for all $t \in [T]$ (*i.e.* known "**Lipschitz bound**").
MetaGrad and Squint achieve these regrets by optimizing **exp-concave** surrogate losses and maintaining multiple learning rates.

Towards Lipschitz Adaptivity via Clipping

Let (B_t) be the sequence of **observed "Lipschitz" values**:

$$B_t := B \vee \max_{t \le T} b_t, \quad \text{where } b_t := \begin{cases} D\|\boldsymbol{g}_t\|_2, & \text{for OCO,} \\ \max_k \langle \hat{\boldsymbol{u}}_t - \boldsymbol{e}_k, \boldsymbol{l}_t \rangle, & \text{for Experts.} \end{cases}$$

When no bound on $(\|\boldsymbol{g}_t\|_2)$ or $(\|\boldsymbol{l}_t\|_\infty)$ is known in advance, we develop MetaGrad+C and Squint+C which, respectively, observe the sequence of **clipped** gradients and loss vectors [Cutkosky, 2019]:

$$\bar{\boldsymbol{g}}_t := \boldsymbol{g}_t \cdot B_{t-1}/B_t, \qquad \bar{\boldsymbol{l}}_t := \boldsymbol{l}_t \cdot B_{t-1}/B_t.$$

Theorem 1. *With initial estimate B of the "Lipschitz" bound, MetaGrad+C [resp. Squint+C] guarantees the regret bound (1) [resp. (2)] with the following **overhead** multiplying $V_T^{\boldsymbol{u}}$ [resp. V_T^k]:*

$$O\left(\ln\ln \frac{\sqrt{\sum_{t=1}^T b_t^2}}{B}\right), \quad \textit{for MetaGrad+C,} \quad (3)$$

$$O\left(\ln\ln \frac{B_T}{B}\right), \quad \textit{for Squint+C.} \quad (4)$$

Limitation of the Clipping Trick

The overheads in (3) and (4) incurred by MetaGrad+C and Squint+C make their respective regret bounds **non-homogeneous**: scaling the losses/gradients by a factor $c > 0$ would not scale the bound by the same factor.

There does not seem to be any safe a-priori way to tune B. If we set it too small, the factors $\ln\ln\left(\sqrt{\sum_{t=1}^T b_t^2}/B\right)$ and $\ln\ln(B_{T-1}/B)$ **explode**. If we set it too large—much larger than the effective range of the data—then the lower-order contribution in the bounds **blows up**.

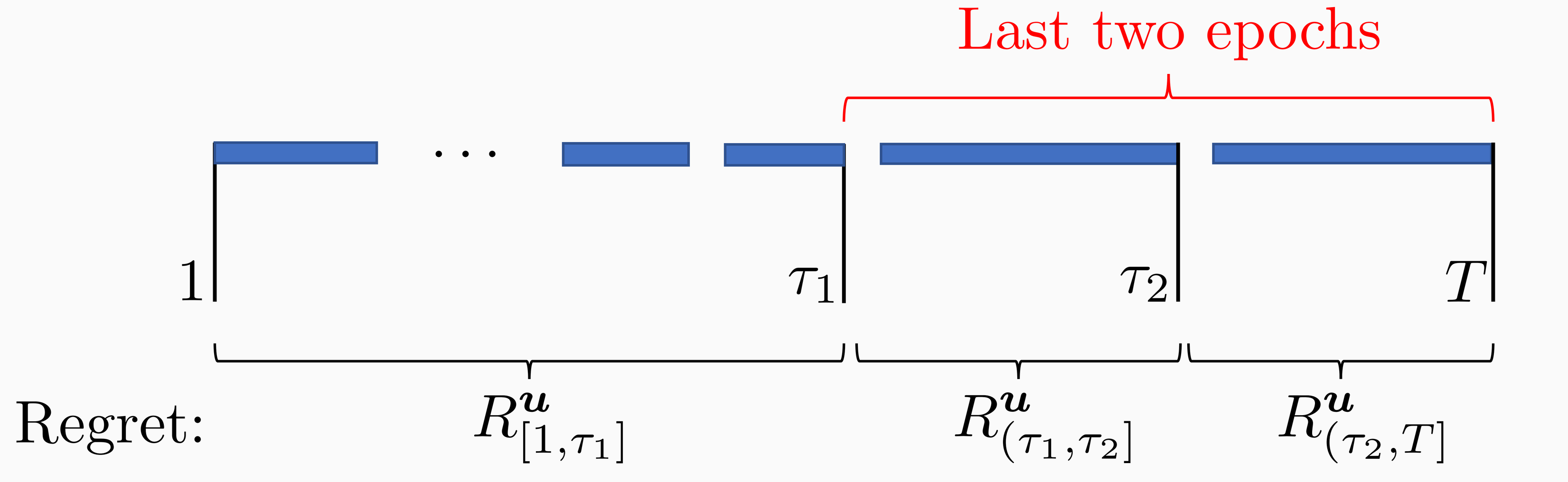
Lipschitz Adaptivity via a Novel Restart Trick

Consider the following algorithm where Alg is either MetaGrad+C or Squint+C, taking as input parameter an initial scale B;

1: Play **0** for OCO or π for experts until the first time $t = \tau_1$ that $b_t \neq 0$;
2: Run Alg with input $B = B_{\tau_1}$ until the first time $t = \tau_2$ that $\frac{B_t}{B_{\tau_1}} > \sum_{s=1}^t \frac{b_s}{B_s}$;
3: Set $\tau_1 = \tau_2$ and goto line 2;

Theorem 2. *Let MetaGrad+L [resp. Squint+L] be the application of the above algorithm with Alg being MetaGrad+C [resp. Squint+C]. Then MetaGrad+L [resp. Squint+L] guarantees the same regret as in (3) [resp. (4)] with **small constant overhead**, and without prior knowledge of a Lipschitz bound.*

Idea Behind the Restart Trick



Regret Decomposition. Given a comparator $\boldsymbol{u} \in \mathcal{U}$, the regret can be decomposed in tree parts (see picture above):

$$\text{Regret}_T^{\boldsymbol{u}} = R^{\boldsymbol{u}}_{[1,\tau_1]} + R^{\boldsymbol{u}}_{(\tau_1,\tau_2]} + R^{\boldsymbol{u}}_{(\tau_2,T]}.$$

For the **last two epochs**, the regret bound for MetaGrad+C [resp. Squint+C] applies, and thus $R^{\boldsymbol{u}}_{(\tau_1,\tau_2]}$ and $R^{\boldsymbol{u}}_{(\tau_2,T]}$ are both of the order of (1) [resp. (2)] with the additional **overhead** in (3) [resp. (4)], which is **at most** $\ln\ln T^{3/2}$ [resp. $\ln\ln T$] due to the restart condition.
The regret for the **earlier epochs** adds up to a lower order term:

$$R^{\boldsymbol{u}}_{(1,\tau_1]} \le \sum_{t=2}^{\tau_1} b_t \le B_{\tau_1} \sum_{t=1}^{\tau_1} \frac{b_t}{B_t} \le B_{\tau_1} \sum_{t=1}^{\tau_2} \frac{b_t}{B_t} \overset{*}{<} B_{\tau_2} \le B_T,$$

where $\overset{*}{<}$ follows from the restart condition.

Efficient Handling of Domain Constraints

MetaGrad requires $O(\log T)$ many projections onto $\mathcal{U}$ in the **Mahalanobis distance** at each round: one projection per output of a slave algorithm. We make use of a recent reduction for constrained optimization due to Cutkosky and Orabona [2018] to incur only a **single Euclidean projection** onto $\mathcal{U}$ while preserving the regret bound in (1).