

# Isotuning With Applications To Scale-Free Online Learning

Laurent Orseau

Marcus Hutter

{lorseau,mhutter}@google.com  
DeepMind, London, UK

## Abstract

We extend and combine several tools of the literature to design fast, adaptive, anytime and scale-free online learning algorithms.<sup>1</sup> Scale-free regret bounds must scale linearly with the maximum loss, both toward large losses and toward very small losses. Adaptive regret bounds demonstrate that an algorithm can take advantage of easy data and potentially have constant regret. We seek to develop fast algorithms that depend on as few parameters as possible, in particular they should be anytime and thus not depend on the time horizon. Our first and main tool, *isotuning*, is a generalization of the idea of balancing the trade-off of the regret. We develop a set of tools to design and analyze such learning rates easily and show that they adapts automatically to the rate of the regret (whether constant,  $O(\log T)$ ,  $O(\sqrt{T})$ , etc.) within a factor 2 of the optimal learning rate in hindsight for the same observed quantities. The second tool is an *online correction*, which allows us to obtain centered bounds for many algorithms, to prevent the regret bounds from being vacuous when the domain is overly large or only partially constrained. The last tool, *null updates*, prevents the algorithm from performing overly large updates, which could result in unbounded regret, or even invalid updates. We develop a general theory using these tools and apply it to several standard algorithms. In particular, we (almost entirely) restore the adaptivity to small losses of FTRL for unbounded domains, design and prove scale-free adaptive guarantees for a variant of Mirror Descent (at least when the Bregman divergence is convex in its second argument), extend Adapt-ML-Prod to scale-free guarantees, and provide several other minor contributions about Prod, AdaHedge, BOA and Soft-Bayes.

## 1 INTRODUCTION

In online convex optimization (Hazan, 2016), at each round  $t = 1, 2, \dots$ , the learner selects a point  $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^N$  and incurs a loss  $\ell_t(\mathbf{x}_t)$ , with  $\ell_t : \mathcal{X} \rightarrow \mathbb{R}$  convex.

<sup>1</sup>This paper is an extended version from an AISTATS 2022 submission.

We also use  $\ell_t \in \mathbb{R}^N$  as a loss vector, with incurred loss  $\langle \mathbf{x}_t, \ell_t \rangle$  when considering online linear optimization. The objective of the learner is to minimize its cumulative loss over  $T$  rounds (with  $T$  unknown in advance), and the difference with the cumulative loss incurred by the best constant point  $\mathbf{x}^* \in \mathcal{X}$  in hindsight is called the regret. Consider the following simplified regret bound (e.g., online gradient descent, Zinkevich (2003)) featuring a time-varying learning rate  $\{\eta_t\}_{t \in [T]}$ :

$$\mathcal{R}_T(\mathbf{x}^*) \leq \frac{q}{\eta_T} + \sum_{t \in [T]} \eta_t a_t. \quad (1)$$

We would like to design a sequence of (adaptive) learning rates  $\eta_1, \eta_2, \dots$  without knowledge of the time horizon  $T$  and with the  $a_t$  discovered sequentially. The usual process of the researcher is to first replace the learning rate with a constant one which minimizes the trade-off  $q/\eta + \eta \sum_t a_t$ , that is,  $\eta = \sqrt{q/\sum_t a_t}$  in hindsight. The second step is to make it time varying:  $\eta_t = \sqrt{q/\sum_{s \leq t} a_s}$ . The last step is to analyze the regret, which often involves ad-hoc lemmas such as  $\sum_t a_t / \sqrt{\sum_{s \leq t} a_s} \leq 2\sqrt{\sum_t a_t}$  (e.g., Auer et al. (2002, Lemma 3.5), Orabona and Pál (2018, Lemma 3), Pogodin and Lattimore (2020, Lemma 4.8)). Generalizing the AdaHedge learning rate scheme from de Rooij et al. (2014), we propose instead to tune the learning rate sequentially so as to *equate* the two terms of the regret:  $q/\eta_T = \sum_t \eta_t a_t$ . We call this method *isotuning*. Hence, instead of having to bound terms of the regret in interaction with a time-varying learning rate, we now merely have to bound  $2q/\eta_T$ , that is, only the last learning rate. We give a general theorem that provides a bound that is always within a factor 2 of the bound that uses the constant optimal learning rate for the same observed quantities, independently of the regime of the regret.

In this paper we are particularly interested in designing learning algorithms with the following properties: (i) anytime: the horizon  $T$  is unknown, (ii) scale-free: the losses are in  $\mathbb{R}^N$  for  $N \in \mathbb{N}_1$  are only sequentially revealed, the range of the losses is unknown in advance, the regret bound must scale linearly with the loss range (whether large or very small), (iii) adap-

tive: the regret bound should feature a main term like  $\sqrt{\sum_t \|\ell_t\|^2}$  (instead of  $L\sqrt{T}$  with  $L = \max_t \|\ell_t\|$ ) so that a smaller bound can be shown for easy problems and the algorithm converges faster, (iv) fast: we are only interested in algorithms that have  $O(N)$  computation complexity per round.

While a desirable property, scale-free bounds and algorithms are not so easy to obtain (see for example the discussions in Orabona and Pál (2018); Mhammedi and Koolen (2020)): in particular even a constant additive term makes the bound vacuous for very small losses, as  $LT$  may be much smaller than 1. Prod and AdaHedge are possibly the first adaptive scale-free algorithms for the probability simplex (Cesa-Bianchi et al., 2007; de Rooij et al., 2014). Important progress has been made by Orabona and Pál (2018), who propose a Follow The Regularized Leader (FTRL) algorithm (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2007) with a scale-free bound for any convex regularizer for both bounded and unbounded domains. Its main drawback is that while it is adaptive to small loss vectors  $\ell_t$  with a bound of the form  $O(\sqrt{\sum_t \|\ell_t\|_*^2})$  for constrained domains, this adaptivity is lost for unconstrained domains and the bound degrades to  $O(L\sqrt{T})$ . Orabona and Pál (2018) also propose a scale-free variant of Mirror Descent (MD) (Beck and Teboulle, 2003), but it requires a bound on the maximum Bregman divergence between any two points, which excludes the most interesting cases (‘AdaHedge’-like and unconstrained ‘gradient descent’-like)—they also provide lower bounds on this algorithm demonstrating its failure cases. (See more discussion in Section 5.)

In this paper we combine and improve several tools of the literature (isotuning, online correction, null updates) to help design algorithms that fit our requirements detailed above. In particular, we derive a new FTRL algorithm with scale-free regret bounds, both for constrained and unconstrained domains and restore *almost* entirely the adaptivity to small losses for the unconstrained case. Instead of a  $O(L\sqrt{T})$  additive term, we obtain a  $O(L\sqrt{\tau})$  one, where  $\tau$  is the last step where the loss exceeds all previously observed losses accumulated. While in the worst case this is still  $O(L\sqrt{T})$ , it remains  $O(1)$  as long as losses do not vary more and more widely (e.g., i.i.d. losses). In combination with isotuning, this enhancement is obtained by performing *null updates*, which prevents the algorithm from performing large updates when the scale of the losses suddenly increases. Such skipping of the updates has been used recently to tackle overly long feedback delays for multi-armed bandits (Thune et al., 2019; Zimmert and Seldin, 2020). We also derive a Mirror-Descent algorithm with the same properties as our FTRL variant for both constrained and

unconstrained domains, at least for Bregman divergences that are convex in their second argument—this includes the ‘AdaHedge’ and unconstrained gradient descent cases. This is made possible by generalizing the usage of the *online correction* of Orseau et al. (2017), which ‘centers’ the regret bound and replaces the term  $B_R(\mathbf{x}^*, \mathbf{x}_T)$  with  $B_R(\mathbf{x}^*, \mathbf{x}_1)$ . With the same tools we also derive new (anytime, adaptive) scale-free regret bounds for Prod (Cesa-Bianchi et al., 2007) (Section 4.3), Adapt-ML-Prod (Gaillard et al., 2014) (Section 4.4), BOA (Wintenberger, 2016) (Remark 44 in Appendix D), provide an improved Soft-Bayes algorithm (Orseau et al., 2017) (Appendix F) and a slightly improved bound for AdaHedge (Appendix E). All algorithms retain their original time complexity of  $O(N)$  per round.

To do so, we first develop a general theory (Section 3) and build our set of tools. We start with isotuning (Section 3.1), and demonstrate its use on a few examples, then introduce the online correction (Section 3.4 while using gradient descent as a rolling example, followed by null updates (Section 3.5), before applying these tools to several algorithms. We finish with more discussion about the literature.

## 2 NOTATION AND BACKGROUND

The set of positive integers is  $\mathbb{N}_1$ , and we write  $[T] = \{1, 2, \dots, T\}$ . Define  $(x)_+ = \max\{0, x\}$ . The scalar product of two vectors  $a$  and  $b$  is denoted  $\langle a, b \rangle$ . Define  $\Delta_p(N) = \{\mathbf{x} \in [0, 1]^N : \|\mathbf{x}\|_1 = 1\}$  be the probability simplex of  $N$  coordinates, with  $N \in \mathbb{N}_1$ . Let  $\mathcal{X} \subseteq \mathbb{R}^N$  be a non-empty closed convex set. Let  $\mathbf{x}^* \in \mathcal{X}$  be a fixed point chosen in hindsight. Let  $\mathbf{x}_t \in \mathcal{X}$  for all  $t \in \mathbb{N}_1$  be the sequence of points chosen by the learner. In online convex optimization, the loss function at step  $t$  is  $\ell_t : \mathcal{X} \rightarrow \mathbb{R}$  and the instantaneous regret is  $r_t = \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}^*)$ , while in online linear optimization the loss is a vector  $\ell_t \in \mathbb{R}^N$  and the instantaneous regret is  $r_t = \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle$ . The cumulative regret up to step  $T$  is  $\mathcal{R}_T(\mathbf{x}^*) = \sum_{t \in [T]} r_t$ , which is the quantity of interest to minimize. Let  $\phi : \mathcal{X} \rightarrow \mathbb{R}$  be a convex function, used for analyzing the regret. We abbreviate  $\phi_t \equiv \phi(\mathbf{x}_t)$ . Let  $\|\cdot\|$  be some norm on  $\mathbb{R}^N$ , then:  $\|\cdot\|_*$  is its dual norm;  $D = \sup_{x, y \in \mathcal{X}} \|x - y\|$  is the diameter of  $\mathcal{X}$ ;  $R : \mathcal{X} \rightarrow \mathbb{R}^+$  is a regularizer function that is differentiable and 1-strongly convex with respect to  $\|\cdot\|$  and  $R^*$  is its Fenchel conjugate;  $B_R$  is a Bregman divergence associated with  $R$ , that is,  $\forall(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, B_R(\mathbf{x}, \mathbf{x}') = R(\mathbf{x}) - R(\mathbf{x}') - \langle \mathbf{x} - \mathbf{x}', \nabla R(\mathbf{x}') \rangle$ . The relative entropy (Kullback-Leibler divergence) of  $(a, b) \in \Delta_p(N)^2$  is  $\text{RE}(a\|b) = \sum_{i \in [N]} a_i \ln(a_i/b_i)$ . The indicator function is  $\mathbb{1}_{\text{test}}$  and is equal to 1 if *test* is

true, 0 otherwise. Let  $\eta_t \geq 0$  for all  $t \in \mathbb{N}_1$  be the *learning rate*. Let  $q > 0$ , which will be a parameter of most algorithms. We will also often use (explicitly) the following assumptions:

**Assumption 1** (Generic isotuning assumptions).

$$q > 0, \Delta_0 = 0, \\ \forall t \in \mathbb{N}_1 : \delta_t \geq 0, \Delta_t = \frac{q}{\eta_t} = \sum_{s \in [t]} \delta_s. \quad \diamond$$

These assumptions are used for AdaHedge (de Rooij et al., 2014) for a specific definition of  $\delta_t$  for the Hedge algorithm and with  $q = \ln N$ . In what follows we will provide a more general definition of  $\delta_t$  to accommodate a wide range of situations.

A table of notation is given on page 32.

### 3 GENERAL THEORY

We describe isotuning, then online correction, and finally null updates, and we provide generic regret bounds that we will use in our applications.

#### 3.1 Isotuning

As explained in the introduction, isotuning is based on the idea of sequentially balancing a trade-off between the two terms of the regret. We first introduce a non-sequential form of this idea, as depicted in Fig. 1, which we will build upon for the sequential form, and also re-use as-is when dealing with null updates. We define the iso operator for convenience, then we give a very simple isopoint lemma that we will use multiple times.

**Definition 2** (Iso operator). Let  $\mathcal{S} \subseteq \mathbb{R}$  be a convex set. Define the operator  $\text{iso}_{x \in \mathcal{S}}(f(x), g(x)) = f(s)$  where  $s \in \mathcal{S}$  is such that  $f(s) = g(s)$  for  $f$  continuous and non-decreasing (resp. non-increasing) on  $\mathcal{S}$ , and  $g$  continuous and non-increasing (resp. non-decreasing) on  $\mathcal{S}$ . Then also  $s = \arg \text{iso}_{\mathcal{S}}(f, g)$  with ties broken in favour of small  $s$ . If  $s$  does not exist the result is undefined.  $\diamond$

**Lemma 3** (Isopoint). Let  $\mathcal{S} \subseteq \mathbb{R}$  be convex, and let  $f$  be continuous non-decreasing on  $\mathcal{S}$  and let  $g$  be continuous non-increasing on  $\mathcal{S}$ . If  $\text{iso}_{s \in \mathcal{S}}(f(s), g(s))$  exists then for all  $x \in \mathcal{S}$ ,

$$\min\{f(x), g(x)\} \leq \\ \text{iso}(f(s), g(s)) \leq \max\{f(x), g(x)\}. \quad \diamond$$

*Proof.* Let  $s$  be such that  $f(s) = g(s)$ . Then by the monotonicity of  $f$  and  $g$ , if  $x \leq s$ , then  $f(x) \leq f(s) = g(s) \leq g(x)$ , and conversely for  $x > s$ .  $\square$

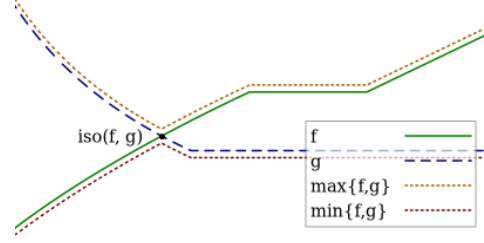


Figure 1: If  $f$  increases and  $g$  decreases, then for all  $x : \min\{f(x), g(x)\} \leq \text{iso}(f, g) \leq \max\{f(x), g(x)\}$ .

**Remark 4.** if  $x^\# = \arg \text{iso}(f, g)$  and  $x^* = \arg \min f + g$  then  $f(x^\#) + g(x^\#) \leq 2 \max\{f(x^*), g(x^*)\}$ .  $\diamond$

Next, we consider a sequential form of the isopoint operator and of the isopoint lemma, for the special case <sup>2</sup> where  $f(x) = x$  for  $x \geq 0$ .

**Definition 5** (Isotuning sequence). A pair  $(X, g)$  is an *isotuning sequence* if the following conditions hold for all  $t \in \mathbb{N}_1$ : (i)  $g_t : (g_t^{\text{low}}, \infty) \rightarrow [0, \infty)$  is continuous and non-increasing, and define  $g_t(g_t^{\text{low}}) = \lim_{\varepsilon \rightarrow 0^+} g_t(g_t^{\text{low}} + \varepsilon)$ , (ii)  $g_t^{\text{low}} \leq X_{t-1} + g_t(g_t^{\text{low}})$ , (iii)  $X_0 = 0$ , (iv)  $X_t = \text{iso}_{y \geq g_t^{\text{low}}}(y, X_{t-1} + g_t(y))$ , that is,  $X_t = X_{t-1} + g_t(X_t)$ .

The associated *domain bound sequence*  $g^{\text{inf}}$  is defined as  $g_T^{\text{inf}} = (\max_{t \in [T]} g_t^{\text{low}})_+$  for all  $T \in \mathbb{N}_1$ , and we set  $g_0^{\text{inf}} = 0$ .  $\diamond$

Condition (ii) ensures that  $X_t$  exists and is unique, as proven in Appendix A.

**Remark 6.**  $\forall T \in \mathbb{N}_1, X_T + \sum_{t=1}^T g_t(X_t) = 2X_T$ .  $\diamond$

We now give our central theorem, which bounds the isotuning sequence within a factor 2 of the optimal tuning *in hindsight*.

**Theorem 7** (Isotuning). Let  $(X, g)$  be an isotuning sequence and define, for all  $T \in \mathbb{N}_0$ ,

$$M_T(x) = x + \sum_{t \in [T]} g_t(x), \quad \text{and} \quad M_T^* = \inf_{x > g_T^{\text{inf}}} M_T(x),$$

then for all  $T \in \mathbb{N}_0$ ,

$$\frac{1}{2} M_T^* \leq \frac{1}{2} M_T(X_T) \leq X_T = \sum_{t \in [T]} g_t(X_t) \leq M_T^*. \quad \diamond$$

*Proof.* For the lower bound, for all  $t \in [T]$ , since  $g_t$  is non-increasing, and since  $g_T^{\text{inf}} \leq X_T$  we have

$$2X_T = X_T + \sum_{t \in [T]} g_t(X_t) \geq X_T + \sum_{t \in [T]} g_t(X_T) \\ = M_T(X_T) \geq M_T^*.$$

<sup>2</sup>The ‘more general’ case of an increasing  $f$  can still be obtained by a change of variables.

For the upper bound, let  $x_T^* = \operatorname{arginf}_{x > g_T^{\inf}} M_T(x)$  and let  $\tau = \min\{t \in [T] : X_t \geq x_T^*\}$ . If  $\tau$  does not exist then  $X_T < x_T^* \leq M_T^*$ , proving the claim. Otherwise, since  $g_t$  is non-increasing then for all  $t \geq \tau$ ,  $g_t(X_t) \leq g_t(x_T^*)$  and so

$$\begin{aligned} X_T &= X_{\tau-1} + \sum_{t=\tau}^T g_t(X_t) \leq x_T^* + \sum_{t=\tau}^T g_t(x_T^*) \\ &\leq M_T(x_T^*) = M_T^*. \quad \square \end{aligned}$$

**Example 8.** Consider the regret bound Eq. (1) of the introduction. Recall from Assumption 1 that  $\Delta_T = q/\eta_T = \sum_t \delta_t$ . Simply define  $\delta_t = \eta_t a_t$ , then:

$$\mathcal{R}_T(\mathbf{x}^*) \leq \Delta_T + \sum_{t \in [T]} \delta_t = 2\Delta_T.$$

Define  $g_t(x) = qa_t/x$  such that  $\delta_t = g_t(\Delta_t)$  for all  $t \in [T]$ , then by Definition 5,  $(\Delta, g)$  is an isotuning sequence (we can extend trivially  $g_t(\cdot) = 0$  for  $t > T$ ), with  $g_t^{\text{low}} = 0$  and thus  $g_T^{\text{inf}} = (\max_t g_t^{\text{low}})_+ = 0$ . Hence, by Theorem 7 where  $X_t \rightsquigarrow \Delta_t$  we obtain

$$\Delta_T \leq M_T^* = \inf_{y > 0} \left\{ y + \sum_{t \in [T]} \frac{qa_t}{y} \right\} = 2\sqrt{q \sum_{t \in [T]} a_t}. \quad \diamond$$

**Example 9.** To demonstrate the adaptivity of isotuning, assume instead that the regret bound is

$$\mathcal{R}_T(\mathbf{x}^*) \leq \frac{q}{\eta_T} + \sum_{t \in [T]} \sqrt{\eta_t} a_t.$$

For example, such a bound can be extracted from Eq. 10 and Prop. 7 by Lattimore and Szepesvári (2020), in the context of partial monitoring. Then we also just set  $\delta_t = \sqrt{\eta_t} a_t = \sqrt{q/\Delta_t} a_t$  and use Assumption 1, so by Theorem 7,

$$\begin{aligned} \mathcal{R}_T(\mathbf{x}^*) &\leq 2\Delta_T \leq 2M_T^* \\ &= 2 \inf_{y > 0} \left\{ y + \sum_{t \in [T]} \sqrt{\frac{q}{y}} a_t \right\} = 2cq^{\frac{1}{3}} \left( \sum_{t \in [T]} a_t \right)^{\frac{2}{3}} \end{aligned}$$

where  $c = 2^{-\frac{2}{3}} + 2^{\frac{1}{3}}$ . Observe that with a change of variables we also have

$$\mathcal{R}_T(\mathbf{x}^*) \leq 2 \inf_{\eta > 0} \left\{ \frac{q}{\eta} + \sum_{t \in [T]} \sqrt{\eta} a_t \right\}$$

which again shows that the regret is within a factor 2 of the optimal learning rate *in hindsight* (for the same sequence of observations).  $\diamond$

## 3.2 Example: AOGD Revisited

Bartlett et al. (2007) propose an adaptive online gradient descent algorithm that automatically and sequentially adapts to the strong convexity of the loss functions, by using a balancing idea similar to that of isotuning to design a specific additional regularizer. For comparison, we recall their bound in Appendix A.2. We show how isotuning can recover their adaptive result while leading to simplified analysis, bound and algorithm, hence just by tuning the learning rate of online gradient descent and without an additional regularizer.

Assume that for all  $t \in [T]$ ,  $\ell_t$  is  $\alpha_t$ -strongly convex, that is, for all  $(x, y) \in \mathcal{X}^2$ ,  $\ell_t(y) \geq \ell_t(x) + \langle y - x, \nabla \ell_t(x) \rangle + \frac{\alpha_t}{2} \|y - x\|_2^2$ . We write  $\nabla_t \equiv \nabla \ell_t(\mathbf{x}_t)$ . Define  $\alpha_{1:t} = \sum_{s \in [t]} \alpha_s$ , and  $r_t = \ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}^*)$ . We use the standard online gradient descent update with projection onto  $\mathcal{X}$ :  $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla_t]$ .

By Lemma 31 (see Appendix A.2), we have, assuming  $1/\eta_0 = 0$  and  $\max_t \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 \leq D^2$  and taking  $q = D^2$ ,

$$2\mathcal{R}_T(\mathbf{x}^*) \leq q \underbrace{\left( \frac{1}{\eta_T} - \alpha_{1:T} \right)}_{(A)} + \underbrace{\sum_{t \in [T]} \eta_t \|\nabla_t\|^2}_{(B)}.$$

The term (A) is decreasing with the learning rate, while the term (B) is increasing with it. Hence, to balance the two terms, we define  $\delta_t = \eta_t \|\nabla_t\|_2^2$  and  $\Delta_T = \sum_{t \in [T]} \delta_t$ , and we also define  $\eta_T$  such that  $\Delta_T = q \left( \frac{1}{\eta_T} - \alpha_{1:T} \right)$ . Thus,

$$\mathcal{R}_T(\mathbf{x}^*) \leq \Delta_T.$$

Stepwise, this corresponds to the update

$$\begin{aligned} \Delta_t &= \Delta_{t-1} + \delta_t \\ q \left( \frac{1}{\eta_t} - \alpha_{1:t} \right) &= q \left( \frac{1}{\eta_{t-1}} - \alpha_{1:t-1} \right) + \eta_t \|\nabla_t\|_2^2 \end{aligned}$$

which is solved for  $\eta_t^{-1} = \psi(\eta_{t-1}^{-1} + \alpha_t, 4\|\nabla_t\|_2^2/q)$  with  $\psi(a, b) = (a + \sqrt{a^2 + b})/2$ ; the learning rate  $\eta_t$  is thus nonnegative non-increasing with  $t$ . To use Theorem 7, we need to express  $\delta_t$  as a function of  $\Delta_t$ :

$$\delta_t = \eta_t \|\nabla_t\|_2^2 = \frac{q \|\nabla_t\|_2^2}{\Delta_t + q\alpha_{1:t}} =: g_t(\Delta_t)$$

and now applying Theorem 7 on the isotuning sequence  $(\Delta, g)$  we obtain

$$\mathcal{R}_T(\mathbf{x}^*) \leq \inf_{y > 0} \left\{ y + \sum_{t \in [T]} \frac{q \|\nabla_t\|_2^2}{y + q\alpha_{1:t}} \right\},$$

Finally, making a change of variables  $z = y/q$  and recalling that  $q = D^2$ ,

$$\mathcal{R}_T(\mathbf{x}^*) \leq \inf_{z>0} \left\{ zD^2 + \sum_{t \in [T]} \frac{\|\nabla_t\|_2^2}{z + \alpha_{1:t}} \right\}.$$

We have thus recovered (and slightly improved) the adaptivity to sequential strong-convexity of the original work (although in a simpler form): if  $\alpha_{1:t} = \Omega(t)$  we can take  $y = 0$  and get a regret bound in  $O(G^2 \log T)$  where  $G = \max_{t \in [T]} \|\nabla_t\|_2$ , but even if  $\alpha_{1:T} = 0$  we also can take  $y = D\sqrt{\sum_{t \in [T]} \|\nabla_t\|_2^2}$  to obtain a regret bounded by  $2y$ . All intermediate regimes can be retrieved as well.

### 3.3 A refined bound

Theorem 7 is fairly general and can accommodate many cases, in particular since formulas of the form  $X \leq \inf_y F(y)$  can easily be simplified by changes of variables, or by picking some good enough value for  $y$ . However, the special case  $g_t(x) = \frac{a_t}{x}$  is frequent enough that it deserves a slightly improved bound.

**Lemma 10.** Let  $(X, g)$  be an isotuning sequence where  $g_t(x) = a_t/x$  with  $a_t \geq 0$  for all  $t \in [T]$  then

$$X_T \leq \sqrt{2 \sum_{t \in [T]} a_t}. \quad \diamond$$

The result follows by Lemma 29 in Appendix A.1, and it can be easily adapted to various other cases where  $g_t(x) \equiv a_t g(x)$  for  $t \geq T_1$  for some  $T_1$ . When applicable, this improved bound saves a factor  $\sqrt{2}$  over Theorem 7.

### 3.4 Online Correction

The standard analysis for online gradient descent (*e.g.*, Hazan (2016)) leads a factor  $\max_{t \in [T]} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \leq D^2$  in the regret bound. However,  $D = \infty$  for unbounded domains. Some algorithms, such as FTRL, manage to have regret bounds that are centered on a fixed point (such as  $\mathbf{x}_1$  or 0) and feature a term  $\|\mathbf{x}_1 - \mathbf{x}^*\|^2$  or  $\|\mathbf{x}^*\|^2$  instead. (Even for bounded convex domains this is an improvement, see Example 42 in Appendix C). With constant learning rates, Mirror Descent is mostly equivalent to FTRL, but when using adaptive learning rates Mirror Descent usually does not provide centered bounds, and features a leading factor of  $\max_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}} B_R(\mathbf{x}_1, \mathbf{x}_2)$  instead. For bounded domains where the Bregman divergence is also bounded, this may be good enough and scale-free guarantees can be obtained (Orabona and Pál, 2018), but this factor can be infinite otherwise.

In this section we use the correction rule of Orseau et al. (2017) to derive general centered regret bounds for a large class of algorithms beyond FTRL. While this correction rule was used for only one algorithm, we show that its usage can be extended to a much larger class, by contrast to other correction rules, such as the exponential correction of Györfi and Ottucsák (2007); Gaillard et al. (2014) (see also Chen et al. (2021); Huang et al. (2021) for other specific corrections). We use Assumption 1.

When deriving centered bounds, an additional difficulty appears: the learning rate becomes ‘off-by-one’ (McMahan, 2017, see also Lemma 12 below), that is, at step  $t$  only the *previous* learning rate  $\eta_{t-1}$  may be used, which means that the update cannot take into account, and adapt to the feedback received at step  $t$ .

At each step, we assume that the point  $\mathbf{x}_t$  is first updated to  $\mathbf{x}_{t'}$  (in a problem-dependent way) using the *previous* learning rate  $\eta_{t-1}$ . Intuitively,

$$\mathbf{x}_{t'} = \text{update}_t(\mathbf{x}_t, \eta_{t-1}). \quad (2)$$

for some algorithm-specific `update` function. Then the following online correction (equivalent to that of Orseau et al. (2017)) is applied:

$$\mathbf{x}_{t+1} = \mathbf{x}_{t'} \frac{\Delta_{t-1}}{\Delta_t} + \frac{\delta_t}{\Delta_t} \mathbf{x}_1. \quad (3)$$

**Lemma 11** (Online correction). Let  $f : \mathcal{X} \rightarrow \mathbb{R}$  be a convex function. Then, under Assumption 1, for all  $t \in [T]$ :

$$\Delta_t f(\mathbf{x}_{t+1}) \leq \Delta_{t-1} f(\mathbf{x}_{t'}) + \delta_t f(\mathbf{x}_1),$$

and holds with equality if  $f$  is affine.  $\diamond$

*Proof.* Follows from  $\Delta_{t-1} + \delta_t = \Delta_t$  and the convexity of  $f$ , then multiplying by  $\Delta_t$  on both sides.  $\square$

Also observe that necessarily  $\mathbf{x}_{t+1} \in \mathcal{X}$  if  $\mathbf{x}_{t'} \in \mathcal{X}$  and  $\mathbf{x}_1 \in \mathcal{X}$ .

Recall that  $\phi : \mathcal{X} \rightarrow \mathbb{R}$  is a convex function used to analyze the regret of the algorithm, and that  $r_t$  is the instantaneous regret. Define for all  $t \in \mathbb{N}_1$ ,

$$\delta_t^* = \left( \frac{1}{\eta_{t-1}} (\phi(\mathbf{x}_{t'}) - \phi(\mathbf{x}_t)) + r_t \right)_+, \quad (4)$$

then we require

$$\delta_t \geq \delta_t^*. \quad (5)$$

In general, this inequality should be as tight as possible.<sup>3</sup> See Algorithm 1. With these conditions we can provide a generic regret bound in this ‘off-by-one’ setting.

<sup>3</sup> $\delta_t$  reduces after cancellations to the *mixability gap* of AdaHedge (de Rooij et al., 2014) in the special case where  $\phi(\mathbf{x}) = \text{RE}(\mathbf{x}^* \|\mathbf{x})$  and  $q = \ln N$ . The main difference is that we explicitly incorporate the whole instantaneous regret  $r_t$  inside  $\delta_t$  so as to obtain a generic regret bound.

---

**Algorithm 1** Generic online convex optimization algorithm sketch in the off-by-one setting. The update of  $\eta_t$  usually has a closed form since  $\delta_t$  depends on  $\eta_{t-1}$  only.

---

```

def generic_OCO_off_by_1( $\mathbf{x}_1 \in \mathcal{X}, q > 0$ ):
     $\Delta_0 = 0$ 
    for  $t = 1, 2, \dots$ :
        predict  $\mathbf{x}_t$ ; observe  $\ell_t$ 
         $\eta_{t-1} = q/\Delta_{t-1}$  # off-by-one learning rate
         $\mathbf{x}_{t'} = \text{update}_t(\mathbf{x}_t, \eta_{t-1})$  # off-by-one update
        set  $\delta_t$  s.t.  $\delta_t \geq \left(\frac{1}{\eta_{t-1}}(\phi(\mathbf{x}_{t'}) - \phi(\mathbf{x}_t)) + r_t\right)_+$ 
            # found by analysis
         $\Delta_t = \Delta_{t-1} + \delta_t$  # isotuning
         $\mathbf{x}_{t+1} = \frac{\Delta_{t-1}}{\Delta_t} \mathbf{x}_{t'} + \frac{\delta_t}{\Delta_t} \mathbf{x}_1$  # online correction

```

---

**Lemma 12** (Generic regret bound, off-by-one setting). Consider Assumption 1. Let  $\phi : \mathcal{X} \rightarrow \mathbb{R}$  be a convex function, and assume that  $\delta_t$  satisfies Eq. (5). The regret of Algorithm 1 is bounded by

$$\mathcal{R}_T(\mathbf{x}^*) \leq \left(1 + \frac{\phi_1}{q} - \frac{\phi_{T+1}}{q}\right) \Delta_T. \quad \diamond$$

*Proof.* From Lemma 11 we have, since  $\phi$  is convex,

$$\begin{aligned} \Delta_t \frac{\phi_{t+1}}{q} &\leq \Delta_{t-1} \frac{\phi_t}{q} + \delta_t \frac{\phi_1}{q} + \Delta_{t-1} \left(\frac{\phi_{t'}}{q} - \frac{\phi_t}{q}\right) \\ \Leftrightarrow r_t &\leq \Delta_{t-1} \frac{\phi_t}{q} - \Delta_t \frac{\phi_{t+1}}{q} + \delta_t \frac{\phi_1}{q} \\ &\quad + \underbrace{\Delta_{t-1} \frac{\phi_{t'} - \phi_t}{q} + r_t}_{\leq \delta_t} \\ &\leq \Delta_{t-1} \frac{\phi_t}{q} - \Delta_t \frac{\phi_{t+1}}{q} + \left(1 + \frac{\phi_1}{q}\right) \delta_t. \end{aligned}$$

Summing over  $t$  and telescoping gives:

$$\sum_{t \in [T]} r_t \leq \Delta_0 \frac{\phi_1}{q} - \Delta_T \frac{\phi_{T+1}}{q} + \left(1 + \frac{\phi_1}{q}\right) \sum_{t \in [T]} \delta_t$$

and the result follows from Assumption 1.  $\square$

Since, in the off-by-one setting,  $\delta_t$  is a function of  $\eta_{t-1}$  and thus of  $\delta_{t-1}$  rather than  $\Delta_t$ , we provide a simple result to deal with this offset.

**Lemma 13** (Isotuning offset). Consider Assumption 1. For all  $t \in \mathbb{N}_1$ , let  $\hat{\delta}_t : (0, \infty) \rightarrow [0, \infty)$  be continuous nonnegative non-increasing functions such that  $\delta_t \leq \hat{\delta}_t(\Delta_t - b_t)$  with  $b_t \geq 0$ . Assume  $(\hat{\Delta}, \hat{\delta})$  is an isotuning sequence. Then, for all  $T \in \mathbb{N}_1$ ,

$$\Delta_T \leq \hat{\Delta}_T + \max_{t \in [T]} b_t. \quad \diamond$$

*Proof.* Assume that  $\Delta_{t-1} \leq \hat{\Delta}_{t-1} + \max_{s < t} b_t$ , which is satisfied for  $t = 1$ . Then

$$\begin{aligned} \Delta_t &\stackrel{(a)}{=} \min\{\Delta_t, \Delta_{t-1} + \hat{\delta}(\Delta_t - b_t)\} \\ &\stackrel{(b)}{\leq} \underset{x \geq (\hat{\delta}_t^{\text{low}} + b_t)_+}{\text{iso}} (x, \Delta_{t-1} + \hat{\delta}(x - b_t)) \\ &\stackrel{(c)}{\leq} \max\{\hat{\Delta}_t + b_t, \hat{\Delta}_{t-1} + \hat{\delta}(\hat{\Delta}_t) + \max_{s < t} b_s\} \\ &= \hat{\Delta}_t + \max_{s \leq t} b_s, \end{aligned}$$

where (a) by assumption, (b) is by Lemma 3, (c) is also by Lemma 3 and taking  $x = \hat{\Delta}_t + b_t$ , and the result holds by induction.  $\square$

**Remark 14.** Such offsets can also be dealt with directly with Theorem 7 by changes of variable.  $\diamond$

**Corollary 15** (Off-by-one). Consider the conditions of Lemma 13, and furthermore  $\delta_t \leq \hat{\delta}_t(\Delta_{t-1})$ . Then

$$\Delta_T \leq \hat{\Delta}_T + \max_{t \in [T]} \{\delta_t\}. \quad \diamond$$

*Proof.* Follows from Lemma 13 where  $b_t = \delta_t$ .  $\square$

Note that Corollary 15 does *not* require that  $\delta_t$  satisfies Eq. (5).

For bounded domains, even with *unknown diameter*, we can already easily obtain scale-free adaptive regret bounds. For example, consider online gradient descent using Algorithm 1 with the (off-by-one) update rule  $\mathbf{x}_{t'} = \mathbf{x}_t - \eta_{t-1} \nabla_t$ . Following Eq. (5) with  $\phi(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^*\|^2/2$  (for some unknown  $\mathbf{x}^* \in \mathcal{X}$ ) we can take  $\delta_t = \eta_{t-1} \|\nabla_t\|^2/2 = q \|\nabla_t\|^2/(2\Delta_{t-1})$ . Define  $\hat{\delta}_t(z) = q \|\nabla_t\|^2/(2z)$  such that  $\delta_t \leq \hat{\delta}_t(\Delta_{t-1})$  for all  $t \in [T]$ , then from Lemma 12, Corollary 15, and Theorem 7 we obtain

$$\mathcal{R}_T(\mathbf{x}^*) \leq \left(1 + \frac{\phi_1}{q}\right) \left(\sqrt{q \sum_{t \in [T]} \|\nabla_t\|^2} + \max_{t \in [T]} \delta_t\right)$$

From the update rule, we also have  $\delta_t = \|\mathbf{x}_{t'} - \mathbf{x}_t\| \|\nabla_t\|/2$ , so for bounded domains  $\delta_t \leq D \max_t \|\nabla_t\|/2$  even if  $D$  is unknown. Using  $\phi_1 = \|\mathbf{x}_1 - \mathbf{x}^*\|^2/2$  we thus have

$$\begin{aligned} \mathcal{R}_T(\mathbf{x}^*) &\leq \\ &\left(1 + \frac{\|\mathbf{x}^* - \mathbf{x}_1\|^2}{2q}\right) \left(\sqrt{q \sum_t \|\ell_t\|^2} + \frac{D}{2} \max_t \|\ell_t\|\right) \end{aligned}$$

which fulfills our requirements of being centered, scale-free and adaptive to small losses. More discussion is given in Example 42. With unknown  $D$ , interesting choices for  $q$  are  $q = 1$ , but also  $q = \|\mathbf{1}\|^2$ ,  $q = \ln T$  or  $q = \|\mathbf{1}\|^2 \ln T$ .

### 3.5 Null Updates

The additive term  $\max_t \delta_t$  of Corollary 15 is unfortunately not well bounded if  $D$  is overly large or infinite (e.g., , for partially constrained domains). For example, with  $\mathcal{X} = \mathbb{R}^N$ , both  $\|\mathbf{x}_{t'} - \mathbf{x}_t\|$  and  $\eta_{t-1} \|\nabla_t\|^2/2$  may be arbitrarily large, because  $\eta_{t-1}$  may be too large compared to  $\|\nabla_t\|_2$ . And enforcing  $\eta_{t-1} \leq 1$  would not lead to scale-free bounds, neither for small loss scales, nor for large loss scales. Note that small loss scales can matter for example when performing gradient descent on an almost flat surface.

At this point, we could use the loss clipping trick of Cutkosky (2019); Mhammedi et al. (2019) with off-by-one isotuning to obtain a bound very similar to the SOLO FTRL bound (hence with the additive  $O(L\sqrt{T})$  term in unbounded domains). Instead, we consider another simple trick that will have additional benefits: updates that are too large, leading to large instantaneous regret, are simply skipped, similarly to skipping updates when delays are too large for bandit algorithms (Thune et al., 2019; Zimmert and Seldin, 2020). This corresponds to performing an update with a zero loss:  $\mathbf{x}_{t'} = \mathbf{x}_t$ , which we call a *null update*, and, by contrast to using the doubling trick (Cesa-Bianchi et al., 1997) (see also Mhammedi et al. (2019)), the algorithm does not need to be additionally restarted. The instantaneous regret on null updates are dealt with separately (outside of  $\Delta_T$ ), as demonstrated in the following generic result. We call  $\mathcal{T}_\emptyset$  the set of steps where null updates are performed.

**Theorem 16** (Generic regret bound, null updates). Consider the assumptions of Lemma 12, but modify Algorithm 1 as follows: When  $t \in \mathcal{T}_\emptyset$  set  $\mathbf{x}_{t'} = \mathbf{x}_t$ , and set  $\delta_t \geq 0$ . Then the regret of the algorithm is at most

$$\mathcal{R}_T(\mathbf{x}^*) \leq \left(1 + \frac{\phi_1}{q} - \frac{\phi_{T+1}}{q}\right) \Delta_T + \sum_{t \in \mathcal{T}_\emptyset} (r_t - \delta_t). \quad \diamond$$

*Proof.* Similar to the proof of Lemma 12, except that for  $t \in \mathcal{T}_\emptyset$ ,  $\phi_{t'} = \phi_t$ .  $\square$

Note that when  $t \in \mathcal{T}_\emptyset$ , it is important to set  $\delta_t > 0$  to increase  $\Delta_t$  (and thus decrease  $\eta_t$ ) and avoid triggering repeated null updates.

Consider the function  $\hat{\delta}_t(\cdot)$  used in Corollary 15. We perform a null update at step  $t$  when the instantaneous regret term coming from the update is larger than all previous ones added up, that is, when  $\hat{\delta}_t(\Delta_{t-1}) \geq c\Delta_{t-1}$  for some  $c > 0$ . Define  $\mathcal{T}_\emptyset = \{t \in [T] : \hat{\delta}_t(\Delta_{t-1}) > c\Delta_{t-1}\}$  to be the set of steps where a null update is performed.

For  $t \notin \mathcal{T}_\emptyset$ , since  $\delta_t \leq \hat{\delta}_t(\Delta_{t-1}) \leq c\Delta_{t-1}$ , using

Lemma 3,

$$\begin{aligned} \delta_t &\leq \hat{\delta}_t(\Delta_{t-1}) = \min\{\hat{\delta}_t(\Delta_{t-1}), c\Delta_{t-1}\} \\ &\leq \text{iso}_{y>0}(\hat{\delta}_t(y), cy). \end{aligned} \quad (6)$$

This iso value has the particularity of being independent of  $\Delta_{t-1}$  and thus of the learning rate  $\eta_{t-1}$ . For  $\tau = \max \mathcal{T}_\emptyset$ , the inequality is reversed, hence,

$$c\Delta_{\tau-1} \leq \text{iso}_{y>0}(\hat{\delta}_\tau(y), cy). \quad (7)$$

Now, what value for  $\delta_t$  should we choose when  $t \in \mathcal{T}_\emptyset$  is a null update step? A good choice is again  $\delta_t = \text{iso}_{y>0}(\hat{\delta}_\tau(y), cy)$ , which ensures that for  $t \in \mathcal{T}_\emptyset$  :  $c\Delta_{t-1} \leq \delta_t \leq \hat{\delta}_t(\Delta_{t-1})$ . This ensures by Lemma 3 that for  $t \notin \mathcal{T}_\emptyset$  :  $\delta_t \leq \hat{\delta}_t(\Delta_{t-1}) \leq c\Delta_{t-1}$ . We now provide a generic theorem that uses this particular value of  $\delta_t$ .

**Theorem 17** (Isotuning with null updates). Under the same conditions as Corollary 15, and additionally let  $c > 0$  and for  $t \in \mathcal{T}_\emptyset$  set  $\delta_t = \text{iso}_{y>0}(\hat{\delta}_t(y), cy)$ . Then for all  $T \in \mathbb{N}_1$  and with  $\tau = \max \mathcal{T}_\emptyset$ ,

$$\Delta_T \leq \hat{\Delta}_T + \max_{t \in [T]} \text{iso}_{y>0}(\hat{\delta}_t(y), cy), \quad (i)$$

$$\sum_{t \in \mathcal{T}_\emptyset} \delta_t \leq \left(\frac{1}{c} + 1\right) \text{iso}_{y>0}(\hat{\delta}_\tau(y), cy). \quad (ii) \quad \diamond$$

*Proof.* (i) For  $t \in \mathcal{T}_\emptyset$ ,  $\delta_t = \text{iso}_{y>0}(\hat{\delta}_t(y), cy) \leq \max\{\hat{\delta}_t(\Delta_{t-1}), c\Delta_{t-1}\} = \hat{\delta}_t(\Delta_{t-1})$  by Lemma 3, hence for all  $t \in [T]$ ,  $\delta_t \leq \hat{\delta}_t(\Delta_{t-1})$ , and we can then use Corollary 15. Moreover, for  $t \notin \mathcal{T}_\emptyset$ ,  $\delta_t \leq \hat{\delta}_t(\Delta_{t-1}) = \min\{\hat{\delta}_t(\Delta_{t-1}), c\Delta_{t-1}\} \leq \text{iso}_{y>0}(\hat{\delta}_t(y), cy)$ , and thus  $\max_t \delta_t \leq \max_t \text{iso}_{y>0}(\hat{\delta}_t(y), cy)$ .

(ii) The result follows from  $\sum_{t \in \mathcal{T}_\emptyset} \delta_t \leq \Delta_\tau = \Delta_{\tau-1} + \delta_\tau$  and Eqs. (6) and (7).  $\square$

**Remark 18.** Theorem 17 can be straightforwardly generalized from  $cy$  to any sequence of continuous non-decreasing functions  $\{f_t(y)\}_t$ , including constant functions.  $\diamond$

Continuing our online gradient descent example, recall that  $\hat{\delta}_t(y) = q\|\nabla_t\|^2/(2y)$ , and thus  $\text{iso}_{y>0}(q\|\nabla_t\|^2/(2y), y) = \sqrt{q/2}\|\nabla_t\|$ . So if we perform null updates with  $c = 1$ , that is, when  $\eta_{t-1}\|\nabla_t\|^2/2 > \Delta_{t-1}$  (or, equivalently, when  $\sqrt{q/2}\|\nabla_t\| > \Delta_{t-1}$ ), then by Theorem 17 (and Lemma 10) we have

$$\Delta_T \leq \sqrt{q \sum_{t \in [T]} \|\nabla_t\|^2} + \sqrt{q/2} \max_t \|\nabla_t\|.$$

Thus, now the additive term is nicely bounded and does not depend on the diameter  $D$ . It remains to

bound the regret incurred on null update rounds as per Theorem 16:

$$\begin{aligned} \sum_{t \in \mathcal{T}_0} r_t &\leq \sum_{t \in \mathcal{T}_0} \|\mathbf{x}^* - \mathbf{x}_t\| \|\nabla_t\| \\ &= \sum_{t \in \mathcal{T}_0} \|\mathbf{x}^* - \mathbf{x}_t\| \sqrt{2/q} \delta_t \\ &\leq 2 \max_{t \in \mathcal{T}_0} \|\mathbf{x}^* - \mathbf{x}_t\| \|\nabla_\tau\| \end{aligned}$$

where we used Theorem 17 (ii) on the last line. Finally,  $\max_{t \in \mathcal{T}_0} \|\mathbf{x}^* - \mathbf{x}_t\|$  is bounded by  $D$ , but also by  $\|\mathbf{x}^* - \mathbf{x}_1\| + \sqrt{2q\tau}$  by Lemma 41 (in Appendix C), where  $\tau = \max \mathcal{T}_0$ .

Putting it all together we have

$$\begin{aligned} \mathcal{R}_T(\mathbf{x}^*) &\leq \\ &\left(1 + \frac{\|\mathbf{x}_1 - \mathbf{x}^*\|^2}{2q}\right) \left(\sqrt{2q \sum_{t \in [T]} \|\nabla_t\|^2} + \sqrt{\frac{q}{2}} \max_t \|\nabla_t\|\right) \\ &\quad + 2 \min\{D, \|\mathbf{x}_1 - \mathbf{x}^*\| + \sqrt{2q\tau}\} \|\nabla_\tau\|. \end{aligned}$$

Now the main difference with the SOLO-FTRL bound (Orabona and Pál, 2018) is that we have the term  $\sqrt{\tau} \|\nabla_\tau\|$  instead of  $\sqrt{T} \max_{t \in [T]} \|\nabla_t\|$ . Naturally, we wish to understand how much of an improvement this is. First of all, since  $\tau \leq T$ , our bound is never worse. Furthermore, since  $\delta_\tau \geq \Delta_{\tau-1} \geq \sum_{t \in \mathcal{T}_0 \setminus \{\tau\}} \delta_t$  we have  $\|\nabla_\tau\| \geq \sum_{t \in \mathcal{T}_0 \setminus \{\tau\}} \|\nabla_t\|$ . This means that a null update is triggered only when the loss is larger than the cumulative losses on all previous null updates. We can draw a few observations: (i) unless losses grow exponentially forever (which makes learning rather difficult),  $\tau$  has a finite value even as  $T$  grows; (ii) if losses are i.i.d. from a bounded distribution, large losses are observed exponentially fast and thus  $\tau$  is necessarily small; (iii) when performing gradient descent on a convex surface, the largest losses are usually observed during the first few steps and then  $\tau = O(1)$ .

The most likely ‘bad’ scenario is when very small gradients  $\ell$  are observed for  $T - 1$  steps and on the  $T$ -th step a large loss  $L$  is incurred (possibly  $\mathbf{x}_1$  starts on a plateau and the loss function is only quasiconvex). Moreover, we will prove in Appendix B that  $\|\nabla_\tau\|^2 \geq \sum_{t < \tau} \|\nabla_t\|^2$ . This means that in this scenario, to trigger a null update, we must have  $L \geq \sqrt{T} \ell$ , even if only one null update has been triggered before. Hence, as long as losses do not grow faster than a factor  $\sqrt{T}$ , no null update is triggered.

Hence, for most situations, the adaptivity to small quadratic losses  $\sqrt{\sum_t \|\nabla_t\|^2}$  is restored.

## 4 APPLICATIONS

We apply our tools to several algorithms: scale-free FTRL, scale-free Mirror Descent, and scale-free Prod, scale-free Adapt-ML-Prod, and also a slight generalization of AdaHedge (Appendix E) and a refinement of Soft-Bayes (Appendix F). We now consider linear losses  $\ell_t \in \mathbb{R}^N$ . For the first two applications, the instantaneous regret is  $r_t = \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle$ .

### 4.1 Scale-Free FTRL

In this section we show how the results we obtain for online gradient descent in Section 3.5 can be generalized to FTRL with any regularizer that is 1-strongly convex to a given norm  $\|\cdot\|$ . As discussed in Section 3.5, we use null updates to restore the adaptivity to small losses for unbounded domains, that was lost for the SOLO FTRL regret bound whose regret is dominated by the  $O(L\sqrt{T})$  term, where  $L = \max_{t \in [T]} \|\ell_t\|_*$ .

We first consider FTRL with isotuning but without null updates (similar to AdaFTRL (Orabona and Pál, 2018, appendix B.1)), then we construct a meta-algorithm that filters out the exceedingly large losses before sending the losses to the subalgorithm. The resulting (compound) algorithm is called isoFTRL (see Algorithm 2).

---

#### Algorithm 2 FTRL with null updates and isotuning

---

**def** isoFTRL( $\mathcal{X}, R, q > 0$ ):

$L_0 = 0$ ;  $\Delta_0 = 0$

**for**  $t = 1, 2, 3, \dots$ :

$\eta = q/\Delta_{t-1}$  #  $\eta_{t-1}$

$\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \left\{ \langle \tilde{L}_{t-1}, \mathbf{x} \rangle + \frac{R(\mathbf{x})}{\eta} \right\}$

**predict**  $\mathbf{x}_t$ ; **observe**  $\ell_t$

**if**  $\frac{1}{2}\eta \|\ell_t\|_*^2 \leq \Delta_{t-1}$ : # **or**  $\sqrt{q/2} \|\ell_t\|_* \leq \Delta_{t-1}$

$\delta_t = \frac{1}{2}\eta \|\ell_t\|_*^2$  # **from Eq. (5)**

$\tilde{\ell}_t = \ell_t$

**else**:

$\delta_t = \sqrt{\frac{1}{2}q} \|\ell_t\|_*$  # **isopoint**

$\tilde{\ell}_t = 0$  # **null update**

$\tilde{L}_t = \tilde{L}_{t-1} + \tilde{\ell}_t$

$\Delta_t = \Delta_{t-1} + \delta_t$  # **isotuning**

---

First we recall (and slightly generalize) an intermediate bound for AdaFTRL.

**Lemma 19** (AdaFTRL, Orabona and Pál (2018), Lemma 5). Consider Assumption 1, and  $\delta_t \geq B_{R^*}(-\eta_{t-1}L_t, -\eta_{t-1}L_{t-1})/\eta_{t-1}$ . Then the regret of



FTRL with isotuning to any  $\mathbf{x}^* \in \mathcal{X}$  is bounded by

$$\mathcal{R}_T(\mathbf{x}^*) \leq \left(1 + \frac{R(\mathbf{x}^*)}{q}\right) \Delta_T. \quad \diamond$$

We now describe isotuning FTRL with null updates, see Algorithm 2. Recall that the regularizer  $R$  is 1-strongly convex with respect to the norm  $\|\cdot\|$ . Define  $\hat{\delta}_t(y) = \frac{q}{2y} \|\ell_t\|_*^2$ . From Section 3.5 we take  $c = 1$ , and recall that  $\mathcal{T}_\emptyset = \{t \in [T] : \hat{\delta}_t(\Delta_{t-1}) > \Delta_{t-1}\}$  is the set of steps where a null update is performed, then we run FTRL with modified losses  $\ell_t = \ell_t \mathbb{1}[t \notin \mathcal{T}_\emptyset]$ . Furthermore, define

$$\delta_t = \begin{cases} \text{iso}_{y>0} \left( q \frac{\|\ell_t\|_*^2}{2y}, y \right) = \sqrt{\frac{1}{2}q} \|\ell_t\|_* & \text{when } t \in \mathcal{T}_\emptyset, \\ \hat{\delta}_t(\Delta_{t-1}) & \text{otherwise.} \end{cases}$$

Then  $\delta_t \geq B_{R^*}(-\eta_{t-1}\tilde{L}_t, -\eta_{t-1}\tilde{L}_{t-1})/\eta_{t-1}$  by proposition 2 of Orabona and Pál (2018).

Then we can show the following bound. The proof is in Appendix B.

**Theorem 20.** The regret of isoFTRL (Algorithm 2) is bounded by

$$\begin{aligned} \mathcal{R}_T(\mathbf{x}^*) \leq & \left( \sqrt{q} + \frac{R(\mathbf{x}^*)}{\sqrt{q}} \right) \left( \sqrt{\sum_{t \in [T]} \|\ell_t\|_*^2} + \frac{1}{\sqrt{2}} \max_t \|\ell_t\|_* \right) \\ & + 2 \min \left\{ D, \sqrt{2R(\mathbf{x}^*)} + 2\sqrt{2q\tau} \right\} \|\ell_\tau\|_*, \end{aligned}$$

where  $\tau = \max \mathcal{T}_\emptyset$  is the last step a null update is performed, and is such that  $\|\ell_\tau\|_*^2 \geq \sum_{t < \tau} \|\ell_t\|_*^2$ .  $\diamond$

## 4.2 Scale-Free Mirror Descent

Orabona and Pál (2018) provide two lower bounds showing that their scale-free Mirror Descent (meant for bounded maximum Bregman divergence between any two points) fails with (super)linear regret: the first one is for the quadratic regularizer (gradient descent), and the second one is for the entropic regularizer for which the Bregman divergence is the relative entropy, which is not bounded even though the regularizer is bounded.

We now propose a scale-free variant of the Mirror Descent algorithm with null updates, isotuning and online correction, for bounded and unbounded domains, at least for Bregman divergences that are convex in their second argument, including the quadratic and the entropic regularizers. The two lower bound examples mentioned above do not apply to our algorithm, thanks to using the online correction to center the bound, leading to a factor  $B_R(\mathbf{x}^*, \mathbf{x}_1)$  instead of  $\sup_{(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2} B_R(\mathbf{x}, \mathbf{x}')$ . Choosing  $\mathbf{x}_1 =$

$\text{argmin}_{\mathbf{x} \in \mathcal{X}} R(\mathbf{x})$  ensures that  $\phi_1 = B_R(\mathbf{x}^*, \mathbf{x}_1) \leq R(\mathbf{x}^*) - R(\mathbf{x}_1)$  similarly to FTRL, but in general one should take  $\mathbf{x}_1 = \text{argmin}_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{x}' \in \mathcal{X}} B_R(\mathbf{x}^*, \mathbf{x})$ . In particular, when  $R(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|_2^2$ , then by Jung's theorem we have  $\|\mathbf{x}^* - \mathbf{x}_1\|_2 \leq D/\sqrt{2}$  and thus we can take  $q = (D/\sqrt{2})^2/2 = D^2/4$ . See additional discussion in Example 42.

The algorithm is described in Algorithm 3. For null updates, we take  $c = 1$  also.

---

**Algorithm 3** Scale-free Mirror Descent for bounded and unbounded decision sets.

---

```

def isoMD( $\mathcal{X}$ ,  $R$ ,  $q > 0$ ):
     $\Delta_0 = 0$  ;  $\mathbf{x}_1 = \text{argmin}_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{x}' \in \mathcal{X}} B_R(\mathbf{x}', \mathbf{x})$ 
    for  $t = 1, 2, \dots$ :
        predict  $\mathbf{x}_t$ ; observe  $\ell_t$ 
         $\eta = q/\Delta_{t-1}$ 
        if  $\frac{1}{2}\eta \|\ell_t\|_*^2 \leq \Delta_{t-1}$ : # or  $\sqrt{q/2} \|\ell_t\|_* \leq \Delta_{t-1}$ 
             $\mathbf{x}_{t'} = \text{argmin}_{\mathbf{x} \in \mathcal{X}} \left\{ \langle \mathbf{x}, \ell_t \rangle + \frac{B_R(\mathbf{x}, \mathbf{x}_t)}{\eta} \right\}$ 
             $\delta_t = \frac{1}{2}\eta \|\ell_t\|_*^2$  # from Eq. (5)
        else:
             $\mathbf{x}_{t'} = \mathbf{x}_t$  # null update
             $\delta_t = \sqrt{\frac{1}{2}q} \|\ell_t\|_*$  # isopoint

         $\Delta_t = \Delta_{t-1} + \delta_t$  # isotuning
         $\mathbf{x}_{t+1} = \frac{\Delta_{t-1}}{\Delta_t} \mathbf{x}_{t'} + \frac{\delta_t}{\Delta_t} \mathbf{x}_1$  # online cor.
```

---

In this section we take  $\phi_t = B_R(\mathbf{x}^*, \mathbf{x}_t)$ , and recall that  $\phi(\cdot)$  must be convex.<sup>4</sup>

**Theorem 21.** With  $\phi_1 = B_R(\mathbf{x}^*, \mathbf{x}_1)$ , the regret of isoMD (Algorithm 3) is bounded by

$$\begin{aligned} \mathcal{R}_T(\mathbf{x}^*) \leq & \left( \sqrt{q} + \frac{\phi_1}{\sqrt{q}} \right) \left( \sqrt{\sum_{t \notin \mathcal{T}_\emptyset} \|\ell_t\|_*^2} + \frac{1}{\sqrt{2}} \max_{t \in [T]} \|\ell_t\|_* \right) \\ & + 2 \min \{ D, \|\mathbf{x}^* - \mathbf{x}_1\| + \sqrt{2q\tau} \} \|\ell_\tau\|_*, \end{aligned}$$

where  $\tau = \max \mathcal{T}_\emptyset$  is the last step a null update is performed, and is such that  $\|\ell_\tau\|_*^2 \geq \sum_{t < \tau} \|\ell_t\|_*^2$ .  $\diamond$

The proof is given in Appendix C.

## 4.3 Scale-free adaptive Prod

The Prod family of algorithms (Cesa-Bianchi et al., 2007; Gaillard et al., 2014; Sani et al., 2014; Orseau et al., 2017) are based on Mirror-Descent style multiplicative updates, but their analysis does not fit

---

<sup>4</sup>This is satisfied at least by  $B_R(a, b) = \|a - b\|_2^2/2$  and  $B_R(a, b) = \sum_i a_i \ln(a_i/b_i)$ .

either the MD nor the FTRL analysis framework. Previous works have resorted to more or less ad-hoc analysis. They can be analysed, however, with our generic off-by-one setting and online correction, and only the quantity of interest  $\delta_t$  needs to be bounded as per Eq. (5) on an algorithm-dependent basis, taking  $\phi(\mathbf{x}) = \text{RE}(\mathbf{x}^*|\mathbf{x}) = \sum_{i \in [N]} \mathbf{x}_i^* \ln(\mathbf{x}_i^*/\mathbf{x}_i)$ . Also see Appendix F for a slightly simplified sparse Soft-Bayes algorithm (Orseau et al., 2017) and enhanced regret bound.

In this section, we demonstrate how to obtain a scale free and adaptive version of the Prod algorithm (Cesa-Bianchi et al., 2007; Gaillard et al., 2014) with a single learning rate. This algorithm was originally analyzed with constant learning rates and a doubling trick with restarts (Cesa-Bianchi et al., 2007). While this single learning-rate algorithm is not so useful in itself in practice as it can be advantageously replaced with AdaHedge (de Rooij et al., 2014), the example is technically useful as it shows how some constraints of the update rule can be easily tackled with null updates, while demonstrating how the tools we built in the previous sections lead to a straightforward analysis of the regret. It also serves as a prelude to the next section where we analyse the multiple learning rate version. It also uses Lemma 46 which tightens the leading constant factor by  $\sqrt{2}$  on lower bounding  $\log(1+x)$  compared to Lemma 1 of Cesa-Bianchi et al. (2007).

We use Assumption 1, and Algorithm 1 with null updates—but without a doubling trick or any restarting. The decision set  $\mathcal{X}$  is the probability simplex of  $N$  coordinates, and we take  $\phi(\mathbf{x}) = \text{RE}(\mathbf{x}^*|\mathbf{x})$  and  $q = \phi_1 = \ln N$ .

For all  $t \in [T]$ , let  $\ell_t \in \mathbb{R}^N$ , and define  $\bar{\ell}_t = \sum_{i \in [N]} \mathbf{x}_{i,t} \ell_{i,t}$ .

Define the prod update (off by one):

$$\mathbf{x}_1 = \left( \frac{1}{N}, \dots, \frac{1}{N} \right)$$

$$\forall i \in [N] : \mathbf{x}_{i,t'} = \mathbf{x}_{i,t} (1 + \eta_{i,t-1} (\bar{\ell}_t - \ell_{i,t})) \quad (8)$$

For all  $t$ , define  $s_t = \max_i |\bar{\ell}_t - \ell_{i,t}|$  and  $\hat{\delta}_t(x) = \frac{qs_t^2}{2x}$ , with  $\hat{\delta}_t(x) = \infty$  for  $x \leq 0$ .

As per Eq. (5), for  $t \notin \mathcal{T}_\emptyset$ , we can define

$$\delta_t = \max_i \left( \frac{-1}{\eta_{t-1}} \ln(1 + \eta_{t-1} (\bar{\ell}_t - \ell_{i,t})) + (\bar{\ell}_t - \ell_{i,t}) \right),$$

and by Lemma 46, we have that, when  $\Delta_{t-1} - qs_t > 0$ ,

$$\delta_t \leq \hat{\delta}_t(\Delta_{t-1} - qs_t) = \hat{\delta}_t(\Delta_t - \delta_t - qs_t). \quad (9)$$

The update in Eq. (8) may be invalid (non-positive) when  $1 - \eta_{t-1}s_t \leq 0$ , which corresponds to  $\Delta_{t-1} - qs_t \leq 0$ , in which case  $\hat{\delta}_t(\Delta_{t-1} - qs_t) = \infty$ . To avoid this

we perform a null update when  $\hat{\delta}_t(\Delta_{t-1} - qs_t) > \Delta_{t-1} - qs_t$ , in which case ( $t \in \mathcal{T}_\emptyset$ ) we set  $\delta_t = \text{iso}_{y>0}(\hat{\delta}_t(y), y) = \sqrt{\frac{q}{2}}s_t$ , which ensures by Lemma 3 that

$$\begin{aligned} \forall t \notin \mathcal{T}_\emptyset : \delta_t &\leq \hat{\delta}_t(\Delta_{t-1} - qs_t) \leq \sqrt{\frac{q}{2}}s_t \leq \Delta_{t-1} - qs_t, \\ \forall t \in \mathcal{T}_\emptyset : \Delta_{t-1} - qs_t &\leq \delta_t = \sqrt{\frac{q}{2}}s_t \leq \hat{\delta}_t(\Delta_{t-1} - qs_t). \end{aligned}$$

We call isoProd the resulting algorithm.

**Theorem 22.** Let  $S = \max_{t \in [T]} s_t$ . The regret of isoProd is bounded by

$$\mathcal{R}_T(\mathbf{x}^*) \leq 2 \sqrt{\ln(N) \sum_t s_t^2} + S(2 \ln N + \sqrt{2 \ln N} + 2). \quad \diamond$$

*Proof.* By Theorem 16 we have  $\mathcal{R}_T(\mathbf{x}^*) \leq 2\Delta_T + \sum_{t \in \mathcal{T}_\emptyset} (r_t - \delta_t)$ . Since  $r_t \leq s_t = \sqrt{2/q}\delta_t$  for  $t \in \mathcal{T}_\emptyset$ ,

$$\begin{aligned} \sum_{t \in \mathcal{T}_\emptyset} (r_t - \delta_t) &\leq (\sqrt{2/q} - 1) \sum_{t \in \mathcal{T}_\emptyset} \delta_t \\ &\leq (\sqrt{2/q} - 1)(\Delta_{\tau-1} + \delta_\tau) \\ &\leq (\sqrt{2/q} - 1)(\sqrt{q/2}S_\tau + qs_\tau + \sqrt{q/2}S_\tau) \\ &\leq 2S, \end{aligned}$$

where  $\tau = \max \mathcal{T}_\emptyset$ , and we used  $\Delta_{t-1} - qs_t \leq \delta_t$  for  $t \in \mathcal{T}_\emptyset$ . Let  $(\hat{\Delta}, \hat{\delta})$  be an isotuning sequence, then by Lemma 13 and Eq. (9) we know that  $\Delta_T \leq \hat{\Delta}_T + \max_t \{\delta_t + qs_t\} \leq \hat{\Delta}_T + \sqrt{q/2}S + qS$ . Finally by Lemma 10 we have  $\hat{\Delta}_T \leq \sqrt{q \sum_t s_t^2}$ .  $\square$

**Remark 23.** Alternatively, we could use the update

$$\mathbf{x}_{i,t'} = \mathbf{x}_{i,t} \frac{1 - \eta_{t-1} \ell_{i,t}}{1 - \eta_{t-1} \bar{\ell}_t}$$

which leads to  $\delta_t = \max_i \eta_{t-1} \frac{\ell_{i,t}^2/2}{1 - \eta_{t-1} \bar{\ell}_t}$  and thus  $\mathcal{R}_T(\mathbf{x}^*) = O(\sqrt{q \sum_t \max_i \ell_{i,t}^2})$  and then use Lemma 48 to sequentially translate the losses arbitrarily.  $\diamond$

#### 4.4 Scale-free Adapt-ML-Prod

Gaillard et al. (2014) propose the Adapt-ML-Prod algorithm, for losses in  $[0, 1]$  with multiple learning rates, to obtain a regret bound which depends only on the excess losses  $\sqrt{\sum_t (\bar{\ell}_t - \ell_{i,t})^2}$  to the best expert  $i$  in hindsight, where  $\bar{\ell}_t$  is the loss incurred by the algorithm and  $\ell_{i,t}$  is the loss incurred by the expert  $i$  at step  $t$  and show interesting consequences of this bound, in particular it is bounded by a constant in expectation when losses are i.i.d. By contrast, AdaHedge (de Rooij et al., 2014; van Erven et al., 2011) can tackle losses

with unknown range, but has a regret bound featuring the quantity  $\sqrt{\sum_t \sum_{i \in [N]} \mathbf{x}_{i,t} (\bar{\ell}_t - \ell_{i,t})^2}$  instead. We extend Adapt-ML-Prod to tackle unbounded and unknown loss ranges using isotuning, online correction, and null updates—or equivalently we extend isoProd of the previous section to using one learning rate per expert.<sup>5</sup>

Gaillard et al. consider losses in  $[0, 1]$ , and their algorithm predicts with positive weights  $\propto \mathbf{x}_{i,t} \eta_{i,t-1}$  and suffers the loss

$$\bar{\ell}_t = \sum_{i \in [N]} \frac{\mathbf{x}_{i,t} \eta_{i,t-1}}{\sum_{j \in [N]} \mathbf{x}_{j,t} \eta_{j,t-1}} \ell_{i,t}. \quad (10)$$

For all  $T \in \mathbb{N}_1, \forall i \in [N]$ , define  $r_{i,t} = \bar{\ell}_t - \ell_{i,t}$  and  $\mathcal{R}_{i,T} = \sum_{t \in [T]} r_{i,t}$ , also  $V_{i,T} = \sum_{t \in [T]} r_{i,t}^2$ , and  $s_t = \max_{i \in [N]} |r_{i,t}|$ . Assuming  $N \geq 2$  and  $T \geq 2$ , we can rewrite the regret bound of Adapt-ML-Prod (Gaillard et al., 2014, Corollary 4) to the best expert  $i$  in hindsight as:

$$\mathcal{R}_{i,T} \leq \left[ \sqrt{(1 + V_{i,T}) \ln N} + 3 \ln N \right] \left[ 4 + \frac{1 + \ln \ln T}{\ln N} \right].$$

Note, for  $T = 10^{40}$  and  $N \geq 4$ ,  $(1 + \ln \ln T) / \ln N < 4$ .

Adapt-ML-Prod is meant for losses in  $[0, 1]$  and can be used for losses in any  $[a, b]$  if an upper bound on  $b - a$  is known. However, when the loss range is not known beforehand, as for isoProd we perform a null update when the update may lead to invalid (negative) weights. We obtain an algorithm with a regret guarantee that is just as good as when the loss range is known beforehand.

As for Adapt-ML-Prod, our algorithm uses one learning rate per coordinate, and the tools of the previous sections will apply almost always coordinate-wise. For all  $i \in [N]$ , we first define the update  $\mathbf{x}_{i,t'}$  for Adapt-ML-Prod:

$$\begin{aligned} \mathbf{x}_{i,1} &= 1, \\ \mathbf{x}_{i,t'} &= \mathbf{x}_{i,t} (1 + \eta_{i,t-1} (\bar{\ell}_t - \ell_{i,t})), \end{aligned} \quad (11)$$

and then apply the online correction of Eq. (3). As for Prod, the problem with Eq. (11) is that it is invalid when  $\mathbf{x}_{i,t'}$  becomes negative. We prevent this issue by performing null updates ( $t \in \mathcal{T}_\emptyset$ ) whenever  $\eta_{i,t-1} |\bar{\ell}_t - \ell_{i,t}| \geq 1/2$  for some  $i \in [N]$ , that is,

<sup>5</sup> Wintenberger (2016) proposes an adaptive algorithm based on exponential weights that is claimed to adapt to the maximum excess losses to the best expert in hindsight (see p.4, top, def. of  $l_{j,t}$ , and p.13, Theorem 3.3). Unfortunately, the main results rely on the following wrong claim to hold (proof of Theorem 1.1): “as  $X$  is centered, we can bound  $[\dots] 1 - \mathbb{E}[X \mathbb{1}_{X < -1/2}] \leq \mathbb{E}[1 + X \mathbb{1}_{X < -1/2}]$ ”. It is not clear whether this can be fixed simply, in particular because the preconditions of Lemma 4 in Cesa-Bianchi et al. (2007) need to be satisfied. See Remark 44 in Appendix D instead.

$q |\bar{\ell}_t - \ell_{i,t}| \geq \Delta_{i,t-1}/2$ . Note that to ensure that  $\sum_i \mathbf{x}_{i,t'} = \sum_i \mathbf{x}_{i,t}$  for all rounds  $t$  (as will be shown in the proof of Theorem 24), we perform a null update *on all coordinates at the same time*, which leads to an additional difficulty compared to isoProd, as  $\hat{\delta}_{i,t}(\Delta_{i,t}) > c \Delta_{i,t-1}$  is not always true when  $t \in \mathcal{T}_\emptyset$ .

We use Assumption 1 and we take  $\phi_i(\mathbf{x}_{i,t}) = -\ln \mathbf{x}_{i,t}$  for all  $i \in [N]$ , and  $\mathcal{X} = [0, \infty)^N$ . Set  $q = \ln N$ . and, recalling that  $r_{i,t} = \bar{\ell}_t - \ell_{i,t}$  and  $s_t = \max_i |r_{i,t}|$ , define from Eq. (5)

$$t \notin \mathcal{T}_\emptyset : \delta_{i,t} = \delta_{i,t}^* = r_{i,t} - \frac{1}{\eta_{i,t-1}} \ln(1 + \eta_{i,t-1} r_{i,t}),$$

$$t \in \mathcal{T}_\emptyset : \delta_{i,t} = s_t.$$

On a null update we set  $\delta_{i,t} = s_t$ , so that  $\delta_{i,t} \geq r_{i,t}$  conveniently absorbs the instantaneous regret; note that we do not set  $\delta_{i,t} = |\bar{\ell}_t - \ell_{i,t}|$  to avoid triggering null updates of the same scale  $N$  times, which would lead to an additive term  $O(N \max_t |r_{i,t}|)$  in the bound instead of  $O(\max_{j,t} |r_{j,t}|)$ .

Observe that  $\phi_{i,1} = 0$ , so from Theorem 16, since also  $\delta_{i,t} \geq r_{i,t}$  for  $t \in \mathcal{T}_\emptyset$ , we can simplify:

$$\mathcal{R}_{i,T} \leq \left( 1 + \frac{\ln \mathbf{x}_{i,T+1}}{\ln N} \right) \Delta_{i,T}. \quad (12)$$

We call the resulting algorithm isoML-Prod (see Algorithm 4) and we prove an adaptive scale-free bound. The proof is given in Appendix D.

---

**Algorithm 4** The isoML-Prod algorithm, which uses one learning rate per coordinate and does not have prior knowledge of the range of the losses.

---

**def isoML\_Prod():**

$\forall i \in [N] : \Delta_{i,0} = 0, \mathbf{x}_{i,1} = 1$

**for**  $t = 1, 2, \dots$ :

$\forall i \in [N] : \tilde{\mathbf{x}}_{i,t} = \mathbf{x}_{i,t} / \Delta_{i,t-1}$

**Predict normalized**  $\tilde{\mathbf{x}}$ ; **observe**  $\ell_t$

$\bar{\ell}_t = \sum_{i \in [N]} \tilde{\mathbf{x}}_{i,t} \ell_{i,t} / \sum_{i \in [N]} \tilde{\mathbf{x}}_{i,t}$

**for each**  $i \in [N]$ :

**if**  $\max_{j \in [N]} q |\bar{\ell}_t - \ell_{j,t}| < \Delta_{j,t-1}$ :

$r = \bar{\ell}_t - \ell_{i,t}$

$\eta = \ln N / \Delta_{i,t-1}$

**# Adapt-ML-Prod (core) update:**

$\mathbf{x}'_{i,t} = \mathbf{x}_{i,t} (1 + \eta r)$

$\delta_{i,t} = r - \frac{1}{\eta} \ln(1 + \eta r)$  **# isotuning**

**else:** **#  $t \in \mathcal{T}_\emptyset$**

$\mathbf{x}'_{i,t} = \mathbf{x}_{i,t}$  **# null update**

$\delta_{i,t} = \max_{j \in [N]} |\bar{\ell}_t - \ell_{j,t}|$

$\Delta_{i,t} = \Delta_{i,t-1} + \delta_{i,t}$  **# isotuning**

$\mathbf{x}_{i,t+1} = \mathbf{x}'_{i,t} \frac{\Delta_{i,t-1}}{\Delta_{i,t}} + \frac{\delta_{i,t}}{\Delta_{i,t}} \mathbf{x}_{i,1}$  **# online cor.**

---

**Theorem 24.** Assume  $N \geq 2$ . We consider losses  $\ell_{i,t} \in \mathbb{R}$  for all experts  $i \in [N]$  and time steps  $t \in \mathbb{N}_1$ . Let  $S = \max_{t \in [T]} s_t$ . Then for all  $i \in [N]$ , of isoML-Prod is bounded by

$$\mathcal{R}_{i,T} \leq \left[ \sqrt{V_{i,T} \ln N} + S(2 + 3 \ln N) \right] \left[ 2 + \frac{\ln(1 + C)}{\ln N} \right]$$

where simultaneously  $C \leq T$ ,  $C \leq \tau + \ln \frac{ST}{s_\tau}$ , and also  $C = o(1) + \ln ST$ .  $\diamond$

## 5 RELATED WORK

**Null updates.** Skipping updates was used in the bandit literature to avoid a large penalty due to long delays (Thune et al., 2019; Zimmert and Seldin, 2020). Concurrently to this paper,<sup>6</sup> Huang et al. (2021) extend this line of work to obtain scale-free (but only partially adaptive) guarantees in the same context.

**Online correction.** As far as we are aware, the online correction of Orseau et al. (2017) has only ever been used in their paper. Other correction terms appear in the literature (*e.g.*, Gaillard et al. (2014); Chen et al. (2021); Huang et al. (2021)) for specific purposes, but none seem as general as the former.

**Loss clipping.** To tackle the off-by-one issue, Cutkosky (2019) proposes a correction term of the loss scales, at the expense of an additive term  $\max_t \|\mathbf{x}_t - \mathbf{x}^*\| \max_t \|\ell_t\|$ , and requires a special treatment for  $t = 0$  akin to a null update. In itself, this correction is not enough and must be combined with an additional mechanism such as the restarting of Mhammedi et al. (2019) to obtain scale-free regret bounds. The SOLO-FTRL bound can easily be recovered with this combination, but not apparently that of Theorem 20 and, by contrast to null updates, it does not help dealing with constraint violations of the update rule as for Prod (Section 4.3), or the offsets in the Prod (Section 4.3) and AdaHedge (Appendix E) bounds. Null updates, combined with isotuning and in particular Corollary 15, avoid the need to restart the algorithm altogether.

**Isotuning.** As mentioned before, the isotuning assumptions of Assumption 1 come from the AdaHedge update scheme (de Rooij et al., 2014), and were adapted to FTRL in AdaFTRL only for domains with bounded diameter *and* bounded Bregman divergence (Orabona and Pál, 2018). Both works use an ad-hoc analysis of the  $\Delta_T$  quantity, while we provide a

<sup>6</sup>Both papers were submitted to AISTATS 2022, but we refrained from uploading our paper to arXiv before receiving the reviews to maintain anonymity.

general framework where we only need to express  $\delta_t$  as a function of  $\Delta_t$  and then invoke Theorem 7 to obtain a first bound. The off-by-one issue is trivially dealt with in AdaHedge because the additive term  $\max_t \delta_t$  is bounded by the maximum loss range (see Appendix E), and is only partially dealt with in AdaFTRL, leading to the heavy constraints mentioned above. By contrast, our isotuning framework comes with more general tools to tackle this issue, in particular Lemma 13 and null updates. See also the comparison between isotuning and the more standard learning rate  $\eta_t = 1/(\sum_t \|\ell_t\|^2)$  in Appendix A.3.

**Unbounded domains.** Let  $L = \max_{t \in [T]} \|\ell_t\|$ ,  $U_T = L \sum_{t \in [T]} \|\ell_t\|$ ,  $V_T = \sum_{t \in [T]} \|\ell_t\|^2$ . For unbounded domains, FTRL-SOLO has regret  $\mathcal{R}_1 = O((\sqrt{q} + \frac{\|\mathbf{x}^*\|^2}{\sqrt{q}})\sqrt{V_T} + L\sqrt{qT})$ , which we improve to  $\mathcal{R}_2 = O((\sqrt{q} + \frac{\|\mathbf{x}^*\|^2}{\sqrt{q}})\sqrt{V_T} + L\sqrt{q\tau})$  where  $\tau$  is the last step where  $\|\ell_\tau\|^2 \geq \sum_{t < \tau} \|\ell_t\|^2$ . FreeRange (Mhammedi and Koolen, 2020) manage to obtain (almost)  $\mathcal{R}_3 = O(\|\mathbf{x}^*\| \sqrt{V_T \log(T\|\mathbf{x}^*\|)} + \sqrt{U_T} + L\|\mathbf{x}^*\|^3 + L\|\mathbf{x}^*\| \ln T)$ , and they give a matching lower bound (in particular  $U_T$  cannot be replaced with  $V_T$  without modifying the tradeoff elsewhere).  $\mathcal{R}_2$  is a strict improvement over  $\mathcal{R}_1$  since  $\tau$  is expected to be a small constant in most situations. When defaulting to  $q = 1$  ( $q = \ln T$  could also be considered),  $\mathcal{R}_2$  and  $\mathcal{R}_3$  are not comparable in general: in the worst case  $\mathcal{R}_3 < \mathcal{R}_2$  by a less-than-constant factor  $\|\mathbf{x}^*\|/\sqrt{\ln T}$  when  $V_T = U_T = \Theta(L\sqrt{T})$  but, assuming that  $\tau$  is a small constant, then  $\mathcal{R}_2 < \mathcal{R}_3$  whenever  $\|\mathbf{x}^*\| \leq \sqrt{\ln T}$  or  $\|\mathbf{x}^*\| \geq \sqrt{V_T}/L$ . The remaining cases are not easy to compare since we may have  $V_T \ll U_T$ , *e.g.*, if  $\ell_t = 100/\sqrt{t}$  (thus  $\tau = 1$ ) and  $T = 100\,000$ , then  $L\sqrt{T} \approx 31\,000$ ,  $\sqrt{U_T} \approx 2\,500$ ,  $\sqrt{V_T} \approx 350$  (and observe that  $\sqrt{\ln T} \approx 3.39$  and  $\sqrt{V_T}/L \approx 3.48$ ). The FreeRange bound also has a leading  $\sqrt{24 \ln T}$  factor and an additive  $48L\|\mathbf{x}^*\| \ln T$  term which prevent constant regret for cases where  $V_T = O(1)$ . Furthermore, FreeRange and FreeGrad results are derived only for the 2-norm, while FTRL and Mirror Descent can be used with any norm and, more generally, convex regularizers and Bregman divergences.

Hence restoring the main dependency on  $V_T$  is desirable and non-trivial, and it can lead to finite regret with stochastic losses, adaptation to smoothness and to strong convexity and has various applications (Gaillard et al., 2014, ICML 2020 parameter-free tutorial<sup>7</sup>).

For bounded domains, see the discussion in Example 42.

**Squint+L and isoML-Prod.** Mhammedi et al. (2019) proposed the Squint+L algorithm for

<sup>7</sup><https://parameterfree.com/icml-tutorial>

the simplex, which impressively mixes a continuous range of learning rates in closed form. It enjoys a scale-free adaptive regret bound of  $O(\sqrt{(\text{RE}(\mathbf{x}^*|\mathbf{x}_1) + \ln \ln T) \sum_i \mathbf{x}_i^* V_{i,T}})$  within  $O(N)$  computation steps per round. With  $\mathbf{x}_1$  in the simplex, the bound for isoML-Prod can be shown to be  $O(\sum_i \mathbf{x}_i^* (1 + \frac{\ln \ln T}{-\ln \mathbf{x}_{i,1}}) \sqrt{\ln(1/\mathbf{x}_{i,1}) V_{i,T}})$  (assuming losses do not grow exponentially). This is pretty close (using Cauchy-Schwarz, see Remark 45, p.25) but not quite the Squint+L bound. The main advantage of isoML-Prod is its far better numerical stability: Even a careful (and complicated) implementation<sup>8</sup> of Squint fails with linear regret for losses drawn *uniformly* in  $[0, 1e-3]$ ; Squint+L works a little better but starts failing for  $[0, 1e-5]$  losses, which defeats the scale-free guarantee. By contrast, a straightforward implementation of isoML-Prod as per Algorithm 4 on standard IEEE double precision floats works evenly (with near-constant regret) even for losses drawn uniformly in  $[0, 1e-300]$ . This is because the algorithm itself is invariant to loss rescaling, by contrast to Squint+L. Furthermore, whether Adapt-ML-Prod could be extended to have scale-free guarantees (and whether BOA could be fixed) without a doubling trick was still an open problem (see also Remark 44 in Appendix D).

Chen et al. (2021) propose a new algorithm with similar guarantees to Squint (but not Squint+L), and use the techniques designed for the latter to derive partially scale-free regret bounds. However, their term  $V_T$  is lower bounded by 3 preventing the regret bound from being properly scale-free for small losses—by contrast to the Squint+L bounds and our bounds. This appears to be due to constraints on the learning rate, hence it seems plausible that null updates could be used to restore the scale-free property. This algorithm also requires  $O(\log T)$  learning rates and as many copies of the algorithm, and the learning rates impractically depends on the initial guess  $B_0$  of the scale of the losses.

## 6 CONCLUSION

Isotuning, null updates and the online correction can help design and analyze anytime, scale-free and adaptive online learning algorithms for constrained and unconstrained domains. One particular advantage of our approach is that the algorithms do not need to be restarted. All algorithms studied also run as fast as online gradient descent, and the tools we developed do *not* introduce any additional line search steps or

logarithmic factors, neither in computation nor in the regret bounds.

### Acknowledgements

The authors would like to thank Tor Lattimore, Andras Gyorgy, and Pooria Joulani as well as some anonymous reviewers for their useful feedback.

### References

- Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 928–935, 2003.
- Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48 – 75, 2002.
- Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018.
- Roman Pogodin and Tor Lattimore. On first-order bounds, variance and gap-dependent bounds for adversarial bandits. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 894–904. PMLR, 22–25 Jul 2020.
- Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, Hedge if you must. *J. Mach. Learn. Res.*, 15(1):1281–1316, January 2014.
- Zakaria Mhammedi and Wouter M. Koolen. Lipschitz and comparator-norm adaptivity in online learning. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2858–2887. PMLR, 2020.
- Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3): 321–352, 2007.
- Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.
- Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, Jerusalem, 2007.

<sup>8</sup><https://bitbucket.org/wmkoolen/squint>, modified for Squint+L according to Mhammedi et al. (2019, Algorithm 1), and  $R - V$  replaced with  $R - V/B_{\tau_1}$  in the weight update.

- Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.
- Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. In *Advances in Neural Information Processing Systems 32*, pages 6538–6547, 2019.
- Julian Zimmert and Yevgeny Seldin. An optimal algorithm for adversarial bandits with arbitrary delays. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3285–3294. PMLR, 2020.
- Laurent Orseau, Tor Lattimore, and Shane Legg. Soft-bayes: Prod for mixtures of experts with log-loss. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory*, volume 76 of *Proceedings of Machine Learning Research*, pages 372–399, 2017.
- Pierre Gaillard, Gilles Stoltz, and Tim van Erven. A second-order bound with excess losses. In *Proceedings of The 27th Conference on Learning Theory, COLT*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 176–196, 2014.
- Olivier Wintenberger. Optimal learning with bernstein online aggregation. *Machine Learning*, 106(1):119–141, 2016.
- Tor Lattimore and Csaba Szepesvári. Exploration by optimisation in partial monitoring. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2488–2515, 09–12 Jul 2020.
- Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20*, pages 65–72, 2007.
- László Györfi and György Ottucsák. Sequential prediction of unbounded stationary time series. *IEEE Transactions on Information Theory*, 53(5):1866–1872, 2007.
- Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Impossible tuning made possible: A new expert algorithm and its applications. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134, pages 1216–1259. PMLR, 15–19 Aug 2021.
- Jiatai Huang, Yan Dai, and Longbo Huang. Scale-free adversarial multi-armed bandit with arbitrary feedback delays. *CoRR*, abs/2110.13400, 2021.
- H. Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, 18(90):1–50, 2017.
- Ashok Cutkosky. Artificial constraints and hints for unbounded online learning. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 874–894. PMLR, 2019.
- Zakaria Mhammedi, Wouter M Koolen, and Tim Van Erven. Lipschitz adaptivity with multiple learning rates in online learning. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2490–2511, 2019.
- Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, May 1997.
- Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- Tim van Erven, Wouter M Koolen, Steven de Rooij, and Peter Grünwald. Adaptive hedge. In *Advances in Neural Information Processing Systems*, pages 1656–1664, 2011.
- Andras György and Pooria Joulani. Adapting to delays and data in adversarial multi-armed bandits. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 3988–3997. PMLR, 2021.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.
- Mark Herbster and Manfred K. Warmuth. Tracking the best regressor. In *Proc. 11th Annu. Conf. on Comput. Learning Theory*, pages 24–31. ACM Press, New York, NY, 1998.
- Jaouad Mourtada and Odalric-Ambrym Maillard. Efficient tracking of a growing number of experts. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory (ALT)*, volume 76 of *Proceedings of Machine Learning Research*, pages 517–539, 2017.
- Thomas M Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

# A ISOTUNING: ADDITIONAL RESULTS AND REMARKS

**Proposition 25** (Existence and uniqueness of  $X_t$ ). If assumptions (i) and (ii) of Definition 5 are satisfied, then  $X_t$  as defined in assumption (iv) exists and is unique.  $\diamond$

*Proof.* Consider any  $t \in \mathbb{N}_1$ . Let  $h_t(x) = X_{t-1} + g_t(x) - x$  for  $x > g_t^{\text{low}}$ . Since  $g_t$  is nonnegative, non-increasing and continuous, then  $h_t$  is strictly decreasing and continuous. Since by assumption  $g_t^{\text{low}} \leq X_{t-1} + g_t(g_t^{\text{low}})$ , we have  $h_t(g_t^{\text{low}}) \geq 0$ . Then, since  $\lim_{x \rightarrow \infty} h_t(x) = -\infty$ , by the intermediate value theorem,  $h_t$  has exactly one root,  $X_t$ .  $\square$

**Remark 26.** In particular, existence is ensured if  $g_t^{\text{low}} = 0$  or  $g_t(g_t^{\text{low}}) = \infty$ .  $\diamond$

**Remark 27.** Since  $g_t$  is nonnegative, necessarily  $X_t \geq X_{t-1}$ .  $\diamond$

## A.1 Tighter Isotuning Bound

The factor 2 guaranteed by Theorem 7, although small, is a little loose (by at most a factor  $\sqrt{2}$ ), in particular in the main region of interest, around  $O(\sqrt{T})$  regret. For a class of cases, we can show a tighter constant with the following lemma, which merely uses convexity. (See also Gaillard et al. (2014, Lemma 14), but their lemma is suboptimal for values greater than 1, as it leads to a *factor* of the largest  $\delta_t$ .)

**Lemma 28.** Let  $(T_1, T_2) \in \mathbb{N}_0^2$  and  $T_1 \leq T_2$ , and for all  $t \in [T_1 + 1, T_2]$ , let  $\delta_t \geq 0$  and  $\Delta_t = \Delta_{t-1} + \delta_t$ . Let  $f : [0, \infty) \rightarrow \mathbb{R}$  be a convex function (that is,  $f'$  is increasing). Then

$$\sum_{t=T_1+1}^{T_2} \delta_t f'(\Delta_{t-1}) \leq f(\Delta_{T_2}) - f(\Delta_{T_1}) \leq \sum_{t=T_1+1}^{T_2} \delta_t f'(\Delta_t).$$

If  $f$  is concave ( $f'$  is decreasing), then the inequalities are in the other directions.  $\diamond$

*Proof.* By convexity of  $f$  we have

$$\begin{aligned} f(\Delta_t) &\leq f(\Delta_{t-1}) + (\Delta_t - \Delta_{t-1})f'(\Delta_t) \\ &= f(\Delta_{t-1}) + \delta_t f'(\Delta_t), \\ f(\Delta_t) &\geq f(\Delta_{t-1}) + (\Delta_t - \Delta_{t-1})f'(\Delta_{t-1}) \\ &= f(\Delta_{t-1}) + \delta_t f'(\Delta_{t-1}), \end{aligned}$$

and the result follows by summing over  $t$  from  $T_1 + 1$  to  $T_2$ .  $\square$

**Lemma 29.** Let  $(X, g)$  be an isotuning sequence where for all  $t : g_t(x) = \frac{a_t}{x}$  with  $a_t \geq 0$ . Then for all  $T \in \mathbb{N}_1$ ,

$$\sqrt{2 \sum_{t \in [T]} a_t} - \max_{t \in [T]} a_t \leq X_T \leq \sqrt{2 \sum_{t \in [T]} a_t} \quad \diamond$$

*Proof.* For the upper bound, we use the upper bound of Lemma 28 with  $f(x) = x^2$  and  $T_1 = 0$ , then  $f'(X_t)2X_t$  and thus

$$X_T^2 \leq \sum_{t \in [T]} \frac{a_t}{X_t} 2X_t.$$

For the lower bound, let  $A = \max_{t \in [T]} a_t$ , then we use the lower bound of Lemma 28 with  $f(x) = (x + A)^2$ , then  $f'(X_{t-1}) = 2(X_{t-1} + A) \geq 2X_t$  and thus

$$(X_T + A)^2 \geq \sum_{t \in [T]} \frac{a_t}{X_t} 2(X_{t-1} + A),$$

and taking the square root finishes the proof.  $\square$

**Example 30.** Take the regret bound Eq. (1) of the introduction. With  $\delta_t = a_t \eta_t = a_t q / \Delta_t$  we can take  $f(y) = y^2$  so that from Lemma 28 we have

$$f(\Delta_T) - f(\Delta_0) \leq \sum_{t \in [T]} \frac{a_t q}{\Delta_t} \cdot 2\Delta_t = q \sum_{t \in [T]} a_t,$$

$$\Delta_T \leq \sqrt{2q \sum_{t \in [T]} a_t},$$

and thus  $\mathcal{R}_T(\mathbf{x}^*) \leq 2\sqrt{2q \sum_t a_t}$ , which is tighter by a factor  $\sqrt{2}$  than by using Theorem 7, and is only a factor  $\sqrt{2}$  away from the hindsight optimal constant learning rate. We do not know of any sequential learning rate that achieves a better rate without prior knowledge of  $T$ .  $\diamond$

See Appendix E for a use case of Lemma 28 in the off-by-one setting.

## A.2 AOGD Revisited

The set  $\mathcal{X}$  is assumed to be convex, closed and bounded. The projection  $\Pi_{\mathcal{X}}[x]$  is according to  $\|\cdot\|_2$ , that is  $\Pi_{\mathcal{X}}[x] = \operatorname{argmin}_{y \in \mathcal{X}} \|x - y\|_2^2$ .

For comparison, we recall the main theorem of Bartlett et al. (2007) for AOGD:

$$\mathcal{R}_T(\mathbf{x}^*) \leq 3 \inf_{\lambda_1^*, \dots, \lambda_T^* \geq 0} \left\{ D^2 \lambda_{1:T}^* + \sum_{t \in [T]} \frac{(\|\nabla_t\| + \lambda_t^* D)^2}{\alpha_{1:T} + \lambda_{1:T}^*} \right\}$$

where  $\lambda_{1:T} = \sum_{t \in [T]} \lambda_t$ , and  $\alpha$  is as in Section 3.2. Their update is  $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t(\nabla_t + \lambda_t \mathbf{x}_t)]$ , with  $\lambda_t = \sqrt{A^2 + 8\|\nabla_t\|^2 / (3D^2)} / 2 - A/2$ ,  $A = \alpha_{1:t} + \lambda_{1:t-1}$ , and  $\eta_t = 1/(A + \lambda_t)$ .

**Lemma 31.** Assume the loss functions  $\ell_t$  are  $\alpha_t$ -strongly convex for all  $t \in [T]$ , and that  $1/\eta_t - \alpha_t - 1/\eta_{t-1} \geq 0$ . Then the regret of online gradient descent with a time varying learning rate  $\eta_t$  is bounded by

$$2\mathcal{R}_T(\mathbf{x}^*) \leq \max_{t \in [T]} \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 \left( \frac{1}{\eta_T} - \alpha_{1:T} - \frac{1}{\eta_0} \right) + \sum_{t \in [T]} \eta_t \|\nabla_t\|_2^2 + \frac{1}{\eta_0} \|\mathbf{x}_1 - \mathbf{x}^*\|_2^2. \quad \diamond$$

*Proof.* Using the update rule on the first line and the definition of  $\alpha_t$ -strong convexity of  $\ell_t$  on the second line we have

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2 &= \|\Pi_{\mathcal{X}}[\mathbf{x}_t - \eta_t \nabla_t] - \mathbf{x}^*\|_2^2 \\ &\stackrel{(a)}{\leq} \|\mathbf{x}_t - \eta_t \nabla_t - \mathbf{x}^*\|_2^2 \\ &= \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 + \eta_t^2 \|\nabla_t\|_2^2 + 2\langle \mathbf{x}^* - \mathbf{x}_t, \eta_t \nabla_t \rangle \\ &\stackrel{(b)}{\leq} \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 + \eta_t^2 \|\nabla_t\|_2^2 + 2\eta_t(\ell_t(\mathbf{x}^*) - \ell_t(\mathbf{x}_t)) + \frac{\alpha_t}{2} \|\mathbf{x}_t - \mathbf{x}^*\|_2^2, \\ &\Leftrightarrow 2(\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}^*)) \stackrel{(c)}{\leq} \left( \frac{1}{\eta_t} - \alpha_t \right) \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 - \frac{1}{\eta_t} \|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2 + \eta_t \|\nabla_t\|_2^2, \\ 2 \sum_{t \in [T]} (\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}^*)) &\stackrel{(d)}{\leq} \sum_{t \in [T]} \left[ \left( \frac{1}{\eta_t} - \alpha_t - \frac{1}{\eta_{t-1}} \right) \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 + \eta_t \|\nabla_t\|_2^2 \right] \\ &\quad - \underbrace{\frac{1}{\eta_T} \|\mathbf{x}_{T+1} - \mathbf{x}^*\|_2^2 + \frac{1}{\eta_0} \|\mathbf{x}_1 - \mathbf{x}^*\|_2^2}_{\geq 0} \\ &\leq \max_t \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 \left( \frac{1}{\eta_T} - \alpha_{1:T} - \frac{1}{\eta_0} \right) + \sum_{t \in [T]} \eta_t \|\nabla_t\|_2^2 + \frac{1}{\eta_0} \|\mathbf{x}_1 - \mathbf{x}^*\|_2^2, \end{aligned}$$

where (a) is by the Pythagorean identity since  $\mathcal{X}$  is a closed and bounded convex set and  $\mathbf{x}^* \in \mathcal{X}$ , (b) follows from the  $\alpha_t$ -strong convexity of  $\ell_t$ , (c) is by dividing by  $\eta_t$  and rearranging, (d) is by summing over  $t$  and factoring by  $\|\mathbf{x}_t - \mathbf{x}^*\|_2^2$ .  $\square$



### A.3 Comparison To The Sequential Hindsight-Optimal Learning Rate

Recall Definition 5 and Theorem 7. For all  $T \in \mathbb{N}_1$ , let  $x_T^* = \operatorname{arginf}_{x > g_T^{\inf}} \left\{ x + \sum_{t \in [T]} g(x) \right\}$ . Then  $M_T(x_T^*) = M_T^*$ . We write  $x_{1:T}^* \equiv (x_t^*)_{t \in [T]}$  and, abusing notation, define  $M_T(x_{1:T}^*) = x_T^* + \sum_{t \in [T]} g_t(x_t^*)$ . We call  $x_{1:T}^*$  the *sequential hindsight-optimal* ‘learning rate.’

It is quite common to use (a variant of)  $x_{1:T}^*$  in adaptive learning rates (*e.g.*, Auer et al. (2002); Cesa-Bianchi and Lugosi (2006); Orabona and Pál (2018); Pogodin and Lattimore (2020); György and Joulani (2021); Gaillard et al. (2014) and many others). We can indeed prove nice and general upper and lower bounds for  $M_T(x_{1:T}^*)$ :

**Theorem 32.** For all  $T \in \mathbb{N}_1$ , with  $M_T$ ,  $M_T^*$ ,  $x_T^*$ , and  $M_T(x_{1:T}^*)$  defined as above,

$$M_T^* \leq M_T(x_{1:T}^*) \leq x_T^* + M_T^* \quad (\leq 2M_T^*). \quad \diamond$$

*Proof.* For the upper bound,

$$M_T^* \geq M_{T-1}^* + g_T(x_T^*) \geq M_{T-2}^* + g_{T-1}(x_{T-1}^*) + g_T(x_T^*) \geq \dots \geq M_0^* + \sum_{t \in [T]} g_t(x_t^*),$$

hence,

$$M_T(x_{1:T}^*) = x_T^* + \sum_{t \in [T]} g_t(x_t^*) \leq x_T^* + M_T^*.$$

And for the lower bound since  $g_t$  is non-increasing:

$$M_T(x_{1:T}^*) = x_T^* + \sum_{t=1}^T g_t(x_t^*) \geq x_T^* + \sum_{t=1}^T g_t(x_T^*) = M_T^*. \quad \square$$

**Example 33.** For the regret bound of Eq. (1) in the introduction, using the learning rate  $x_{1:t}^* \equiv 1/\eta_t$ , with  $M_T(x) = qx + \sum_t a_t/x$ , one can then use Theorem 32 with  $x_T^* = \operatorname{arginf}_{x > 0} M_T(x) = \sqrt{q \sum_{t \in [T]} a_t}$  to obtain straightforwardly (see also the well-known Lemma 3.5 from Auer et al. (2002)),

$$\mathcal{R}_T(\mathbf{x}^*) \leq M_T(x_{1:T}^*) \leq x_T^* + M_T^* = 3 \sqrt{q \sum_{t \in [T]} a_t} \quad (= \frac{3}{2} M_T(x_T^*)).$$

This factor 3 is tighter than the factor 4 obtained with Theorem 7 for the isotuning learning rate  $\Delta_T$ , but worse than the factor  $2\sqrt{2}$  that can be obtained via Lemma 28 also for isotuning (see Example 30), that is,  $M_T(\Delta_T) \leq \sqrt{2} M_T^*$ . It is known, however, that one can also obtain a factor  $2\sqrt{2}$  by ‘manually’ rebalancing  $x_{1:T}^*$  and taking  $\tilde{x}_t^* = x_t^* \sqrt{2}$  in place of  $x_t^*$ , but this rebalancing is dependent on the regret bound.  $\diamond$

**Example 34.** When  $M_T^* = x_T^* + O(1)$  we can see that  $M_t(x_{1:T}^*)$  may reach its upper bound of  $2M_T^*$  in Theorem 32. Indeed this is reached in particular for logarithmic ‘regret:’

$$\begin{aligned} g_t(x) &= e^{-x}, & M_T(x) &= x + T e^{-x}, & M_T^* &= \ln T + 1, & x_T^* &= \ln T, \\ M_T(x_{1:T}^*) &= x_T^* + \sum_{t \in [T]} g_t(x_t^*) &= \ln T + \sum_{t \in [T]} e^{-\ln t} &\geq 2 \ln T - O(1) &= 2M_T^* - O(1). \end{aligned} \quad \diamond$$

Hence the sequential hindsight-optimal learning rate and the isotuning learning rate behave rather similarly for the regret bound of the introduction. However, in the off-by-one setting, by contrast to  $x_{1:T}^*$ , isotuning ensures that the additive term is  $\max_t \delta_t$  (which was obtained using the isotuning-specific property  $\Delta_{t-1} = \Delta_t - \delta_t$ ), which in some circumstances may be nicely bounded (see for example Appendix E). This property of isotuning is what allows us to develop our theory in Section 3.4 and Section 3.5. In particular for null updates, we could bound the remaining additive term  $\max_{t \notin \mathcal{T}_\emptyset} \delta_t \leq c \Delta_{t-1}$  and use Eq. (6) to further bound with a value that is independent of the learning rate.

For the sequential hindsight-optimal learning rate in the off-by-one setting, the matter is not as nice, as shown in the following example.

**Example 35.** Consider the introductory example of Section 3.5, where for isotuning we have <sup>9</sup>  $\max_t \delta_t \leq D \|\nabla_t\|_2$ . By contrast, using  $x_t^* = \sqrt{\sum_{s \in [t]} a_s}$  (assuming  $q = 1$ ) in the off-by-one setting we do not know of a significantly better lemma than the following bound (see also for example Gaillard et al. (2014, Eq. (25)), Pogodin and Lattimore (2020, Lemma 4.8), Duchi et al. (2011) and others):

$$\sum_{t \in [T]} \frac{a_t}{\varepsilon + x_{t-1}^*} \leq \sum_{t \in [T]} \frac{a_t}{\varepsilon + x_t^*} + \max_{t \in [T]} a_t \sum_{t \in [T]} \left( \frac{1}{\varepsilon + x_{t-1}^*} - \frac{1}{\varepsilon + x_t^*} \right) \leq \sum_{t \in [T]} \frac{a_t}{x_t^*} + \max_{t \in [T]} \frac{a_t}{\varepsilon}.$$

Combining with Theorem 32 we obtain for  $1/\eta_t = \varepsilon + x_t^* = \varepsilon + \sqrt{\sum_{s \in [t]} a_s}$  for the regret bound of Eq. (1) in the off-by-one-setting

$$\mathcal{R}_T(\mathbf{x}^*) \leq \frac{1}{\eta_T} + \sum_{t \in [T]} \eta_{t-1} a_t \leq \varepsilon + x_T^* + \sum_{t \in [T]} \frac{a_t}{\varepsilon + x_{t-1}^*} \leq 3 \sqrt{\sum_{t \in [T]} a_t} + \max_t \frac{a_t}{\varepsilon} + \varepsilon.$$

Hence in this bound the additive term depends on  $\varepsilon > 0$  which must be chosen in advance, and optimally should be  $\max_t \sqrt{a_t}$  (which would also give a scale-free bound), while this a quantity that may not be known in advance. To track  $\varepsilon$  over time, one may have to resolve to restarting the algorithm altogether (*e.g.*, Mhammedi et al. (2019)).  $\diamond$

**Remark 36.** Another advantage of isotuning is that the learning rate is updated stepwise with  $\Delta_t = \Delta_{t-1} + \delta_t$ , rather than calculated explicitly as for  $x_{1:t}^*$ . While this does not matter for the simple cases we have encountered in this paper, one could conceive of a regret bound of the form, with  $\alpha_t > 0$  and  $a_t \in [0, 1]$  for all  $t$ ,

$$\mathcal{R}_T(\mathbf{x}^*) \leq \frac{q}{\eta_T} + \sum_{t \in [T]} (\eta_{t-1})^{\alpha_t} a_t.$$

Because there is no closed form for the optimal constant learning rate in hindsight, one would have to make strong and likely undesirable assumptions to design an adaptive learning rate based on  $x_{1:t}^*$  and analyze the regret, or use a growing number of learning rates. By contrast, with isotuning one simply needs to set  $\delta_t = (\eta_{t-1})^{\alpha_t} a_t = (q/\Delta_{t-1})^{\alpha_t} a_t$ , and use Corollary 15 and Theorem 7 to obtain a regret bound that is within a factor 2 of the (unknown!) optimal constant learning rate in hindsight.  $\diamond$

## B SCALE-FREE FTRL: DETAILED PROOF

Define  $\tilde{L}_t = \sum_{s \in [t]} \tilde{\ell}_s$ . When  $\tilde{L}_t = 0$ , as for SOLO-FTRL (Orabona and Pál, 2018) we choose  $\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} R(\mathbf{x})$ .

First we need a few of lemmas. The first one gives a generic lower bound on  $\Delta_T$  when using null updates, then we provide upper and lower bounds on  $\Delta_T$  for the problem at hand. Finally, we upper bound the ‘travel distance’  $\|\mathbf{x}_t - \mathbf{x}^*\|$  before proving the main theorem.

**Theorem 37** (Isotuning lower bound, null updates,  $c = 1$ ). Consider the conditions of Theorem 17, but assume that for  $t \notin \mathcal{T}_\emptyset$ ,  $\delta_t = \hat{\delta}_t(\Delta_{t-1})$  while for  $t \in \mathcal{T}_\emptyset$ ,  $\delta_t = \operatorname{iso}_{y>0}(\hat{\delta}_t(y), y)$ . Then for all  $T \in \mathbb{N}_1$ ,

$$\Delta_T \geq \operatorname{iso}_{y>0} \left( \sum_{t \in [T]} \hat{\delta}_t(y), y \right). \quad \diamond$$

*Proof.* Since for  $t \in \mathcal{T}_\emptyset$ ,  $\delta_t = \operatorname{iso}_{y>0}(\hat{\delta}_t(y), y)$ , then by Lemma 3,

$$\min \left\{ \hat{\delta}_t(\Delta_t), \Delta_t \right\} \leq \delta_t \leq \max \left\{ \hat{\delta}_t(\Delta_t), \Delta_t \right\}$$

and since  $\delta_t \leq \Delta_t$  then necessarily  $\hat{\delta}_t(\Delta_t) \leq \delta_t$ . Then, recalling that  $\hat{\delta}_t$  is non-increasing,

$$\Delta_T = \sum_{t \in [T]} \delta_t \geq \sum_{t \notin \mathcal{T}_\emptyset} \hat{\delta}_t(\Delta_{t-1}) + \sum_{t \in \mathcal{T}_\emptyset} \hat{\delta}_t(\Delta_t) \geq \sum_{t \in [T]} \hat{\delta}_t(\Delta_T).$$

<sup>9</sup>Since  $\eta_0^{-1} = 0$ , for bounded domains from Eq. (5) we can take  $\delta_1 \geq (r_1)_+$ , and  $r_1 \leq \|\mathbf{x}_1 - \mathbf{x}^*\|_2 \|\nabla_1\|_2 \leq D \|\nabla_1\|_2$ .

Using Lemma 3 again on the inequality above we obtain

$$\operatorname{iso}_{y>0} \left( \sum_{t \in [T]} \hat{\delta}_t(y), y \right) \leq \max \left\{ \Delta_T, \sum_{t \in [T]} \hat{\delta}_t(\Delta_T) \right\} = \Delta_T. \quad \square$$

**Lemma 38** ( $\Delta_T$  bounds). For Algorithm 2, for all  $T \in \mathbb{N}_1$ ,  $\Delta_T$  satisfies

$$\sqrt{\frac{q}{2} \sum_{t \in [T]} \|\ell_t\|_*^2} \leq \Delta_T \leq \sqrt{q \sum_{t \in [T]} \|\ell_t\|_*^2} + \sqrt{\frac{q}{2} \max_{t \in [T]} \|\ell_t\|_*}. \quad \diamond$$

*Proof.* Recalling that  $\hat{\delta}_t(y) = q\|\ell_t\|_*^2/(2y)$ , the lower bound follows by a straightforward application of Theorem 37, while the upper bound follows similarly from Theorem 17 (i) and Lemma 10.  $\square$

We now prove the same identity as Orabona and Pál (2018) about the maximum travel distance in the off-by-one setting.

**Lemma 39** (Upper bounding  $\|\mathbf{x}^* - \mathbf{x}_t\|$ ). For Algorithm 2, for all  $\mathbf{x} \in \mathcal{X}$ ,

$$\|\mathbf{x} - \mathbf{x}_t\| \leq \sqrt{2R(\mathbf{x})} + 2\sqrt{2qt}. \quad \diamond$$

*Proof.* Recall that  $\tilde{L}_t = \sum_{s \in [t]} \tilde{\ell}_s$ , and that if  $\tilde{L}_t = 0$ , then  $\mathbf{x}_t = \mathbf{x}_1$ . We decompose the update  $\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \left\{ \langle \tilde{L}_{t-1}, \mathbf{x} \rangle + \frac{R(\mathbf{x})}{\eta_{t-1}} \right\}$  into the unconstrained update  $\hat{\mathbf{x}}_t = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^N} \left\{ \langle \tilde{L}_{t-1}, \mathbf{x} \rangle + \frac{R(\mathbf{x})}{\eta_{t-1}} \right\}$  and the projection onto  $\mathcal{X}$ ,  $\mathbf{x}_t = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} B_R(\mathbf{x}, \hat{\mathbf{x}}_t)$ . Hence by the generalized Pythagorean identity (e.g., Herbster and Warmuth (1998) and references therein), for all  $\mathbf{x} \in \mathcal{X}$ ,

$$B_R(\mathbf{x}, \mathbf{x}_t) \leq B_R(\mathbf{x}, \hat{\mathbf{x}}_t).$$

Furthermore, for all  $\mathbf{x} \in \mathcal{X}$ ,

$$\begin{aligned} B_R(\mathbf{x}, \hat{\mathbf{x}}_t) &= R(\mathbf{x}) - R(\hat{\mathbf{x}}_t) - \langle \mathbf{x} - \hat{\mathbf{x}}_t, \nabla R(\hat{\mathbf{x}}_t) \rangle \\ &\stackrel{(a)}{\leq} R(\mathbf{x}) + \|\mathbf{x} - \hat{\mathbf{x}}_t\| \|\nabla R(\hat{\mathbf{x}}_t)\|_* \\ &\stackrel{(b)}{\leq} R(\mathbf{x}) + \sqrt{2B_R(\mathbf{x}, \hat{\mathbf{x}}_t)} \eta_{t-1} \|\tilde{L}_{t-1}\|_*, \end{aligned}$$

where (a) since  $R(\cdot) \geq 0$  by assumption and using Hölder's inequality (generalized to dual norms), (b) using the 1-strong convexity of  $R$  with respect to  $\|\cdot\|$  and  $\nabla R(\hat{\mathbf{x}}_t) = -\eta_{t-1} \tilde{L}_{t-1}$ . This inequality is of the form  $x^2 \leq c + bx$  with  $x, b, c \geq 0$ , which implies  $x \leq \sqrt{c} + b$  and thus

$$\|\mathbf{x} - \mathbf{x}_t\| \leq \sqrt{2B_R(\mathbf{x}, \mathbf{x}_t)} \leq \sqrt{2B_R(\mathbf{x}, \hat{\mathbf{x}}_t)} \leq \sqrt{2R(\mathbf{x})} + 2\eta_{t-1} \|\tilde{L}_{t-1}\|_*.$$

Finally, using Cauchy-Schwarz on the first inequality, and Assumption 1,

$$\eta_{t-1}^2 \|\tilde{L}_{t-1}\|_*^2 \leq \eta_{t-1}^2 (t-1) \sum_{s < t} \|\tilde{\ell}_s\|_*^2 \leq \eta_{t-1} (t-1) \sum_{s < t} \eta_{s-1} \|\tilde{\ell}_s\|_*^2 \leq \eta_{t-1} (t-1) 2\Delta_{t-1} \leq 2qt,$$

which finishes the proof.  $\square$

We are now ready to prove the main result.

*Proof of Theorem 20.* We split the regret in two terms:

$$\mathcal{R}_T(\mathbf{x}^*) = \sum_{t \in [T]} \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle = \sum_{t \in [T] \setminus \mathcal{T}_0} \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle + \sum_{t \in \mathcal{T}_0} \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle.$$

The first term corresponds to the regret of FTRL with a time-varying learning rate running on the pseudo-losses  $\tilde{\ell}_t$ , as per Lemma 19:

$$\sum_{t \in [T] \setminus \mathcal{T}_0} \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle = \sum_{t \in [T]} \langle \mathbf{x}_t - \mathbf{x}^*, \tilde{\ell}_t \rangle \leq \left( 1 + \frac{R(\mathbf{x}^*)}{q} \right) \Delta_T,$$

and  $\Delta_T$  is bounded from above in Lemma 38.

The second term corresponds to the losses incurred on null updates, which are not accounted for by FTRL on the pseudo losses:

$$\begin{aligned}
\sum_{t \in \mathcal{T}_0} \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle &\stackrel{(a)}{\leq} \sum_{t \in \mathcal{T}_0} \|\mathbf{x}_t - \mathbf{x}^*\| \|\ell_t\|_* \leq \max_{t \in \mathcal{T}_0} \|\mathbf{x}_t - \mathbf{x}^*\| \sum_{t \in \mathcal{T}_0} \|\ell_t\|_* \\
&\stackrel{(b)}{=} \max_{t \in \mathcal{T}_0} \|\mathbf{x}_t - \mathbf{x}^*\| \sum_{t \in \mathcal{T}_0} \sqrt{\frac{2}{q}} \delta_t \leq \max_{t \in \mathcal{T}_0} \|\mathbf{x}_t - \mathbf{x}^*\| \sqrt{\frac{2}{q}} \Delta_\tau \\
&\stackrel{(c)}{\leq} 2 \min\{D, \sqrt{2R(\mathbf{x}^*)} + 2\sqrt{2q\tau}\} \|\ell_\tau\|_*.
\end{aligned}$$

where (a) follows from Hölder's inequality (generalized to dual norms), (b) by definition of  $\delta_t$  for  $t \in \mathcal{T}_0$ , and (c) from Lemma 39 and Theorem 17 (ii).

**Lower bounding  $\|\ell_\tau\|_*$ .** Since  $\tau \in \mathcal{T}_0$ , from Eq. (7) we have  $\Delta_{\tau-1} \leq \sqrt{q/2} \|\ell_\tau\|_*$ , hence  $\|\ell_\tau\|_*^2 > \frac{2}{q} \Delta_{\tau-1}^2$ , and the claim is proven by using the lower bound in Lemma 38.  $\square$

## C SCALE-FREE MIRROR DESCENT: DETAILED PROOF

The proof is very similar to the FTRL case in Appendix B.

First we show that  $\delta_t$  satisfies Eq. (5) for  $t \notin \mathcal{T}_0$ :

$$\begin{aligned}
\delta_t^* &= \left( \frac{1}{\eta_{t-1}} (\phi_{t'} - \phi_t) + r_t \right)_+ = \left( \frac{1}{\eta_{t-1}} (B_R(\mathbf{x}^*, \mathbf{x}_{t'}) - B_R(\mathbf{x}^*, \mathbf{x}_t)) + \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle \right)_+ \\
&\leq \langle \mathbf{x}_t - \mathbf{x}_{t'}, \ell_t \rangle - \frac{1}{\eta_{t-1}} B_R(\mathbf{x}_{t'}, \mathbf{x}_t) \\
&\leq \|\mathbf{x}_t - \mathbf{x}_{t'}\| \|\ell_t\|_* - \frac{1}{2\eta_{t-1}} \|\mathbf{x}_t - \mathbf{x}_{t'}\|^2 \\
&\leq \max_{z \in \mathbb{R}} \left\{ \|\ell_t\|_* z - \frac{1}{2\eta_{t-1}} z^2 \right\} = \frac{\eta_{t-1}}{2} \|\ell_t\|_*^2 = \delta_t.
\end{aligned} \tag{13}$$

where on the second line we used the proof of Lemma 2 from Orabona and Pál (2018), where we take  $w_t \rightsquigarrow \mathbf{x}_t, w_{t+1} \rightsquigarrow \mathbf{x}_{t'}, u \rightsquigarrow \mathbf{x}^*, \lambda \rightsquigarrow 1$  and  $B_{R_t}(\cdot, \cdot) \rightsquigarrow B_R(\cdot, \cdot)/\eta_{t-1}$ , and recalling that the regularizer  $R$  is 1-strongly convex with respect to the norm  $\|\cdot\|$ , that is,  $B_R(a, b) \geq \|a - b\|^2/2$ .

Also note that from Eq. (13) we can deduce for  $t \notin \mathcal{T}_0$ ,

$$\delta_t \geq \frac{1}{\eta_{t-1}} B_R(\mathbf{x}_t, \mathbf{x}_{t'}) \geq 0. \tag{14}$$

**Remark 40.** We could actually take  $\delta_t = \frac{1}{\eta_{t-1}} B_R(\mathbf{x}_t, \mathbf{x}_{t'})$  without changing the regret bound.  $\diamond$

Before proving the main result, as for the FTRL case we first bound the travel distance.

**Lemma 41** (Bounding  $\|\mathbf{x}_t - \mathbf{x}_1\|$ ). For Algorithm 3, for all  $t \in \mathbb{N}_1$ ,

$$\|\mathbf{x}_{t+1} - \mathbf{x}_1\| \leq \sqrt{2qt}. \tag{15}$$

$\diamond$

*Proof.* Using Lemma 11 with  $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_1\|^2$ ,

$$\begin{aligned}
\Delta_t \|\mathbf{x}_{t+1} - \mathbf{x}_1\|^2 &\leq \Delta_{t-1} \|\mathbf{x}_{t'} - \mathbf{x}_1\|^2, \\
\sqrt{\Delta_t} \|\mathbf{x}_{t+1} - \mathbf{x}_1\| &\stackrel{(a)}{\leq} \sqrt{\Delta_{t-1}} (\|\mathbf{x}_{t'} - \mathbf{x}_t\| + \|\mathbf{x}_t - \mathbf{x}_1\|) \\
&\stackrel{(b)}{\leq} \sqrt{\Delta_{t-1}} (\sqrt{2B_R(\mathbf{x}_t, \mathbf{x}_{t'})} + \|\mathbf{x}_t - \mathbf{x}_1\|) \\
&\stackrel{(c)}{\leq} \sqrt{2q\delta_t} + \sqrt{\Delta_{t-1}} \|\mathbf{x}_t - \mathbf{x}_1\| \\
&\stackrel{(d)}{\leq} \sum_{s \in [t]} \sqrt{2q\delta_s} \\
&\stackrel{(e)}{\leq} \sqrt{t \sum_{s \in [t]} 2q\delta_s} = \sqrt{2qt\Delta_t}, \\
\|\mathbf{x}_{t+1} - \mathbf{x}_1\| &\leq \sqrt{2qt},
\end{aligned}$$

where (a) follows by  $\Delta_t = q/\eta_t$ , (b) by 1-strong convexity of  $B_R$  with respect to  $\|\cdot\|$ , (c) using Eq. (14) for  $t \notin \mathcal{T}_\emptyset$  with  $\Delta_{t-1} = q/\eta_{t-1}$ , and  $B_R(\mathbf{x}_t, \mathbf{x}_{t'}) = 0$  for  $t \in \mathcal{T}_\emptyset$ , (d) by induction, (e) by Cauchy-Schwarz.  $\square$

We can now prove the main result.

*Proof of Theorem 21.* From Theorem 16, since  $\phi_{T+1} \geq 0$  we can simplify to

$$\mathcal{R}_T(\mathbf{x}^*) \leq \left(1 + \frac{\phi_1}{q}\right) \Delta_T + \sum_{t \in \mathcal{T}_\emptyset} r_t.$$

Define  $\hat{\delta}_t(x) = q \frac{\|\ell_t\|_*^2}{2x}$  and so  $\text{iso}_{y>0}(\hat{\delta}_t(y), y) = \sqrt{q/2} \|\ell_t\|_*$ . Then from Theorem 17 (i) and Lemma 10 we have

$$\Delta_T \leq \hat{\Delta}_T + \sqrt{\frac{q}{2} \max_{t \in [T]} \|\ell_t\|_*}, \quad \hat{\Delta}_T \sqrt{q \sum_{t \in [T]} \|\ell_t\|_*^2}.$$

Since  $\delta_t = \sqrt{q/2} \|\ell_t\|_*$  for  $t \in \mathcal{T}_\emptyset$ , we have, using Theorem 17 (ii) on the inequality,

$$\sum_{t \in \mathcal{T}_\emptyset} \|\ell_t\| = \sqrt{\frac{2}{q}} \sum_{t \in \mathcal{T}_\emptyset} \delta_t \leq 2 \|\ell_\tau\|_*.$$

Therefore, using Hölder's inequality we have

$$\begin{aligned}
\sum_{t \in \mathcal{T}_\emptyset} r_t &= \sum_{t \in \mathcal{T}_\emptyset} \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle \leq \max_{t \in \mathcal{T}_\emptyset} \|\mathbf{x}^* - \mathbf{x}_t\| \sum_{t \in \mathcal{T}_\emptyset} \|\ell_t\|_* \leq 2 \max_{t \in \mathcal{T}_\emptyset} \|\mathbf{x}^* - \mathbf{x}_t\| \|\ell_\tau\|_* \\
&\leq 2 \min\{D, \|\mathbf{x}^* - \mathbf{x}_1\| + \max_{t \in \mathcal{T}_\emptyset} \|\mathbf{x}_1 - \mathbf{x}_t\|\} \|\ell_\tau\|_* \\
&\leq 2 \min\{D, \|\mathbf{x}^* - \mathbf{x}_1\| + \sqrt{2q\tau}\} \|\ell_\tau\|_*.
\end{aligned}$$

where we used Lemma 41 on the last line. Finally, the result follows by putting it all together.  $\square$

**Example 42** (Scale-Free Online Gradient Descent, MD style). When  $B_R(x, y) = \|x - y\|_2^2/2$  with  $\|\cdot\| = \|\cdot\|_* = \|\cdot\|_2$  in a bounded decision set of diameter  $D$ , then by choosing  $\mathbf{x}_1$  at the center of  $\mathcal{X}$ , using Jung's theorem we have  $\phi_1 \leq (D/\sqrt{2})^2/2$ , so taking  $q = D^2/4$  gives, taking  $\ell_t = \nabla_t$ ,

$$\mathcal{R}_T(\mathbf{x}^*) \leq D \left( \sqrt{\sum_{t \in [T]} \|\nabla_t\|_2^2} + (2 + \frac{1}{\sqrt{2}}) \max_{t \in [T]} \|\nabla_t\|_2 \right).$$

Let us call isoGD this instance of our algorithm. As far as we are aware, this bound has the best leading factor for bounded-domain OGD with an adaptive learning rate when using a quadratic regularizer. SOLO-FTRL has

a leading factor of  $\sqrt{2.75}$ , for AdaFTRL it is  $5.3/2$ , while uncentered bounds (including that of Scale-Free Mirror Descent (Orabona and Pál, 2018)) have at least a leading factor of  $\sqrt{2}$  (e.g., Hazan (2016)). McMahan (2017) also obtains a centered bound with the optimal leading factor, but it is not adaptive and it is assumed that the largest gradient is known to tune a (non-adaptive) learning rate so as to avoid the off-by-one issue.

For non-quadratic regularizers and for bounded domains, it must be noted that the FreeGrad algorithm (Mhammedi and Koolen, 2020) achieves a scale-free regret of  $O(\|\mathbf{x}^* - \mathbf{x}_1\| \sqrt{V_T \ln(\|\mathbf{x}^* - \mathbf{x}_1\| \sqrt{V_T})} + D \max_t \|\ell_t\|)$ , which is qualitatively better than our quadratic-regularizer bounds but features a (slowly) growing  $\sqrt{\ln T}$  term. In practice, the exponential component in the FreeRange update does make it move fast toward  $\mathbf{x}^*$ , but it also tends to make it take large strides back and forth once in the vicinity of  $\mathbf{x}^*$ . Let us take a concrete example. Assume  $\mathcal{X}$  is an unknown but bounded superset of  $[-10, 10]$ , and let  $\mathbf{x}^* = 3$  and  $\mathbf{x}_1 = -10$ , define the loss function  $\ell(\mathbf{x}) = \mathbf{x}^* - \mathbf{x}$  if  $\mathbf{x} < \mathbf{x}^*$  and  $\ell(\mathbf{x}) = \exp(\mathbf{x} - \mathbf{x}^*) - 1$  otherwise. We consider FreeGrad with loss-clipping and restarting with hints being the current maximum observed absolute gradient (Mhammedi and Koolen, 2020) (but without projection within an increasing domain). For isoGD we take  $q = 1$  agnostically so the leading factor of isoGD is  $|\mathbf{x} - \mathbf{x}^*|^2 = 13^2$ . See the results in Fig. 2 after  $T = 100\,000$ . Despite coming close to  $\mathbf{x}^*$  very early ( $t < 20$ ), FreeGrad incurs a large loss, restarts, moves more carefully then oscillates around  $\mathbf{x}^*$  with large strides, even after  $10^5$  rounds, incurring a large cumulative regret, both in the loss function and in the distance to  $\mathbf{x}^*$ , even on average. By contrast, isoGD moves ‘slowly’ toward  $\mathbf{x}^*$  but once in the vicinity of  $\mathbf{x}^*$  ( $t < 180$ ) it converges to  $\mathbf{x}^*$  significantly faster than FreeGrad. While this is of course just a single example, and that without doubt there are examples where FreeGrad significantly outperforms isoGD (say, probably, if  $\|\mathbf{x}^* - \mathbf{x}_1\|$  is very large), it does demonstrate that FreeGrad is not *strictly* better than isoGD even for bounded domains.  $\diamond$

**Example 43** (MD on the simplex). When the Bregman divergence is the the relative entropy  $B_R(a, b) = \sum_{i \in [N]} a_i \ln(a_i/b_i)$ , and  $\mathcal{X} = \Delta_p(N)$  is the probability simplex, then  $\|\cdot\| = \|\cdot\|_1$  and  $\|\cdot\|_* = \|\cdot\|_\infty$ ,  $\phi_1 \leq \ln N$ , and  $D = 2$  so taking  $q = \ln N$  gives

$$\mathcal{R}_T(\mathbf{x}^*) \leq 2 \sqrt{\ln N \sum_{t \in [T]} \|\ell_t\|_\infty^2} + (4 + \sqrt{2 \ln N}) \max_{t \in [T]} \|\ell_t\|_\infty.$$

See also Appendix E for a more adaptive so-called second order bound with excess losses like AdaHedge (de Rooij et al., 2014) but with a Mirror-Descent-style projection, and a loss translation lemma on the simplex, which applies to the bound above.  $\diamond$

## D ISOML-PROD DETAILED PROOF

*Proof of Theorem 24.* Recall Eq. (12):

$$\mathcal{R}_{i,T} = \sum_{t \in [T]} (\bar{\ell}_t - \ell_{i,t}) \leq \left(1 + \frac{\ln \mathbf{x}_{i,T+1}}{\ln N}\right) \Delta_{i,T}.$$

We proceed to bound  $\mathbf{x}_{i,T+1}$  and  $\Delta_{i,T}$  in turn.

**Bounding  $\mathbf{x}_{i,T+1}$ .** For  $t \notin \mathcal{T}_0$ , combining with Eq. (11) and the online correction Eq. (3) we have

$$\begin{aligned} \sum_{j \in [N]} \mathbf{x}_{j,t+1} &\leq \sum_{j \in [N]} \left[ \mathbf{x}_{j,t} (1 + \eta_{j,t-1} (\bar{\ell}_t - \ell_{j,t})) + \frac{\delta_{j,t}}{\Delta_{j,t}} \mathbf{x}_{j,1} \right] \\ &= \sum_{j \in [N]} \mathbf{x}_{j,t} + \underbrace{\sum_{j \in [N]} \mathbf{x}_{j,t} \eta_{j,t-1} \bar{\ell}_t - \sum_{j \in [N]} \mathbf{x}_{j,t} \eta_{j,t-1} \ell_{j,t}}_{=0 \text{ by Eq. (10)}} + \sum_{j \in [N]} \frac{\delta_{j,t}}{\Delta_{j,t}} \mathbf{x}_{j,1} \\ &\leq \sum_{j \in [N]} \mathbf{x}_{j,t} + \sum_{j \in [N]} \frac{\delta_{j,t}}{\Delta_{j,t}} \mathbf{x}_{j,1}. \end{aligned} \tag{15}$$

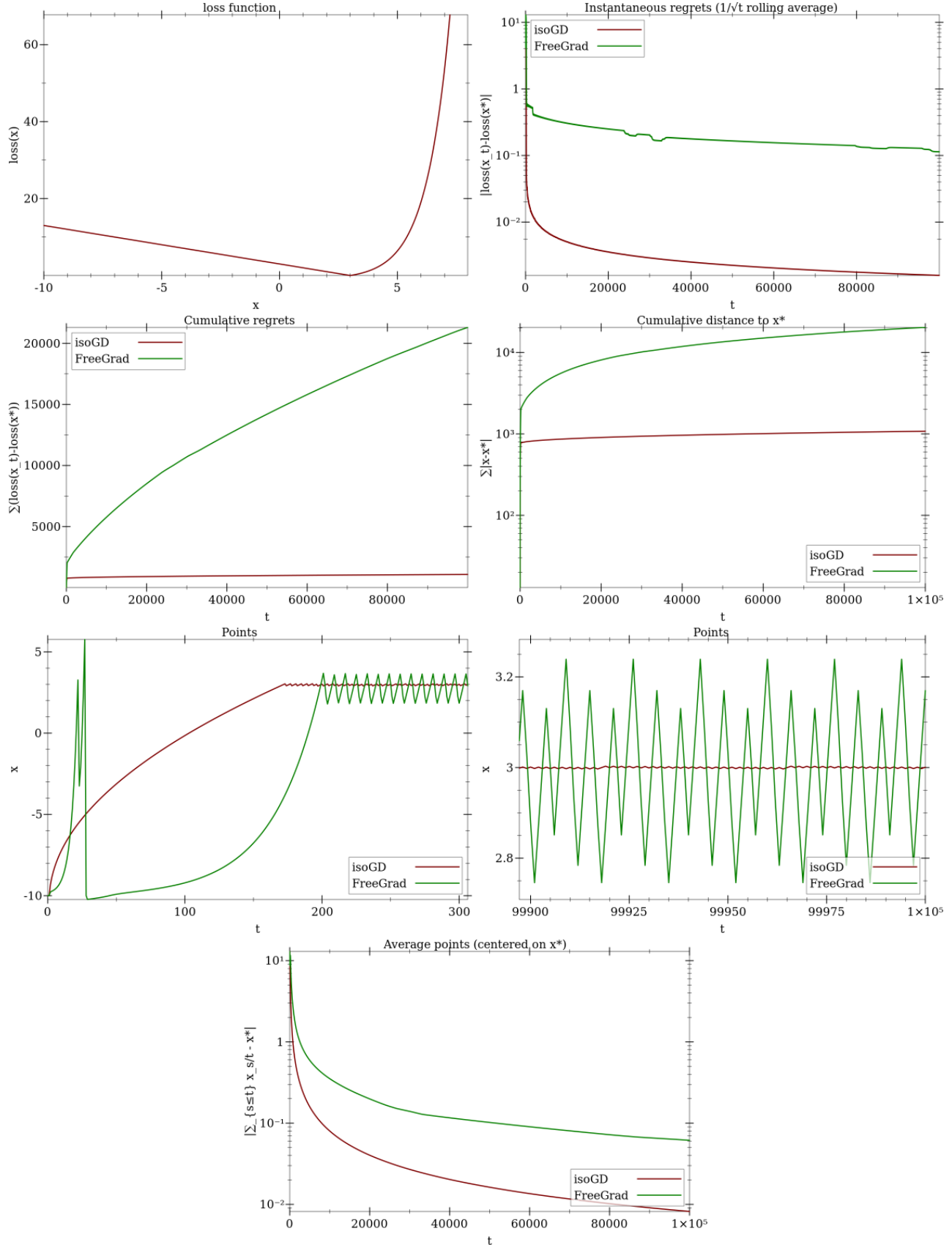


Figure 2: Behaviour of isoGD ( $q = 1$ ) and FreeRange on performing gradient descent on  $\ell(\mathbf{x}) = \mathbf{x}^* - \mathbf{x}$  if  $\mathbf{x} < \mathbf{x}^*$  and  $\ell(\mathbf{x}) = \exp(\mathbf{x} - \mathbf{x}^*) - 1$  otherwise. Note the log scales on some of the y-axis.

Observe that Eq. (15) holds straightforwardly also for  $t \in \mathcal{T}_\emptyset$  since  $\mathbf{x}_{t'} = \mathbf{x}_t$ —this would not be the case if null updates were not performed on all coordinates at once. Hence, by summing over  $t \in [T]$  and telescoping we have, using  $\mathbf{x}_{j,1} = 1$ ,

$$\sum_{j \in [N]} \mathbf{x}_{j,T+1} \leq N + \sum_{j \in [N]} \sum_{t \in [T]} \frac{\delta_{j,t}}{\Delta_{j,t}}.$$

Note that if  $\Delta_{j,t} = 0$ , necessarily by Eq. (3) we have  $\mathbf{x}_{t+1} = \mathbf{x}_{j,t'} = \mathbf{x}_{j,1}$ , so we can choose  $\delta_{j,t}/\Delta_{j,t-1} = 0$ . Define  $C_{j,T} = \sum_{t \in [T]} \frac{\delta_{j,t}}{\Delta_{j,t}}$ , then

$$\mathbf{x}_{i,T+1} \leq \sum_{j \in [N]} \mathbf{x}_{j,T+1} \leq N + N \max_{j \in [N]} C_{j,T}.$$

**Bounding  $\Delta_{i,T}$ .** Recall that  $\tau = \max \mathcal{T}_\emptyset$  is the last step where a null update is performed, and note that  $\tau \geq 1$  when  $T \geq 1$  unless the losses are 0. Let  $\mu \in [N]$  be such that  $\eta_{\mu,\tau-1} |\bar{\ell}_t - \ell_{\mu,t}| > 1/2$  ( $\mu$  necessarily exists by definition of  $\mathcal{T}_\emptyset$ ). This implies that  $\eta_{\mu,\tau-1}^{-1} < 2S$ , and thus  $\Delta_{\mu,\tau-1} < 2S \ln N$ . Therefore, since null updates are performed on all coordinates at the same time with  $\delta_{\cdot,t} = s_t$  for  $t \in \mathcal{T}_\emptyset$ , for all  $i \in [N]$  we have

$$\sum_{t \in \mathcal{T}_\emptyset} \delta_{i,t} = \sum_{t \in \mathcal{T}_\emptyset} \delta_{\mu,t} \leq \Delta_{\mu,\tau} = \Delta_{\mu,\tau-1} + \delta_{\mu,\tau} \leq 2S \ln N + S. \quad (16)$$

Now define

$$\forall t \notin \mathcal{T}_\emptyset : \quad \forall y > 0 : \hat{\delta}_t(y) = \frac{qr_{i,t}^2}{2y}, \quad \forall y \leq 0 : \hat{\delta}_t(y) = \infty, \quad \forall t \in \mathcal{T}_\emptyset : \hat{\delta}_t(\cdot) = 0.$$

Also define  $A_t = \sum_{u \in \mathcal{T}_\emptyset \cap [t]} \delta_{i,s}$ . Then by Lemma 46, for  $t \notin \mathcal{T}_\emptyset$ ,

$$\delta_t \leq \hat{\delta}_t(\Delta_{t-1} - q|r_{i,t}|) = \hat{\delta}_t((\Delta_t - A_t) - \delta_t - q|r_{i,t}| + A_t) \leq \hat{\delta}_t((\Delta_t - A_t) - \delta_t - q|r_{i,t}|).$$

Define  $\tilde{\Delta}_t = \Delta_t - A_t$ , then for all  $t \notin \mathcal{T}_\emptyset$ ,  $\tilde{\Delta}_t = \tilde{\Delta}_{t-1} + \delta_t$  with  $\delta_t \leq \hat{\delta}_t(\tilde{\Delta}_t - \delta_t - q|r_{i,t}|)$  and for  $t \in \mathcal{T}_\emptyset$ ,  $\tilde{\Delta}_t = \tilde{\Delta}_{t-1} + 0 = \tilde{\Delta}_t + \hat{\delta}_t(\tilde{\Delta}_t)$ . Therefore, by Lemma 13, with the isotuning sequence  $(\hat{\Delta}, \hat{\delta})$ , and using Lemma 10 on  $\hat{\Delta}_T$ ,

$$\begin{aligned} \tilde{\Delta}_T &\leq \hat{\Delta}_T + \max_{t \in [T]} \{\delta_t + q|r_{i,t}|\}, \\ \Delta_T &\leq \sqrt{q \sum_{t \notin \mathcal{T}_\emptyset} r_{i,t}^2} + \max_{t \notin \mathcal{T}_\emptyset} \{\delta_t + q|r_{i,t}|\} + A_T \leq \sqrt{V_{i,T} \ln N} + S/5 + S \ln N + 2S \ln N + S \end{aligned}$$

where  $A_T$  is bounded in Eq. (16), and  $\delta_t = r_{i,t} (1 - \frac{1}{u} \ln(1+u))$  with  $u = \eta_{i,t-1} r_{i,t} \leq 1/2$  can be bounded by  $r_{i,t}/5$ .

**Finally.** We can now put it all together:

$$\begin{aligned} \mathcal{R}_{i,T} &\leq \left(1 + \frac{\ln(N + N \max_{j \in [N]} C_{j,T})}{\ln N}\right) \left(\sqrt{V_{i,T} \ln N} + 3S \ln N + \frac{6}{5}S\right) \\ &\leq \left(2 + \frac{\ln(1 + \max_{j \in [N]} C_{j,T})}{\ln N}\right) \left(\sqrt{V_{i,T} \ln N} + S\left(\frac{6}{5} + 3 \ln N\right)\right). \end{aligned}$$

Let  $t_0 = \min\{t \in [T] : \Delta_{j,t} > 0\}$ , then for all  $\theta \in \{t_0, \dots, T\}$ ,

$$C_{j,T} = \sum_{t \in [T]} \left(1 - \frac{\Delta_{j,t-1}}{\Delta_{j,t}}\right) = \sum_{t=t_0+1}^T \left(1 - \frac{\Delta_{j,t-1}}{\Delta_{j,t}}\right) = \sum_{t=t_0+1}^{\theta} \underbrace{\left(1 - \frac{\Delta_{j,t-1}}{\Delta_{j,t}}\right)}_{\leq 1} + \sum_{t=\theta}^T \ln \frac{\Delta_{j,t}}{\Delta_{j,t-1}} \leq \theta - t_0 + \ln \frac{\Delta_{j,T}}{\Delta_{j,\theta}}$$



where we used  $1 - 1/x \leq \ln x$  and telescoping. Hence,

$$C_{j,T} \leq \min_{\theta \in \{t_0, \dots, T\}} \theta - t_0 + \ln \frac{\Delta_{j,T}}{\Delta_{j,\theta}}.$$

From this we can deduce all the following simultaneously:

$$\begin{aligned} C_{j,T} &\leq T, & \text{with } \theta &= T, \\ C_{j,T} &\leq o(1) + \ln ST & \text{with } \theta &= \operatorname{argmin}_{t \in [T]} \left\{ \Delta_{j,t} \geq \frac{\Delta_{j,T}}{ST} \right\} \text{ since } \Delta_{j,T} = O(\sqrt{ST} \ln T), \\ & & \text{then } \frac{\Delta_{j,T}}{ST} &= o(1), \text{ and thus } \theta = t_0 + o(1), \\ C_{j,T} &\leq \tau + \ln \frac{ST}{s_\tau} & \text{with } \theta &= \tau = \max \mathcal{T}_\theta \text{ since } \Delta_{j,\tau} \geq \delta_{j,\tau} = s_\tau \\ & & \text{and } \Delta_{j,T} &\leq ST \text{ since } \delta_{j,t} \leq S \text{ for all } t \in [T]. \quad \square \end{aligned}$$

**Remark 44.** With a slight modification of Algorithm 4, we can obtain a variant of the Bernstein Optimal Aggregation (BOA) algorithm (Wintenberger, 2016) (see Footnote 5 in main text) with online correction and null updates and prove the same bound as for isoML-Prod under the same constraints. Replace Eq. (11) with

$$\forall i \in [N] : \mathbf{x}_{i,t'} = \mathbf{x}_{i,t} f(\eta_{i,t-1} r_{i,t}) \quad \text{with } f(y) = \exp\left(y - \frac{y^2/2}{1 - |y|}\right).$$

Recall that  $\phi_i(\mathbf{x}) = -\ln \mathbf{x}_i$  and  $\mathcal{X} = [0, \infty)^N$ . To satisfy Eq. (5) we take

$$\delta_{i,t} = \delta_{i,t}^* = \left( \frac{1}{\eta_{i,t-1}} (\phi_{i,t'} - \phi_{i,t}) + r_{i,t} \right)_+ = \eta_{i,t-1} \frac{r_{i,t}^2/2}{1 - \eta_{i,t-1} |r_{i,t}|}.$$

We also need to show that Eq. (15) still holds: We do this by reducing it to the isoML-Prod case by using  $f(y) \leq 1 + y$  for  $|y| < 1$  (from Lemma 46) with  $y = \eta_{i,t-1} r_{i,t}$ ; though as for isoML-Prod we enforce  $|y| \leq 1/2$  to bound  $\max_t \delta_t \leq S$  (which is slightly worse than  $S/5$  for isoML-Prod). This constraint means that we need to perform null updates in the same conditions as for isoML-Prod, and we also still need null updates to be performed on all coordinates at the same time. Therefore the bound of Theorem 24 holds also for this variant of BOA. Note that Lemma 46 also saves a factor  $\sqrt{2}$  compared to prior work where  $\exp(y - y^2)$  is used instead of  $\exp(y - y^2/(2(1 - |y|)))$ .  $\diamond$

**Remark 45.** It is straightforward to adapt the proof to allow for different initial weights  $\mathbf{x}_{i,1}$  that may depend on  $i$ , as well as extending to a growing number of experts (Mourtada and Maillard, 2017). For example, if we can ensure that  $\sum_{i \in [N]} \mathbf{x}_{i,1} \leq 1$  (and still  $\mathbf{x}_{i,1} \geq 0$ ), then we would have  $\phi_{i,1} = \ln \frac{1}{\mathbf{x}_{i,1}}$  (instead of 0 in Eq. (12)) but from Eq. (15) we have by telescoping

$$\mathbf{x}_{i,T+1} \leq \sum_{j \in [N]} \mathbf{x}_{j,1} + \sum_{j \in [N]} \mathbf{x}_{j,1} C_{j,T} \leq 1 + \langle \mathbf{x}_1, C_{\cdot,T} \rangle$$

and setting  $q_i = \ln \frac{1}{\mathbf{x}_{i,1}} = \phi_{i,1}$  we obtain the regret bound, using  $\mathcal{R}_T(\mathbf{x}^*) = \langle \mathbf{x}^*, (\mathcal{R}_{i,T})_{i \in [N]} \rangle$  on the simplex,

$$\begin{aligned} \mathcal{R}_T(\mathbf{x}^*) &\leq \sum_i \mathbf{x}_i^* \left( 1 + \frac{\phi_{i,1}}{q_i} + \frac{\ln(1 + \langle \mathbf{x}_1, C_{\cdot,T} \rangle)}{q_i} \right) \left( \sqrt{q_i V_{i,T}} + 3S q_i + 2S \right) \\ &\leq \left( 2 + \frac{\ln(1 + \langle \mathbf{x}_1, C_{\cdot,T} \rangle)}{\min_i \ln \frac{1}{\mathbf{x}_{i,1}}} \right) \left( \sqrt{\mathbb{H}(\mathbf{x}^*, \mathbf{x}_1) \sum_i \mathbf{x}_i^* V_{i,T}} + 3S \mathbb{H}(\mathbf{x}^*, \mathbf{x}_1) + 2S \right), \end{aligned}$$

where  $\mathbb{H}(\mathbf{x}^*, \mathbf{x}_1) = \sum_i \mathbf{x}_i^* \ln \frac{1}{\mathbf{x}_{i,1}}$  is the cross-entropy of  $\mathbf{x}^*$  and  $\mathbf{x}_1$  (and is equal to the relative entropy  $\operatorname{RE}(\mathbf{x}^* \parallel \mathbf{x}_1)$  plus the constant entropy term  $\mathbb{H}(\mathbf{x}^*, \mathbf{x}^*)$ ), and we used Cauchy-Schwarz on  $\sum_i \mathbf{x}_i^* \sqrt{q_i V_{i,T}} = \sum_i \sqrt{\mathbf{x}_i^* q_i} \sqrt{\mathbf{x}_i^* V_{i,T}} \leq \sqrt{\sum_i \mathbf{x}_i^* q_i} \sqrt{\sum_i \mathbf{x}_i^* V_{i,T}}$ . Finally, note that  $C_{j,T}$  is independent of  $q_j$ . We recover exactly the bound of Theorem 24 when  $\mathbf{x}_{i,1} = 1/N$ .  $\diamond$

**Lemma 46.** For all  $x \in (-1, 1)$ ,

$$\ln(1+x) \geq x - \frac{x^2}{2(1-|x|)}. \quad \diamond$$

*Proof.* For  $x \in [0, 1)$ , define  $g_+(x) = \frac{1}{2}x^2 + (1-x)(\ln(1+x)-x)$ . Then  $g'_+(x) = x - (\ln(1+x)-x) + \frac{1-x}{1+x} - (1-x) \geq 0$ , hence  $g_+$  is increasing on  $[0, 1)$  and since  $g_+(0) = 0$  then  $g_+(x) \geq 0$  for all  $x \in [0, 1)$ .

Similarly, for  $x \in (-1, 0]$ , define  $g_-(x) = \frac{1}{2}x^2 + (1+x)(\ln(1+x)-x)$ . Then  $g'_-(x) = x + (\ln(1+x)-x) + 1 - (1+x) = \ln(1+x) - x \leq 0$  and since  $g_-(0) = 0$  then  $g_-(x) \geq 0$  for all  $x \in (-1, 0]$ , which concludes the proof.  $\square$

## E ADAHEDGE, MIRROR-DESCENT STYLE

As mentioned before, the present paper is building upon the balancing of the regret idea of de Rooij et al. (2014) for AdaHedge. We show how to build a variant of AdaHedge that uses Mirror Descent-style projection with online correction based on Algorithm 1 without null updates, and straightforwardly recover the same regret bound with our tools, with a slight generalization.

We assume  $\mathcal{X} = \Delta_p(N)$ . Recall that the loss of the algorithm is  $\bar{\ell}_t = \sum_{i \in [N]} \mathbf{x}_{i,t} \ell_{i,t}$  with  $\ell_t \in \mathbb{R}^N$  and the instantaneous regret is  $r_t = \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle$ . Define, for some unknown  $T \in \mathbb{N}_1$ ,

$$\begin{aligned} L &= \max_{t \in [T], i \in [N]} |\ell_{i,t}|, & S &= \max_{t \in [T], i \in [N]} |\bar{\ell}_t - \ell_{i,t}|, \\ u_t &= \sum_i \mathbf{x}_{i,t} (\ell_{i,t} - m_t)^2, & U_T &= \sum_{t \in [T]} u_t, & v'_t &= \sum_i \mathbf{x}_{i,t} (\bar{\ell}_t - \ell_{i,t})^2, & V'_T &= \sum_{t \in [T]} v'_t. \end{aligned}$$

de Rooij et al. (2014) prove the scale-free bound of AdaHedge by first analyzing the regret for  $[0, 1]$  losses and then showing that the algorithm outputs predictions that are invariant to any rescaling of the losses.

**Theorem 47** (Theorem 6, de Rooij et al. (2014), rescaled). The regret of AdaHedge is bounded by

$$\mathcal{R}_{i,T} \leq 2\sqrt{V'_T \ln N} + S' \left(2 + \frac{4}{3} \ln N\right).$$

where  $S' = \max_{t \in [T]} (\max_{i \in [N]} \ell_{i,t} - \min_{j \in [N]} \ell_{j,t})$ .  $\diamond$

This regret bound has the nice property of being invariant to any translation of the losses. However, surprisingly, it turns out that on the simplex non-translation invariant bounds have better properties since they can always be made translation invariant, while the converse may not hold:

**Lemma 48** (Loss translation on the simplex). Let  $\mathcal{X} = \Delta_p(N)$ ,  $\ell_t \in \mathbb{R}^N$ ,  $\forall t \in \mathbb{N}_1$ . For all  $\mathbf{x}^* \in \mathcal{X}$ , if an algorithm  $\mathcal{A}$  sequentially predicts  $\mathbf{x}_t$  and has a regret of the form

$$\mathcal{R}_T(\mathbf{x}^*) \leq f(\ell_1, \ell_2, \dots)$$

then for all sequentially available  $m_t \in \mathbb{R}$ , algorithm  $\mathcal{A}$  fed with the translated losses  $\tilde{\ell}_t = \ell_t - m_t$  achieves the regret

$$\mathcal{R}_T(\mathbf{x}^*) = \sum_t \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle \leq f(\ell_1 - m_1, \ell_2 - m_2, \dots). \quad \diamond$$

*Proof.* Simply observe that on the simplex the regret is translation invariant:  $\langle \mathbf{x}_t - \mathbf{x}^*, \ell_t - m_t \rangle = \langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle$ .  $\square$

In particular when  $m_t = \bar{\ell}_t$  the regret bound becomes translation invariant. See also Chen et al. (2021) for a coordinate-wise generalization of this property and additional discussion of its benefits.

We now describe the isoHedge algorithm. We use Assumption 1, and we take and  $\phi(\mathbf{x}) = \text{RE}(\mathbf{x}^* \parallel \mathbf{x})$ . Define the update rule, for all  $i \in [N]$  and for all  $t \in \mathbb{N}_1$ :

$$\begin{aligned} \mathbf{x}_{i,1} &= \frac{1}{N}, \\ \mathbf{x}_{i,t'} &= \mathbf{x}_{i,t} \frac{\exp(-\eta_{t-1} \ell_{i,t})}{\sum_{j \in [N]} \mathbf{x}_{j,t} \exp(-\eta_{t-1} \ell_{j,t})} \quad (\text{AdaHedge update and 'projection', MD style}) \end{aligned}$$

and we also use the online correction of Eq. (3). Null updates are unnecessary in this case because the update is always valid and  $\delta_t$  will always be bounded by the largest loss. Define  $q = \ln N$ , and we define  $\delta_t$  as tightly as possible from Eq. (5):

$$\delta_t = \delta_t^* = \left( \frac{1}{\eta_{t-1}} \ln \frac{\mathbf{x}_{i,t}}{\mathbf{x}_{i,t'}} + r_t \right)_+ = \frac{1}{\eta_{t-1}} \ln \left( \sum_{j \in [N]} \mathbf{x}_{j,t} \exp(-\eta_{t-1} \ell_{j,t}) \right) + \bar{\ell}_t. \quad (17)$$

Note that the definition of  $\delta_t$  is independent of the coordinate  $i$ . See Algorithm 5.

---

**Algorithm 5** Instance of Algorithm 1 for AdaHedge with Mirror-Descent-style projection, online correction and isotuning (no null updates). Note that if  $\Delta_{t-1} = 0$ , the limit  $\eta_{t-1}^{-1} \rightarrow 0$  gives  $\delta_t = \max_{i \in [N]} (\bar{\ell}_t - \ell_{i,t})$ , and the online correction implies that  $\mathbf{x}_{t+1} = \mathbf{x}_1$ .

---

**def** isoHedge():

$$\mathbf{x}_1 = \left( \frac{1}{N}, \dots, \frac{1}{N} \right)$$

$$\Delta_0 = 0$$

**for**  $t = 1, 2, \dots$ :

**predict**  $\mathbf{x}_t$  ; **observe**  $\ell_t$

**choose**  $m_t \in \mathbb{R}$  # *can be  $\bar{\ell}_t$*

$$\eta = q/\Delta_{t-1} \text{ # } \eta_{t-1}$$

$$Z = \sum_{j \in [N]} \mathbf{x}_{j,t} \exp(\eta(m_t - \ell_{j,t}))$$

**for**  $i \in [N]$ :

$$\mathbf{x}_{i,t'} = \frac{1}{Z} \mathbf{x}_{i,t} \exp(\eta(m_t - \ell_{i,t})) \text{ # AdaHedge update, MD style, projection = normalization}$$

$$\delta_t = \frac{1}{\eta} \ln Z + \bar{\ell}_t \text{ # AdaHedge's mixability gap}$$

$$\Delta_t = \Delta_{t-1} + \delta_t \text{ # isotuning}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t'} \frac{\Delta_{t-1}}{\Delta_t} + \frac{\delta_t}{\Delta_t} \mathbf{x}_1 \text{ # online correction}$$


---

**Theorem 49.** For all sequentially chosen  $\{m_t\}_t \in \mathbb{R}^T$  the regret of isoHedge in Algorithm 5 after  $T$  rounds is at most

$$\mathcal{R}_T(\mathbf{x}^*) \leq 2\sqrt{U_t \ln N} + 2S + \frac{2}{3}L \ln N.$$

In particular with  $m_t = \bar{\ell}_t$ , the regret is bounded by

$$\mathcal{R}_T(\mathbf{x}^*) \leq 2\sqrt{V'_t \ln N} + 2S + \frac{2}{3}S \ln N. \quad \diamond$$

*Proof.* We prove for  $m_t = 0, \forall t$  and apply Lemma 48.

As usual, we need to express  $\delta_t$  as a function of  $\Delta_t$ . Define  $\hat{\delta}_t(x) = qu_t/(2x)$  for  $x > 0$  and  $\hat{\delta}_t(x) = \infty$  otherwise. From Corollary 52, we know that  $e^x \leq 1 + x + \frac{x^2/2}{1-x/3}$  for  $x < 3$  and so, with  $\ln(1+x) \leq x$ ,

$$\begin{aligned} \delta_t &\leq \frac{1}{\eta_{t-1}} \ln \left( \sum_j \mathbf{x}_{j,t} \left( 1 - \ell_{j,t} + \frac{\eta_{t-1}^2 \ell_{j,t}^2 / 2}{1 + \eta_{t-1} \ell_{j,t} / 3} \right) \right) + \bar{\ell}_t \\ &\leq \frac{\eta_{t-1} u_t / 2}{1 - \eta_{t-1} L / 3} = \frac{qu_t / 2}{\Delta_{t-1} - qL / 3} = \frac{qu_t / 2}{\Delta_t - \delta_t - qL / 3} \end{aligned}$$

which is valid at least when  $\Delta_t - \delta_t - qL/3 > 0$ , and thus  $\delta_t \leq \hat{\delta}_t(\Delta_t - \delta_t - qL/3)$  for all  $t \in [T]$ . Therefore, using Lemma 13, let  $(\hat{\Delta}, \hat{\delta})$  be an isotuning sequence, then

$$\Delta_T \leq \hat{\Delta}_T + \max_{t \in [T]} \{\delta_t + qL/3\} \leq \sqrt{q \sum_{t \in [T]} u_t} + S + qL/3$$

by Lemma 10 and using  $\delta_t \leq S$  from Eq. (17). The result follows from Lemma 12, since with  $\phi_1 = \ln N = q$  we have  $\mathcal{R}_{i,T} \leq 2\Delta_T$ .  $\square$

Observe that by contrast to the original analysis of AdaHedge, our analysis applies directly to losses in  $\mathbb{R}^N$ . Furthermore, the only application-specific difficulty is to bound  $\delta_t$ .

**Remark 50.** One difference between isoHedge (when  $m_t = \bar{\ell}_t$ ) and AdaHedge is that for  $t = 2$  AdaHedge predicts at a corner of the probability simplex like Follow The Leader (FTL) (de Rooij et al., 2014), while isoHedge predicts at the ‘center’  $(1/N, 1/N, \dots)$ . This latter behaviour is also the choice made by SOLO FTRL (Orabona and Pál, 2018), which deals with both bounded and unbounded domains—where playing at a corner is not always possible or even desirable. In particular, when losses are gradients, playing at a corner may lead to unbounded or infinite losses, e.g.,  $\ell_{i,t} = \frac{\partial(-\log(\mathbf{x}_t, f_t))}{\partial \mathbf{x}_{i,t}}$  for some  $f_t$ .  $\diamond$

The following results are technical lemmas used in the proof of Theorem 49. A similar result can be extracted from the proofs of de Rooij et al. (2014).

**Lemma 51.** For all  $x \in \mathbb{R}$ ,  $(e^x - x - 1)(1 - x/3) \leq x^2/2$ .  $\diamond$

*Proof.* Let

$$\begin{aligned} f(x) &= (e^x - x - 1)(1 - x/3) - x^2/2 \\ &= e^x(3 - x)/3 + (x + 1)(x/3 - 1) - x^2/2, \\ f'(x) &= e^x(2 - x)/3 - x/3 - 2/3, \\ f''(x) &= e^x(1 - x)/3 - 1/3, \\ f'''(x) &= -xe^x/3. \end{aligned}$$

Notice that  $f(0) = f'(0) = f''(0) = 0$  and all functions above are continuous on  $\mathbb{R}$ . Therefore a Taylor expansion at 0 with Lagrange remainder gives  $f(x) = -x^3ye^y/3$  for some  $y$  between  $x$  and 0 and thus  $f(x) \leq 0$  for all  $x \in \mathbb{R}$ , which proves the result.  $\square$

**Corollary 52.** For all  $x < 3$ ,  $e^x \leq 1 + x + \frac{x^2/2}{1-x/3}$ .  $\diamond$

*Proof.* Follows from Lemma 51.  $\square$

## F SOFT-BAYES REVISITED

Soft-Bayes (Orseau et al., 2017) is an algorithm for the universal portfolio problem (Cover, 1991), with running time  $O(N)$  per step for  $N$  experts and a regret bound of  $O(\sqrt{NT \ln N})$ . A variant is also given with a better bound of  $O(\sqrt{mT \ln N})$  where  $m$  is the number of experts in hindsight that have been ‘good’ at least once after  $T$  steps. While not changing the computational complexity of the algorithm, this variant requires keeping track of these  $m$  experts online with an array of size  $N$ , thus making the algorithm less simple and makes the bound dependent on the specifics of the tracking algorithm.

We show how isotuning can be used to obtain a regret bound of  $O(\sqrt{mT \ln N})$  *without* having to keep track of  $m$ , just by using the prescribed isotuning learning rate. We do not need null updates as we can bound  $\max_t \delta_t$  directly.

We are in the online convex optimization setting, where  $\mathcal{X}$  is the probability simplex of  $N$  coordinates, and  $\ell_t(\mathbf{x}) = -\ln\langle \mathbf{x}, p_t \rangle$  where  $p_t \in [0, \infty)^N$ .

Let  $\mathcal{N}_T \in \operatorname{argmin}_{S \subseteq [N]} \{|S| \mid \forall t \in [T] : S \cap \operatorname{argmax}_{i \in [N]} \{p_{i,t}\} \neq \emptyset\}$  be the smallest subset of  $[N]$  such that at each step  $t$  at least one of the best experts at step  $t$  is in this subset. Ties are broken arbitrarily. Note that neither  $\mathcal{N}_T$  nor  $|\mathcal{N}_T|$  need to be tracked by the algorithm, by contrast to the original work, which allows us to compete against a better subset in hindsight.

The core update of the Soft-Bayes algorithm is as follows (note that the *notation* for the index of the learning rate is offset by one compared to the notation of the original paper, to be aligned with the notation of the current paper). For all  $i \in [N]$  and  $t \in [T]$ ,

$$\begin{aligned} \mathbf{x}_{i,1} &= \frac{1}{N}, \\ \bar{p}_t &= \langle \mathbf{x}_t, p_t \rangle, && \text{(mixture prediction)} \\ \mathbf{x}_{i,t'} &= \mathbf{x}_{i,t} \left( 1 - \eta_{t-1} + \eta_{t-1} \frac{p_{i,t}}{\bar{p}_t} \right), && \text{(Soft-Bayes update)} \end{aligned} \tag{18}$$

The online correction of Eq. (3) is also applied (as in the original work), but not null updates. See Algorithm 6, and note that by contrast to the original algorithm, it does not indeed require keeping track of  $m$ .

---

**Algorithm 6** The Soft-Bayes algorithm with isotuning with regret  $O(\sqrt{mT \ln N})$  where  $m$  is the number of ‘good’ experts. Null updates are not used as the additive term  $\max_t \delta_t$  is well bounded.

---

**def** isoSoft-Bayes( $N \geq 2$ ):

```

 $\Delta_0 = 2 \ln N$ 
 $\mathbf{x}_1 = \{\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\}$ 
for  $t = 1, 2, \dots$ :
  predict  $\mathbf{x}_t$ ; observe  $p_t$  # loss is  $-\ln \bar{p}_t$ 
   $\bar{p}_t = \langle \mathbf{x}_t, p_t \rangle$ 
   $\eta = \frac{\ln N}{\Delta_{t-1}}$  #  $\eta_{t-1}$ 
   $\forall i \in [N] : \mathbf{x}_{i,t'} = \mathbf{x}_{i,t} \left(1 - \eta + \eta \frac{p_{i,t}}{\bar{p}_t}\right)$  # Soft-Bayes update
   $\delta_t = \max_{i \in [N]} \ln \left(1 + \frac{\eta}{1-\eta} \frac{p_{i,t}}{\bar{p}_t}\right)$ 
   $\Delta_t = \Delta_{t-1} + \delta_t$  # isotuning
   $\mathbf{x}_{t+1} = \mathbf{x}_{t'} \frac{\Delta_{t-1}}{\Delta_t} + \frac{\delta_t}{\Delta_t} \mathbf{x}_1$  # online correction

```

---

First we recall a central lemma of the original paper:

**Lemma 53** (Lemma 4, Orseau et al. (2017)). Let  $\mathbf{x} \in \mathcal{X}$  and  $b \in [0, \infty)^N$ , then for all  $\eta \in (0, 1)$ ,

$$\ln \sum_{i \in [N]} \mathbf{x}_i b_i \leq \frac{1}{\eta} \sum_{i \in [N]} \mathbf{x}_i \ln(1 - \eta + \eta b_i) + \max_{i \in [N]} \ln \left(1 + \frac{\eta}{1-\eta} b_i\right). \quad \diamond$$

Define for all  $t \in [T]$  (note that this slightly differs from Assumption 1),

$$q = \ln N, \quad \Delta_t = \frac{q}{\eta_t}, \quad \Delta_0 = 2q, \quad \Delta_t = \Delta_{t-1} + \delta_t.$$

**Theorem 54.** The regret of isoSoft-Bayes (Algorithm 6) for all  $\mathbf{x}^* \in \mathcal{X}$  is bounded for all  $T \in \mathbb{N}_1$  by

$$\mathcal{R}_T(\mathbf{x}^*) \leq 4\sqrt{mT \ln N} + 3 \ln O(NT) + 10m \ln \frac{N}{m} + 6 \ln N. \quad \diamond$$

*Proof.* Define  $\phi(\mathbf{x}) = \sum_{i \in [N]} \mathbf{x}_i \ln \frac{\mathbf{x}_i^*}{\mathbf{x}_i}$  and so we take  $q = \ln N \geq \phi_1$ . To satisfy Eq. (5),

$$\begin{aligned} \delta_t^* &= \left( \frac{1}{\eta_{t-1}} \sum_{i \in [N]} \mathbf{x}_i^* \ln \frac{\mathbf{x}_{i,t'}}{\mathbf{x}_{i,t}} + \ln \frac{\langle \mathbf{x}^*, p_t \rangle}{\bar{p}_t} \right)_+ \\ &= \left( -\frac{1}{\eta_{t-1}} \sum_{i \in [N]} \mathbf{x}_i^* \ln \left(1 - \eta_{t-1} + \eta_{t-1} \frac{p_{i,t}}{\bar{p}_t}\right) + \ln \sum_{i \in [N]} \mathbf{x}_i^* \frac{p_{i,t}}{\bar{p}_t} \right)_+ \\ &\leq \max_{i \in [N]} \ln \left(1 + \frac{\eta_{t-1}}{1 - \eta_{t-1}} \frac{p_{i,t}}{\bar{p}_t}\right) =: \delta_t \end{aligned}$$

where we used Lemma 53 on the last line. Hence this definition of  $\delta_t$  satisfies Eq. (5).

From Lemma 11 and the concavity of the log function, for all  $i \in [N]$  we have, using the update rule Eq. (18)

on the second line along with  $\ln(1 + \eta - \eta x) = \ln(1 + \frac{\eta}{1-\eta}x) + \ln(1 - \eta)$ :

$$\begin{aligned}
& \Delta_t \ln \mathbf{x}_{i,t+1} \geq \Delta_{t-1} \ln \mathbf{x}_{i,t'} + \delta_t \ln \mathbf{x}_{i,1} \\
\Leftrightarrow & \Delta_{t-1} \ln \mathbf{x}_{i,t+1} + \delta_t \ln \mathbf{x}_{i,t+1} \geq \Delta_{t-1} \ln \mathbf{x}_{i,t} + \Delta_{t-1} \ln \left( 1 + \frac{\eta_{t-1}}{1-\eta_{t-1}} \frac{p_{i,t}}{\bar{p}_t} \right) + \Delta_{t-1} \ln(1 - \eta_{t-1}) + \delta_t \ln \mathbf{x}_{i,1} \\
\Leftrightarrow & \ln \left( 1 + \frac{\eta_{t-1}}{1-\eta_{t-1}} \frac{p_{i,t}}{\bar{p}_t} \right) \leq \ln \frac{\mathbf{x}_{i,t+1}}{\mathbf{x}_{i,t}} + \frac{\delta_t}{\Delta_{t-1}} \ln \frac{\mathbf{x}_{i,t+1}}{\mathbf{x}_{i,1}} \underbrace{- \ln(1 - \eta_{t-1})}_{\leq \frac{\eta_{t-1}}{1-\eta_{t-1}} = \frac{q}{\Delta_{t-1}-q}}, \\
\delta_t \leq & \sum_{i \in \mathcal{N}_T} \ln \left( 1 + \frac{\eta_{t-1}}{1-\eta_{t-1}} \frac{p_{i,t}}{\bar{p}_t} \right) \leq \sum_{i \in \mathcal{N}_T} \ln \frac{\mathbf{x}_{i,t+1}}{\mathbf{x}_{i,t}} + \frac{\delta_t}{\Delta_{t-1}} \underbrace{\sum_{i \in \mathcal{N}_T} \ln \frac{\mathbf{x}_{i,t+1}}{\mathbf{x}_{i,1}}}_{\leq m \ln \frac{N}{m}} + m \frac{q}{\Delta_{t-1} - q} \tag{19}
\end{aligned}$$

Define  $a = m \ln \frac{N}{m}$ , and  $b_t = \sum_{i \in \mathcal{N}_T} \ln \frac{\mathbf{x}_{i,t+1}}{\mathbf{x}_{i,t}}$  and we write  $b_{1:T} = \sum_{s \in [T]} b_s$ . Then when  $1 - a/\Delta_{t-1} > 0$  we have

$$\delta_t \left( 1 - \frac{a}{\Delta_{t-1}} \right) \leq b_t + \frac{mq}{\Delta_{t-1} - q}, \quad \Leftrightarrow \quad \delta_t \leq \left( 1 + \frac{a}{\Delta_{t-1} - a} \right) \left( b_t + \frac{mq}{\Delta_{t-1} - q} \right).$$

Define, for all  $t \in [T]$ ,

$$\forall x > 0 : \hat{\delta}_t(x) = \left( 1 + \frac{a}{x} \right) \left( b_t + \frac{mq}{x} \right), \quad \text{and } \forall x \leq 0 : \hat{\delta}_t(x) = \infty.$$

Then, for all  $t \in [T]$ , since  $\hat{\delta}_t(\Delta_{t-1} - a) = \infty$  if  $1 - a/\Delta_{t-1} \leq 0$ ,

$$\delta_t \leq \hat{\delta}_t(\Delta_{t-1} - \max\{a, q\}) \leq \hat{\delta}_t((\Delta_t - \Delta_0) - (\delta_t + \max\{a, q\} - \Delta_0)_+),$$

and observe that  $(\Delta_t - \Delta_0) = (\Delta_{t-1} - \Delta_0) + \delta_t$ . Let  $(\hat{\Delta}, \hat{\delta})$  be an isotuning sequence, then by Lemma 13,

$$\begin{aligned}
\Delta_T - \Delta_0 & \leq \hat{\Delta}_T + \max_{t \in [T]} (\delta_t + \max\{a, q\} - \Delta_0)_+, \\
\Delta_T & \leq \hat{\Delta}_T + \max_{t \in [T]} \max\{\delta_t + \max\{a, q\}, \Delta_0\} \leq \hat{\Delta}_T + \max_{t \in [T]} \delta_t + a + 2q.
\end{aligned}$$

By Theorem 7 we have

$$\hat{\Delta}_T \leq \inf_{y > 0} \left\{ y + \sum_{t \in [T]} \left( 1 + \frac{a}{y} \right) \left( b_t + \frac{mq}{y} \right) \right\} = \inf_{y > 0} \left\{ y + \frac{ab_{1:T} + mqT}{y} + \frac{amqT}{y^2} \right\} + b_{1:T} \leq 2\sqrt{mqT} + 4a,$$

where the last inequality is because  $b_{1:T} = \sum_{i \in \mathcal{N}_T} \ln \frac{\mathbf{x}_{i,T+1}}{\mathbf{x}_{i,1}} \leq m \ln \frac{N}{m} = a$ , taking  $y = \sqrt{a^2 + mqT}$  and using  $\sqrt{u+v} \leq \sqrt{u} + \sqrt{v}$ .

We proceed to bound  $\max_t \delta_t$ . Using  $\bar{p}_t \geq \mathbf{x}_{i,t} p_{i,t}$  and  $\eta_t/(1-\eta_t) \leq 2\eta_t$  (since  $\eta_0 = \Delta_0/q = \frac{1}{2}$ ),

$$\delta_t \leq \max_i \ln \left( 1 + 2\eta_{t-1} \frac{1}{\mathbf{x}_{i,t}} \right) = \max_i \ln \left( 1 + 2\frac{q}{\Delta_{t-1}} \frac{1}{\mathbf{x}_{i,t}} \right)$$

Furthermore, since  $\bar{p}_t \leq \max_i p_{i,t}$ , then  $\delta_t \geq \ln(1 + \eta_{t-1}/(1-\eta_{t-1})) \geq \eta_{t-1}$ . Also, from the online correction Eq. (3), multiplying by  $\Delta_t$  on both side, we obtain

$$\Delta_t \mathbf{x}_{i,t+1} = \Delta_{t-1} \mathbf{x}_{i,t'} + \delta_t \mathbf{x}_{i,1} \geq \delta_t \mathbf{x}_{i,1} \geq \eta_{t-1} \mathbf{x}_{i,1} = \frac{q}{N\Delta_{t-1}} \tag{20}$$

which gives

$$\delta_t \leq \ln(1 + 2N\Delta_{t-2}) \leq \ln(1 + 2\Delta_T) + \ln N.$$

Therefore, putting it all together:

$$\Delta_T \leq 2\sqrt{mT \ln N} + \ln(1 + 2\Delta_T) + 5m \ln \frac{N}{m} + 3 \ln N,$$

where  $\ln(1 + 2\Delta_T) = \ln O(NT)$ , and the result follows by  $\mathcal{R}_T(\mathbf{x}^*) \leq 2\Delta_T$  from Lemma 12 since  $q = \ln N \geq \phi_1 = \text{RE}(\mathbf{x}^* \parallel \mathbf{x}_1)$ .  $\square$

**Remark 55.** We could even exclude  $O(\sqrt{mT})$  outliers  $\delta_t$  from the analysis, since  $\delta_t \leq \ln O(NT)$ .  $\diamond$

**Remark 56.** Theorem 54 actually applies to all  $m' \in \{m, \dots, N\}$  and in particular for  $m' = N$  for which  $m' \ln \frac{N}{m'} = 0$ .  $\diamond$

**Remark 57.** The leading factor 4 is likely due to using the convenient Theorem 7, but it seems plausible that a more careful analysis would remove a factor  $\sqrt{2}$ .  $\diamond$

## Table of Notation

Notation	Meaning
$[n]$	$\{1, 2, \dots, n\}$
$(x)_+$	$\max\{0, x\}$
$T$	some (unknown) number of rounds to play
$t$	some round index, usually $t \in [T]$
$N$	number of coordinates/experts
$\mathcal{X}$	convex subset of $\mathbb{R}^N$
$\Delta_p(N)$	probability simplex with $N$ coordinates
$\mathbf{x}^*$	some competitor point in $\mathcal{X}$
$\mathbf{x}_t$	point in $\mathcal{X}$ played by the algorithm
$\mathbf{x}_{t'}$	algorithm-specific update of $\mathbf{x}_t$ before online correction
$\ell_t(\mathbf{x})$	OCO: loss function at round $t$
$\nabla_t$	OCO: $\nabla \ell_t$ gradient of the loss function at round $t$
$\ell_t$	OLO: in $\mathbb{R}^N$ , loss vector at round $t$ ; corresponds to $\nabla_t$ in OCO
$\ell_{i,t}$	in $\mathbb{R}$ , loss of coordinate/expert $i$ at round $t$
$\bar{\ell}_t$	$\sum_i \mathbf{x}_{i,t} \ell_{i,t}$ loss of the expert algorithm at round $t$
$G$	OCO: $\max_{t \in [T]} \ \nabla_t\ _*$ , largest observed gradient
$L$	OLO: $\max_{t \in [T]} \ \ell_t\ _*$ , largest observed loss
$L_t$	FTRL: $\sum_{s \in [t]} \ell_s$ , cumulative loss
$\mathcal{R}_T(\mathbf{x}^*)$	cumulative regret of the algorithm compared to $\mathbf{x}^* \in \mathcal{X}$
$r_t$	instantaneous regret of the algorithm, usually $\langle \mathbf{x}_t - \mathbf{x}^*, \ell_t \rangle$ (OLO) or $\ell_t(\mathbf{x}_t) - \ell_t(\mathbf{x}^*)$ (OCO)
$r_{i,t}$	$\bar{\ell}_t - \ell_{i,t}$ instantaneous regret of the algorithm against expert $i$ at round $t$
$s_t$	$\max_{i \in [N]}  r_{i,t} $
$S$	$\max_{t \in [T]} s_t$
$V_{i,T}$	$\sum_{t \in [T]} r_{i,t}^2$
$D$	$\max_{(a,b) \in \mathcal{X}^2} \ a - b\ $ , diameter of $\mathcal{X}$ according to $\ \cdot\ $
$\eta$	learning rate, $\geq 0$
$q$	positive parameter of the algorithms
$\Delta_t$	$\Delta_0 + \sum_{s \in [t]} \Delta_s$
$\delta_t^*$	algorithm-specific quantity, Eq. (4)
$\delta_t$	algorithm-specific quantity, usually satisfies Eq. (5)
$\hat{\delta}_t$	non-increasing continuous positive function such that usually $\delta_t \leq \hat{\delta}_t(\Delta_{t-1})$
$\ \cdot\ $	some norm
$\ \cdot\ _*$	dual norm of $\ \cdot\ $
$R$	regularizer, usually 1-strongly convex w.r.t. some norm $\ \cdot\ $ and non-negative
$B_R$	Bregman divergence
$\text{RE}(\cdot\ \cdot)$	relative entropy
$\phi$	some convex function
$\mathbb{I}_{\{test\}}$	indicator function
$g_t^{\text{low}}$	lower bound of the domain of some function $g_t : (g_t^{\text{low}}, \infty) \rightarrow [0, \infty)$ (Definition 5)
$g_T^{\text{inf}}$	$\max_{t \in [T]} (g_t^{\text{low}})_+$ for a given sequence of functions $\{g_t\}_{t \in [T]}$
$\mathcal{T}_\emptyset$	subset of $[T]$ where a null update is performed (Section 3.5)
$\tau$	$\max \mathcal{T}_\emptyset$ , last round before $T$ a null update is performed (Section 3.5)
$c$	null update factor (Section 3.5)
$\text{iso}(\cdot, \cdot)$	iso operator (Section 3.1)