

Artificial Constraints and Hints for Unbounded Online Learning

Ashok Cutkosky

Google Research

Mountain View, CA, USA

ASHOK@CUTKOSKY.COM

Editors: Alina Beygelzimer and Daniel Hsu

Abstract

We provide algorithms that guarantees regret $R_T(u) \leq \tilde{O}(G\|u\|^3 + G(\|u\| + 1)\sqrt{T})$ or $R_T(u) \leq \tilde{O}(G\|u\|^3 T^{1/3} + GT^{1/3} + G\|u\|\sqrt{T})$ for online convex optimization with G -Lipschitz losses for any comparison point u without prior knowledge of either G or $\|u\|$. Previous algorithms dispense with the $O(\|u\|^3)$ term at the expense of knowledge of one or both of these parameters, while a lower bound shows that some additional penalty term over $G\|u\|\sqrt{T}$ is necessary. Previous penalties were *exponential* while our bounds are polynomial in all quantities. Further, given a known bound $\|u\| \leq D$, our same techniques allow us to design algorithms that adapt optimally to the unknown value of $\|u\|$ without requiring knowledge of G .

1. Unconstrained Online Convex Optimization

Online convex optimization (OCO) is a popular theoretical framework for designing algorithms that operate on streams of input data (Shalev-Shwartz, 2011; Zinkevich, 2003). Such problems abound in today’s world of extremely large datasets, and so many empirically successful algorithms are analyzed in the OCO framework (e.g. (Duchi et al., 2010; Ross et al., 2013; McMahan et al., 2013)). In detail, OCO is a game consisting of T rounds. In each round, the learning algorithm first outputs a point w_t in some Hilbert space W^1 , and then the environment outputs a convex loss function $\ell_t : W \rightarrow \mathbb{R}$. The learner’s goal is to minimize the *regret*, which is the total loss suffered by the learner in comparison to the loss suffered at some benchmark point $\hat{w} \in W$:

$$R_T(\hat{w}) = \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w})$$

Online learning algorithms can be naturally applied to stochastic optimization problems, in which each ℓ_t is an i.i.d. random variable with $\mathbb{E}[\ell_t] = \mathcal{L}$ for some fixed loss function \mathcal{L} . In this case, the online-to-batch conversion argument (Cesa-Bianchi et al., 2004) tells us that $\mathbb{E}[\mathcal{L}(\bar{w}) - \mathcal{L}(\hat{w})] \leq \frac{\mathbb{E}[R_T(\hat{w})]}{T}$, where $\bar{w} = \frac{\sum_{t=1}^T w_t}{T}$. Thus, we refer to $\frac{R_T(\hat{w})}{T}$ as the *convergence rate*. We wish to achieve a convergence rate such that $\lim_{T \rightarrow \infty} \frac{R_T(\hat{w})}{T} = 0$, known as sublinear regret.

We can reduce OCO to online *linear* optimization (OLO) in which each loss ℓ_t must be linear through the use of subgradients: if $g_t \in \partial \ell_t(w_t)$, we have

$$R_T(\hat{w}) \leq \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle \tag{1}$$

1. Our results also apply in Banach spaces, but we focus our exposition on Hilbert spaces for simplicity

Thus by supplying the linear losses $w \mapsto \langle g_t, w \rangle$ to an OLO algorithm, we obtain an OCO algorithm. Because it is often easy to compute gradients, many of the popular OCO algorithms are in fact OLO algorithms. We will also take this road, and consider exclusively the case that losses are linear functions specified by the subgradients g_t so that (1) is an equality.

Our goal is to design algorithms which take no data-dependent parameters as input and yet nevertheless guarantee regret matching the minimax optimal regret for online linear optimization algorithms, sometimes called “parameter-free” algorithms (Abernethy et al., 2008; McMahan and Streeter, 2012; Orabona, 2013; Foster et al., 2015, 2017). Specifically, we want to obtain:

$$\begin{aligned} R_T(\hat{w}) &= O \left(\|\hat{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2 \log(GT + 1)} \right) \\ &= O \left(\|\hat{w}\| G \sqrt{T \log(GT + 1)} \right) \end{aligned} \tag{2}$$

for all \hat{w} simultaneously, where $G = \max \|g_t\|_*$, $\|\cdot\|$ is some norm and $\|\cdot\|_*$ is the dual norm. Classical gradient-descent algorithms require learning rates that are tuned to the values of $\|\hat{w}\|$ and $\|g_t\|_*$, while parameter-free algorithms automatically adapt to these unknown parameters, and so can largely dispense with tuning.

1.1. Previous Progress and Lower Bounds

Previous approaches to designing parameter-free algorithms typically relax the problem by assuming either a bound D on $\|\hat{w}\|$, or a bound \mathfrak{G} on $G = \max_t \|g_t\|_*$. In the case of known value for D , classical approaches based on Follow-the-Regularized-Leader or Mirror Descent with strongly-convex potentials can obtain regret $O \left(D \sqrt{\sum_{t=1}^T \|g_t\|^2} \right)$ (Duchi et al., 2010; Hazan et al., 2008; Orabona and Pál, 2018). In the alternative when a bound $G \leq \mathfrak{G}$ is known, the problem seems somewhat harder. Most prior work in this setting instead obtains the slightly-less-good regret bound $R_T(\hat{w}) = \tilde{O} \left(\|\hat{w}\| \sqrt{\mathfrak{G} \sum_{t=1}^T \|g_t\|_*} \right)$ (Orabona, 2014; Orabona and Tommasi, 2017; Cutkosky and Boahen, 2017), and the better rate of $\tilde{O} \left(\|\hat{w}\| \sqrt{\mathfrak{G}^2 + \sum_{t=1}^T \|g_t\|_*^2} \right)$ has only recently been developed (Cutkosky and Orabona, 2018).

When neither $\|\hat{w}\|$ nor \mathfrak{G} is known, prior lower bounds show that it is actually impossible to obtain the desired result (Cutkosky and Boahen, 2016, 2017). Instead, there is a frontier of lower-bounds trading off between the “ideal” $\tilde{O}(\|\hat{w}\|G\sqrt{T})$ term and an *exponential* penalty of up to $O(\exp(T)/T)$, depending on how fast the gradients grow².

We will improve upon this background in two ways. First, observe that there is a curious asymmetry between the results for known D and known \mathfrak{G} . On the one hand, in the known D case the algorithms do not adapt very well to the true value of $\|\hat{w}\|$, instead obtaining a regret bound proportional to $DG\sqrt{T}$. On the other hand, in the known \mathfrak{G} case, so long as \mathfrak{G} is not greater than $G\sqrt{T}$, the resulting bound depends on $\|\hat{w}\|G\sqrt{T}$. So in a sense, the known \mathfrak{G} algorithms are more robust than than the known D ones. We close this gap by providing an algorithm that takes an upper bound

2. It is possible to sacrifice the near-linear dependence on $\|\hat{w}\|$ to obtain a bound of $\|\hat{w}\|^2 G \sqrt{T}$ without any knowledge (Orabona and Pál, 2018).

D but no bound \mathfrak{G} and maintains regret $\tilde{O}(\|\dot{w}\|G\sqrt{T} + DG)$. The principal technique used in this analysis will then allow us to improve in a second way: when neither \mathfrak{G} nor a bound D is known, we design an algorithm that still guarantees sublinear regret and whose only T -dependent regret terms are linear in $\|\dot{w}\|$.

The lower bound makes it seem that there is no hope for this second goal - but it is not so! The key observation is that the exponential lower bound applies only to algorithms which insist on a regret bound whose dependence on $\|\dot{w}\|$ is $\tilde{O}(\|\dot{w}\|)$. However, in practice we suspect that it is more important to maintain a T -dependence of \sqrt{T} . As a result, we will replace the aesthetically unappealing $\exp(T)/T$ term with a more palatable penalty of $G\|\dot{w}\|^3$. This new regret bound is incomparable to the previous lower bound, in the sense that neither function is dominated by the other: for very large $\|\dot{w}\|$, $\|\dot{w}\|^3 \geq \exp(T)/T$. However, we note that our new penalty is *independent of T* , and so as T becomes large compared to $\|\dot{w}\|$, our asymptotic convergence rate is guaranteed to match the optimal rate. To our knowledge, ours is the first guarantee of this kind in online learning.

2. Overview of Techniques and Results

For any user-specified p and k , We will construct an online linear optimization algorithm, LEASHED, that guarantees the regret bound:

$$R_T(\dot{w}) \leq \tilde{O} \left[\|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2} + G_T k \left(\max_t \frac{\sum_{t'=1}^t \|g_{t'}\|_*}{G_t} \right)^p + G_T \min_{q \in [0,1]} \frac{\|\dot{w}\|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} \left(\frac{\sum_{t=1}^T \|g_t\|_*}{G_T} \right)^q \right] \quad (3)$$

where $G_t = \max_{t' \leq t} \|g_{t'}\|_*$ and $\|\dot{w}\|$ are unknown to the algorithm ahead of time and \tilde{O} hides a logarithmic factor. Using this result, we can apply $p = 1/2$ and $q = 0$ to obtain the bound:

$$\begin{aligned} R_T(\dot{w}) &\leq \tilde{O} \left(\|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2} + G_T k \sqrt{\max_t \frac{\sum_{t'=1}^t \|g_{t'}\|_*}{G_t}} + G_T \frac{\|\dot{w}\|^3}{k^2} \right) \\ &= \tilde{O} \left((\|\dot{w}\| + k) G_T \sqrt{T} + G_T \frac{\|\dot{w}\|^3}{k^2} \right) \end{aligned}$$

Another interesting setting is $p = 1/3$, $q = 1/3$, $k = 1$, which yields $\tilde{O}(G\|\dot{w}\|^3 T^{1/3} + G T^{1/3} + \|\dot{w}\|G\sqrt{T})$ and makes the dominant T -dependent term $\tilde{O}(\|\dot{w}\|G\sqrt{T})$ without any $\|\dot{w}\| + k$ terms.

Finally, observe that setting $p = 1$, $q = 0$ in (3) yields an expression (up to logs) of the form $G\|\dot{w}\|^2/k + GkT + \|\dot{w}\|G\sqrt{T}$, which is reminiscent of the results in FTRL or Mirror Descent analysis with strongly-convex regularizers, in which one needs to tune the “learning rate” k to be $O\left(\frac{1}{\sqrt{T}}\right)$ in order to obtain $O(\sqrt{T})$ regret.

The term $\max_t \frac{\sum_{t'=1}^t \|g_{t'}\|_*}{G_t}$ may seem a little awkward. It appears for technical reasons and it is very easy to replace it with $C \sum_{t=1}^T \|g_t\|_*$ for any user-specified constant C . We choose not to do this simple because the stated expression has the appealing property that scaling all the gradients by

any constant scales the expression by the same constant (although the entire regret bound does not maintain this property).

2.1. Outline and Proof Steps

First, we observe that it suffices to achieve our desired bounds in the one-dimensional case $W = \mathbb{R}$, as it is easy to convert any one-dimensional algorithm to a dimension-free algorithm via a recent reduction argument (Cutkosky and Orabona, 2018) (see Section A for details). Our one-dimensional algorithm is then constructed via three steps:

1. In step 1, we develop an online linear optimization algorithm for a modified setup in which the algorithm is given access to a sequence of “hints” $h_1 \leq \dots \leq h_T$ such that $|g_t| \leq h_t$ and h_t is revealed before g_t . The algorithm utilizes the hints to avoid suffering any penalty for a priori unknown bounds on G . This step is the most technical step, although it is essentially just a careful verification that prior analysis is completely unchanged when incorporating these “just in time” bounds on $|g_t|$ (Section 3).
2. Although these hints aren’t actually available, we approximate them by $h_t = \max_{i < t} |g_i|$. We then analyze the error from this approximation, and show that it results in adding a penalty of $G \max_t |w_t|$ to the regret (Section 4).
3. From the previous step, it seems that we should try to control $\max_t |w_t|$. We do this by enforcing an “artificial constraint”: we use the constraint set reduction in (Cutkosky and Orabona, 2018) to ensure $|w_t| \leq \sqrt{T}$ for all t . This results in good regret for all $|\dot{w}| \leq \sqrt{T}$, but does not control regret for $|\dot{w}| > \sqrt{T}$. To address $|\dot{w}| > \sqrt{T}$, we then observe that $R_T(\dot{w}) \leq R_T(0) + |\dot{w}|GT \leq R_T(0) + G|\dot{w}|^3$ and use the fact that $R_T(0)$ is constant (because $|0| \leq \sqrt{T}$) to conclude the desired results (Section 5).

We point out that the second step of our proof actually gives a rather general way to convert algorithms that require bounded losses into ones that do not require bounded losses, so long as the domain W is itself bounded. Thus, it is our hope that it may have broader applicability.

3. Step 1: Unconstrained Optimization With Hints

Our overall approach can be viewed as a sequence of elaborate “tricks” designed to convert an algorithm that requires a Lipschitz bound G into one that does not. Our first step in this section is to consider a slightly easier OLO game in which the algorithm is given access to “hints” h_t that bound the next loss g_t . Formally, each round of the OLO game now consists of:

1. Learner receives hint $h_t \in \mathbb{R}$, with $h_t \geq h_{t-1}$.
2. Learner plays w_t .
3. Learner receives loss g_t with $|g_t| \leq h_t$.

We will look for regret bounds that depend both on \dot{w} as well as the hints, taking the form:

$$\sum_{t=1}^T g_t(w_t - \dot{w}) \leq R_T(\dot{w}, h_T)$$

To accomplish this, we adapt the algorithm of (Cutkosky and Orabona, 2018) to this setting and obtain an algorithm that guarantees regret

$$R_T(\hat{w}) = \tilde{O} \left[|\hat{w}| \max \left(h_T, \sqrt{\sum_{t=1}^T g_t^2} \right) \right]$$

Previous algorithms (Orabona, 2014; Cutkosky and Boahen, 2017) can also be adapted to this setting, but the dependencies on h_T are worse.

The algorithm (Algorithm 1) operates in the coin-betting framework (Orabona and Pál, 2016). A coin-betting algorithm achieves low regret by maintaining high *wealth*, defined by $\text{Wealth}_T = \epsilon - \sum_{t=1}^T g_t w_t$ for some user-specified $\epsilon > 0$. The wealth is increased by playing $w_t = v_t \text{Wealth}_t$ for some $v_t \in [-1, 1]$, which corresponds to “betting” a fraction v_t of Wealth_t on the value of the “coin” g_t , because $\text{Wealth}_{t+1} = \text{Wealth}_t - g_t v_t \text{Wealth}_t$. Thus the problem of choosing w_t reduces to the problem of choosing v_t . We solve this problem in the same way as (Cutkosky and Orabona, 2018) by recasting choosing v_t as an online exp-concave optimization problem in 1 dimension, and then use the Online Newton Step (ONS) algorithm (Hazan et al., 2007) to optimize v_t .

This strategy can become nonsensical if the value of Wealth_t ever becomes non-positive. To avoid this, we wish to guarantee $|v_t| < 1/|g_t|$ for all t , which implies $\text{Wealth}_{t+1} = (1 - v_t g_t) \text{Wealth}_t > 0$. Thus, we use our “hint” h_t and restrict the value v_t to the range $[-1/2h_t, 1/2h_t]$ to achieve the desired outcome.

Algorithm 1 Coin-Betting through ONS With Hints

Require: Initial wealth $\epsilon > 0$, parameter $\alpha > 0$

- 1: **Initialize:** $\text{Wealth}_0 = \epsilon$, initial betting fraction $v_1 = 0$, initial hint h_1
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Bet $w_t = v_t \text{Wealth}_{t-1}$
 - 4: Receive $g_t \leq h_t$
 - 5: Receive $h_{t+1} \geq h_t$
 - 6: Update $\text{Wealth}_t = \text{Wealth}_{t-1} - \tilde{g}_t w_t$
 - 7: compute new betting fraction $v_{t+1} \in [-1/2h_{t+1}, 1/2h_{t+1}]$ via ONS update on losses $-\ln(1 - \tilde{g}_t v)$
 - 8: Set $z_t = \frac{d}{dv_t} (-\ln(1 - \tilde{g}_t v_t)) = \frac{\tilde{g}_t}{1 - \tilde{g}_t v_t}$
 - 9: Set $A_t = 4\alpha + \sum_{i=1}^t z_i^2$
 - 10: $v_{t+1} = \max \left(\min \left(v_t - \frac{2}{2 - \ln(3)} \frac{z_t}{A_t}, \frac{1}{2h_{t+1}} \right), -\frac{1}{2h_{t+1}} \right)$
 - 11: **end for**
-

The analysis of this algorithm is nearly identical to that in (Cutkosky and Orabona, 2018), although we reproduce the main steps for completeness. The major deviation is that we are performing ONS updates on shrinking domains $[-1/2h_t, 1/2h_t]$, which we analyze carefully in appendix.

Theorem 1 *The regret of Algorithm 1 is bounded by:*

$$R_T(\dot{w}, h_T) \leq \epsilon + |\dot{w}| \max \left(8h_T \left(\ln \frac{16|\dot{w}|h_T \exp(\alpha/4h_T^2) \left(1 + \frac{\sum_{t=1}^T g_t^2}{\alpha}\right)^{4.5}}{\epsilon} - 1 \right), \right. \\ \left. 2 \sqrt{\sum_{t=1}^T g_t^2 \ln \left(\frac{4 \left(\sum_{t=1}^T g_t^2\right)^{10} \exp(\alpha/2h_T^2) \dot{w}^2}{\epsilon^2} + 1 \right)} \right)$$

4. Step 2: Without Hints, Regret is Small if $\|w_t\|$ is Small

In this section we remove the need for externally supplied hints used in Section 3. The technique presented here may apply more generally than our present focus, so we state the technique in terms of general norms, rather than restricting to a one-dimensional problem as we do in the other sections. Specifically, we show how to convert any algorithm that uses hints h_t and obtains regret $R_T(\dot{w}, h_T)$ into one that does not receive hints and obtains regret $R_T(\dot{w}, G) + G \max_t \|w_t\|$ (where again $G = \max_t \|g_t\|_*$). The procedure is very simple: we run the algorithm using (incorrect!) hints $h_t = \max_{i < t} \|g_i\|_*$. Whenever we observe $\|g_t\| > h_t$ (i.e. when the hint is wrong), we “lie” to the algorithm: we replace g_t with a “truncated gradient” $g_t^{\text{trunc}} = h_t \frac{g_t}{\|g_t\|_*}$ to make the hint correct, and then bound the error produced by this truncation. Pseudocode is provided in Algorithm 2.

Algorithm 2 Algorithm Without Hints

Require: Algorithm \mathcal{A} that takes hints h_t , initial value \mathbf{g}

- 1: **Initialize:** initialize $h_1 = \mathbf{g}$
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Get w_t from \mathcal{A} , play w_t
 - 4: Receive g_t
 - 5: **if** $\|g_t\| \geq h_t$ **then**
 - 6: $g_t^{\text{trunc}} \leftarrow h_t \frac{g_t}{\|g_t\|_*}$
 - 7: **else**
 - 8: $g_t^{\text{trunc}} \leftarrow g_t$
 - 9: **end if**
 - 10: $h_{t+1} = \max(h_t, \|g_t\|_*)$
 - 11: send g_t^{trunc} and h_{t+1} to \mathcal{A}
 - 12: **end for**
-

Theorem 2 *Suppose \mathcal{A} obtains $R_T(\dot{w}, h_T)$ given hints $h_1 \leq \dots \leq h_T$. Then Algorithm 2 obtains*

$$R_T(\dot{w}) \leq R_T(\dot{w}, \max(\mathbf{g}, G)) + G \max_t \|w_t\| + G \|\dot{w}\|$$

where $G = \max_t \|g_t\|_*$.

Proof First, we observe that the gradients g_t^{trunc} provided to \mathcal{A} do indeed respect the hints, $\|g_t^{\text{trunc}}\|_* \leq h_t$. Thus we have:

$$\sum_{t=1}^T g_t^{\text{trunc}} \cdot w_t - g_t^{\text{trunc}} \cdot \hat{w} \leq R_T(\hat{w}, h_T)$$

Moving on to the true regret, we define $G_t = \max_{i \leq t} \|g_i\|_*$ for convenience, and then compute:

$$\begin{aligned} \sum_{t=1}^T g_t \cdot w_t - g_t \cdot \hat{w} &= \sum_{t=1}^T g_t^{\text{trunc}} \cdot (w_t - \hat{w}) + (g_t - g_t^{\text{trunc}}) \cdot (w_t - \hat{w}) \\ &\leq R_T(\hat{w}, h_T) + (\|\hat{w}\| + \max_t \|w_t\|) \sum_{t=1}^T \|g_t - g_t^{\text{trunc}}\|_* \\ &\leq R_T(\hat{w}, h_T) + (\|\hat{w}\| + \max_t \|w_t\|) \sum_{t|h_t < G_t} G_t - h_t \\ &\leq R_T(\hat{w}, h_T) + (\|\hat{w}\| + \max_t \|w_t\|) \sum_{t|h_t < G_t} G_t - G_{t-1} \\ &\leq R_T(\hat{w}, h_T) + (\|\hat{w}\| + \max_t \|w_t\|) G \end{aligned}$$

where we have observed $h_t \geq G_{t-1}$ in the second-to-last line. Now we see that $h_T = \max(\mathfrak{g}, G)$ to complete the proof. \blacksquare

Combining this result with our Algorithm 1, we obtain a regret bound of

$$R_T(\hat{w}) \leq \tilde{O} \left[\|\hat{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2} + \max(\mathfrak{g}, G) \|\hat{w}\| + G \max_t \|w_t\| + \epsilon \right]$$

where we have temporarily suppressed all logarithmic factors for ease of exposition. Intuitively, we obtain nearly the same regret guarantee as before, but suffer an additional penalty that is small so long as the w_t s do not grow too much.

This reduction allows us to address the asymmetry between the prior algorithms with known D versus known \mathfrak{G} . Specifically, when we operate within a setting with bounded diameter D , then $\max_T \|w_t\| \leq D$. We can construct an algorithm with domain W that takes hints and obtains regret that adapts to $\|\hat{w}\|$ by applying the one-dimensional-to-dimension-free reductions and unconstrained-to-constrained reductions of (Cutkosky and Orabona, 2018) to Algorithm 1. Then by applying the reduction of this section and leveraging $\max_T \|\hat{w}_t\| \leq D$, we obtain the regret bound:

$$R_T(\hat{w}) \leq \tilde{O} \left(\|\hat{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2} + DG \right)$$

Further, by applying these reductions to Algorithm 6 of (Cutkosky and Orabona, 2018) we can obtain:

$$R_T(\hat{w}) \leq \tilde{O} \left(\sqrt{\sum_{t=1}^T \|w_t - \hat{w}\|^2 \|g_t\|_*^2} + DG \right)$$

The latter regret bound is of interest as it implies logarithmic regret on strongly-convex losses without requiring any knowledge of the strong-convexity parameter or any Lipschitz bounds. We expect that other algorithms involving bounded domains and Lipschitz bounds can also take advantage of this technique to remove the Lipschitz bound requirement.

5. Step 3: Artificial Constraints

Returning to a one-dimensional problem, in this last step, we leverage the result of the previous section by preventing our algorithm from choosing w_t s with overly-large magnitudes. To gain some intuition for our strategy, suppose we can constrain the algorithm in the previous section to the set $\left[-\sqrt{\sum_{t=1}^T |g_t|}, \sqrt{\sum_{t=1}^T |g_t|}\right]$, while still maintaining the same regret bound for any \hat{w} in this interval³. This enforces $\max_t \|w_t\| \leq \sqrt{\sum_{t=1}^T \|g_t\|}$, so that intuitively we have a bound of

$$R_T(\hat{w}) \leq \tilde{O} \left[\epsilon + G \sqrt{\sum_{t=1}^T |g_t|} + |\hat{w}| \sqrt{\sum_{t=1}^T g_t^2 + \epsilon} \right].$$

for any \hat{w} with $|\hat{w}| \leq \sqrt{\sum_{t=1}^T |g_t|}$. Thus it remains to address \hat{w} outside the constraining interval. For \hat{w} outside the interval, we have

$$\begin{aligned} R_T(\hat{w}) &= \sum_{t=1}^T g_t w_t - g_t \hat{w} \\ &\leq R_T(0) + |\hat{w}| \sum_{t=1}^T |g_t| \\ &\leq R_T(0) + |\hat{w}|^3 \end{aligned}$$

Where in the last step we used $|\hat{w}| \geq \sqrt{\sum_{t=1}^T |g_t|}$. Combining the two guarantees, with our result from the previous section for which $R_T(0) \leq \epsilon$, we have for all \hat{w} :

$$R_T(\hat{w}) \leq \tilde{O} \left[|\hat{w}|^3 + G \sqrt{\sum_{t=1}^T |g_t|} + G|\hat{w}| + \epsilon + \max(\mathbf{g}, G) \|\hat{w}\| + |\hat{w}| \sqrt{\sum_{t=1}^T g_t^2} \right].$$

There are two issues with this intuition that need to be addressed. First, we need to show how to restrict to the desired interval without affecting the regret bound for \hat{w} inside the interval, and second we need to deal with the fact that we do not know the value of $\sum_{t=1}^T |g_t|$ apriori. We address the first issue by appealing to the constraint-set reduction of (Cutkosky and Orabona, 2018), which provides exactly the desired mechanism (full details are reproduced in our proof of Theorem 3). We address the second issue in greater generality by considering a t -varying constraint-set $\left[-k \left(\sum_{i=1}^{t-1} |g_i|\right)^p, k \left(\sum_{i=1}^{t-1} |g_i|\right)^p\right]$ for user-specified p and k .

We present the pseudo-code for the final algorithm in Algorithm 3, LEASHED below.

Algorithm 3 LEASHED

Require: Algorithm \mathcal{A} that takes hints, parameters k, p, \mathbf{g}

```

1: Initialize: initialize  $h_1 = \mathbf{g}, G_0 = 0, B_1 = 0$ 
2: Send initial hint  $h_1$  to  $\mathcal{A}$ 
3: for  $t = 1$  to  $T$  do
4:   Get  $w_t$  from  $\mathcal{A}$ 
5:   if  $|w_t| \geq B_t$  then
6:     //project to artificial constraint set  $[-B_t, B_t]$ 
7:      $\tilde{w}_t \leftarrow B_t \frac{w_t}{|w_t|}$ 
8:   else
9:      $\tilde{w}_t \leftarrow w_t$ 
10:  end if
11:  Play  $\tilde{w}_t$ , receive  $g_t$ 
12:   $G_t \leftarrow \max(G_{t-1}, |g_t|)$ .
13:   $h_{t+1} \leftarrow \max(h_t, |g_t|)$ .
14:  //update artificial constraint
15:   $B_{t+1} \leftarrow k \left( \sum_{i=1}^t |g_i| / G_t \right)^p$ 
16:  //deal with increasing gradient sizes
17:  if  $|g_t| \geq h_t$  then
18:    //replace  $g_t$  with truncated version
19:     $g_t^{\text{trunc}} \leftarrow h_t \frac{g_t}{|g_t|}$ 
20:  else
21:    //no need to modify  $g_t$ 
22:     $g_t^{\text{trunc}} \leftarrow g_t$ 
23:  end if
24:  //modify gradient to respect artificial constraint
25:  Set  $\tilde{\ell}_t(w) = \frac{1}{2} (g_t^{\text{trunc}} w + |g_t^{\text{trunc}}| \max(0, |w| - B_t))$ 
26:  Compute  $\tilde{g}_t \in \partial \tilde{\ell}_t(w_t)$ 
27:  Send  $\tilde{g}_t$  and  $h_{t+1}$  to  $\mathcal{A}$ 
28: end for
    
```

5.1. The Final Algorithm

Theorem 3 Suppose \mathcal{A} guarantees regret $R_T^{\mathcal{A}}(\hat{w}, h_T)$ given gradients $\tilde{g}_1, \dots, \tilde{g}_T$ and hints $h_1 \leq \dots \leq h_T$ such that $|\tilde{g}_t| \leq h_t$. Then LEASHED obtains regret

$$\begin{aligned}
 R_T(\hat{w}) &\leq 2R_T^{\mathcal{A}}(\hat{w}, \max(\mathbf{g}, G)) + Gk \left[\max_{t \leq T} \left(\sum_{i=1}^t |g_i| / G_t \right)^p \right] + 2G|\hat{w}| \\
 &\quad + \min_{q \in [0,1]} G \left[\frac{|\hat{w}|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} \left(\sum_{t=1}^T |g_t| / G \right)^q \right]
 \end{aligned}$$

where $G = \max_{t \leq T} |g_t|$.

3. Note that we don't know this interval a priori - this is just a thought-experiment to gain intuition.

Proof The additional components of this proof over that of Theorem 2 are inspired by the proof of the constraint-set reduction in (Cutkosky and Orabona, 2018) (Theorem 3), with some modification to deal with the time-varying constraints.

First, we mirror the argument of Theorem 2:

$$\begin{aligned}
 \sum_{t=1}^T g_t(\tilde{w}_t - \dot{w}) &\leq \sum_{t=1}^T g_t^{\text{trunc}}(\tilde{w}_t - \dot{w}) + (g_t - g_t^{\text{trunc}})(\tilde{w}_t - \dot{w}) \\
 &\leq \sum_{t=1}^T g_t^{\text{trunc}}(\tilde{w}_t - \dot{w}) + \sum_{t=1}^T |g_t - g_t^{\text{trunc}}|(\max_t |\tilde{w}_t| + |\dot{w}|) \\
 &\leq \sum_{t=1}^T g_t^{\text{trunc}}(\tilde{w}_t - \dot{w}) + G \left[\max_t B_t + |\dot{w}| \right]
 \end{aligned}$$

Now we deal with the first term. Let $\tilde{\tilde{w}}_t$ be the projection of \dot{w} to $[-B_t, B_t]$. Then:

$$\begin{aligned}
 \sum_{t=1}^T g_t^{\text{trunc}}(\tilde{w}_t - \dot{w}) &\leq \sum_{t=1}^T g_t^{\text{trunc}} w_t + |g_t^{\text{trunc}}| |\tilde{w}_t - w_t| - (g_t^{\text{trunc}} \dot{w} + |g_t^{\text{trunc}}| |\dot{w} - \tilde{\tilde{w}}_t|) + |g_t^{\text{trunc}}| |\dot{w} - \tilde{\tilde{w}}_t| \\
 &= 2 \sum_{t=1}^T \tilde{\ell}_t(w_t) - \tilde{\ell}_t(\dot{w}) + \sum_{t=1}^T |g_t^{\text{trunc}}| |\dot{w} - \tilde{\tilde{w}}_t| \\
 &\leq 2 \sum_{t=1}^T \tilde{g}_t(w_t - \dot{w}) + \sum_{t=1}^T |g_t^{\text{trunc}}| |\dot{w} - \tilde{\tilde{w}}_t|
 \end{aligned}$$

We will analyze these two sums separately. First, observe that $\tilde{\ell}_t$ is $|g_t^{\text{trunc}}|$ -Lipschitz, so that $|\tilde{g}_t| \leq |g_t^{\text{trunc}}| \leq |g_t| \leq h_t$ for all t . Therefore we have

$$2 \sum_{t=1}^T \tilde{g}_t(w_t - \dot{w}) \leq 2R_T^A(\dot{w}, h_T) = 2R_T^A(\dot{w}, \max(\mathbf{g}, G))$$

Where we have observed that w_t is generated by running \mathcal{A} on gradients \tilde{g}_t (which satisfy $|\tilde{g}_t| \leq |g_t|$). For the second sum, we have $|g_t^{\text{trunc}}| \leq |g_t|$ and $|\dot{w} - \tilde{\tilde{w}}_t| \leq |\dot{w}|$ so that

$$\begin{aligned}
 \sum_{t=1}^T |g_t^{\text{trunc}}| |\dot{w} - \tilde{\tilde{w}}_t| &\leq \sum_{t: \dot{w} \neq \tilde{\tilde{w}}_t} |g_t| |\dot{w}| \\
 &\leq \sum_{t: |\dot{w}| \geq k(\sum_{i=1}^{t-1} |g_i|/G)^p} |g_t| |\dot{w}|
 \end{aligned}$$

Let \mathcal{T} be the largest value in $\{1, \dots, T\}$ such that $|\dot{w}| \geq B_{\mathcal{T}} = k \left(\sum_{i=1}^{\mathcal{T}-1} |g_i|/G_{\mathcal{T}-1} \right)^p$. Then

$$\begin{aligned}
 \sum_{t=1}^{\mathcal{T}} |g_t^{\text{trunc}}| |\dot{w} - \tilde{w}_t| &\leq G_{\mathcal{T}} |\dot{w}| + \sum_{t=1}^{\mathcal{T}-1} |g_t| |\dot{w}| \\
 &\leq G_{\mathcal{T}} |\dot{w}| + \min_{q \in [0,1]} \left[\left(\sum_{t=1}^{\mathcal{T}-1} |g_t| \right)^q \left(\sum_{t=1}^{\mathcal{T}-1} |g_t| \right)^{1-q} |\dot{w}| \right] \\
 &\leq G_{\mathcal{T}} |\dot{w}| + \min_{q \in [0,1]} \left[\frac{|\dot{w}|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} G_{\mathcal{T}-1}^{1-q} \left(\sum_{t=1}^{\mathcal{T}-1} |g_t| \right)^q \right] \\
 &\leq G_{\mathcal{T}} |\dot{w}| + \min_{q \in [0,1]} \left[\frac{|\dot{w}|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} G_{\mathcal{T}} \left(\sum_{t=1}^{\mathcal{T}} |g_t|/G_{\mathcal{T}} \right)^q \right]
 \end{aligned}$$

where in the second step we used $|\dot{w}| \geq k \left(\sum_{i=1}^{\mathcal{T}-1} |g_i| \right)^p$.

Putting all this together, we have

$$\begin{aligned}
 R_T(\dot{w}) &\leq 2R_T^A(\dot{w}, \max(\mathbf{g}, G_T)) \\
 &\quad + G_T \left[k \min_{t \leq T} \left(\sum_{i=1}^t |g_i|/G_t \right)^p + 2|\dot{w}| \right] + \min_{q \in [0,1]} \left[G_T \frac{|\dot{w}|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} \left(\sum_{t=1}^T |g_t|/G_T \right)^q \right]
 \end{aligned}$$

If we combine this reduction with our result from Section 3, we obtain the following:

Corollary 4 *Applying the reduction of Algorithm 3 to Algorithm 1, we guarantee regret:*

$$\begin{aligned}
 R_T(\dot{w}) &\leq 2\epsilon + 2|\dot{w}| \max \left[8h_T \ln \left(\frac{16|\dot{w}|h_T \exp(\alpha/4h_T^2) \left(1 + \frac{\sum_{t=1}^T g_t^2}{\alpha} \right)^{4.5}}{\epsilon} \right) - h_T, \right. \\
 &\quad \left. 2 \sqrt{\sum_{t=1}^T g_t^2 \ln \left(\frac{4 \left(\sum_{t=1}^T g_t^2 \right)^{10} \exp(\alpha/4h_T^2) \dot{w}^2}{\epsilon^2} + 1 \right)} \right] \\
 &\quad + G \left[k \max_{t \leq T} \left(\sum_{i=1}^t |g_i|/G_t \right)^p + 2|\dot{w}| \right] + \min_{q \in [0,1]} \left[G \frac{|\dot{w}|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} \left(\sum_{t=1}^T |g_t|/G \right)^q \right]
 \end{aligned}$$

where $G = \max_t |g_t|$ and $h_T = \max(\mathbf{g}, G)$.

6. Discussion of Parameters

Although our algorithm does not need to know the data-dependent parameters $\|\dot{w}\|$ and G , we nevertheless retain dependence on several user-specified parameters which we discuss in this section. In brief, the parameters are:

1. ϵ : The regret at the origin.
2. \mathfrak{g} : Initial hint value, ideally this should be set to an under-estimate of G .
3. α : Initial regularizer for ONS.
4. k and p : These control how fast the values of w_t are allowed to grow.
5. q : This exists only for analysis purposes and controls the tradeoff between higher-order dependence on $|\dot{w}|$ and lower-order dependence on T .

Of these parameters, we observe that both ϵ and \mathfrak{g} appear only in logarithmic terms. As a result, our algorithm is robust to these parameters. It remains to investigate k, p and q , which we do by considering a few settings of interest already highlighted in the introduction.

1. With the setting $p = 1/2, q = 0$, our regret bound takes the form:

$$R_T(\dot{w}) \leq \tilde{O} \left((|\dot{w}| + k)G\sqrt{T} + G|\dot{w}| + G\frac{|\dot{w}|^3}{k^2} + \epsilon \right)$$

2. With the setting $p = q = 1/3$, our regret bound takes the form:

$$R_T(\dot{w}) \leq \tilde{O} \left(|\dot{w}|G\sqrt{T} + G|\dot{w}| + \left(\frac{|\dot{w}|^3}{k^2} + k \right) GT^{1/3} \right)$$

We note that in all cases it appears that the optimal value of k is $O(|\dot{w}|)$, so that k is playing a similar role to the scaling of a learning rate in gradient-descent style algorithms. However, the optimal k does not depend on T and so we retain $O(\sqrt{T})$ regret no matter what value is chosen for k . The second example above has the interesting property that for large enough T , the dominant term is $\tilde{O}(\|\dot{w}\|G\sqrt{T})$ for any fixed $\dot{w} \neq 0$ for *any* choice of k (we remove the $kG\sqrt{T}$ term), so that for large T we obtain the optimal scaling with respect to $|\dot{w}|$ even for very small $|\dot{w}|$.

7. Conclusion and Open Problems

We have presented a new online convex optimization algorithm, LEASHED, which adapts to both unknown $\|\dot{w}\|$ and G while guaranteeing sublinear regret. Although the only T -dependent term in LEASHED’s regret bound matches the optimal bound of $\tilde{O} \left(\|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2} \right)$, we avoid exponential lower bounds by adding a T -independent penalty $O(\|\dot{w}\|^3)$. Our algorithm’s principle hyperparameter is the value k , which “morally” should be an estimate of $\|\dot{w}\|$. As a result, in the large- T limit, our our regret grows as $\tilde{O}(\|\dot{w}\|G\sqrt{T})$ without knowledge of either $\|\dot{w}\|$ or G .

There are a few natural open problems suggested by this work. First, our technique provides a simple way to “sidestep” the lower-bound frontier of (Cutkosky and Boahen, 2017), and so naturally suggests the question of whether there is an extension to this frontier that provides some guidance into whether our regret bounds are optimal. Second, we point out that our dependence on values such as \mathfrak{g} or ϵ is a bit undesirable and could hopefully be improved - both of these should ideally be proportional to G , and when they are not the algorithm is not quite “scale-free”. Finally, our regret bound maintains a dependence on $\sqrt{\sum_{t=1}^T \|g_t\|_*}$ rather than $\sqrt{\sum_{t=1}^T \|g_t\|_*^2}$. The latter bound would provide much better behavior on smooth losses (Srebro et al., 2010), and so we hope future work will yield such an improved algorithm.

References

- Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proc. of the nineteenth annual conference on computational learning theory*, 2008.
- Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50(9):2050–2057, 2004.
- Ashok Cutkosky and Kwabena Boahen. Online learning without prior information. In Satyen Kale and Ohad Shamir, editors, *Proc. of the 2017 Conference on Learning Theory*, volume 65 of *Proc. of Machine Learning Research*, pages 643–677, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.
- Ashok Cutkosky and Kwabena A Boahen. Online convex optimization with unconstrained domains and losses. In *Advances in Neural Information Processing Systems 29*, pages 748–756, 2016.
- Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. *arXiv preprint arXiv:1802.06293*, 2018.
- J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory (COLT)*, 2010.
- Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Adaptive online learning. In *Advances in Neural Information Processing Systems 28*, pages 3375–3383. 2015.
- Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems*, pages 6022–6032, 2017.
- Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- Elad Hazan, Alexander Rakhlin, and Peter L Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pages 65–72, 2008.
- Brendan McMahan and Matthew Streeter. No-regret algorithms for unconstrained online convex optimization. In *Advances in neural information processing systems*, pages 2402–2410, 2012.
- H. Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *arXiv preprint arXiv:1403.3465*, 2014.
- H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.
- Francesco Orabona. Dimension-free exponentiated gradient. In *Advances in Neural Information Processing Systems*, pages 1806–1814, 2013.

- Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.
- Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems 29*, pages 577–585, 2016.
- Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716: 50–69, 2018.
- Francesco Orabona and Tatiana Tommasi. Backprop without learning rates through coin betting. *CoRR*, abs/1705.07795, 2017. URL <http://arxiv.org/abs/1705.07795>.
- Stephane Ross, Paul Mineiro, and John Langford. Normalized online learning. In *Proc. of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.
- Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems 23*, pages 2199–2207, 2010.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

Appendix A. Dimension-Free Bound in Banach Spaces

In this section, we observe that by use of the one-dimensional to dimension-free reduction proposed by (Cutkosky and Orabona, 2018), we may seamlessly convert the result of Theorem 3 into a dimension-free regret bound, resulting in Algorithm 4. We give pseudo-code for this reduction for completeness below.

Algorithm 4 Dimension-Free LEASHED

Require: Parameters k, p, g, ϵ, τ , Banach space W .

- 1: **Initialize:** Instantiate LEASHED with Algorithm 1, k, p and g as \mathcal{A}_{1D} . Instantiate an adaptive unit-ball algorithm \mathcal{A}_B
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Get x_t from \mathcal{A}_{1D}
 - 4: Get y_t from \mathcal{A}_B
 - 5: Play $w_t = x_t y_t$.
 - 6: Receive gradient g_t .
 - 7: Send g_t to \mathcal{A}_B .
 - 8: Send $\langle g_t, y_t \rangle$ to \mathcal{A}_{1D} .
 - 9: **end for**
-

Corollary 5 *Suppose A_B guarantees regret $R_T^{A_B}(z)$ for any z in the unit ball. Then Dimension-Free LEASHED guarantees regret:*

$$\begin{aligned}
 R_T(\dot{w}) \leq & 2\epsilon + 2\|\dot{w}\| \max \left[8h_T \ln \left(\frac{16\|\dot{w}\|h_T \exp(\alpha/4h_T^2) \left(1 + \frac{\sum_{t=1}^T \|g_t\|^2}{\alpha}\right)^{4.5}}{\epsilon} \right) - h_T, \right. \\
 & \left. 2 \sqrt{\sum_{t=1}^T \|g_t\|_*^2 \ln \left(\frac{4 \left(\sum_{t=1}^T \|g_t\|_*^2\right)^{10} \exp(\alpha/4h_T^2) \dot{w}^2}{\epsilon^2} + 1 \right)} \right] \\
 & + G \left[k \max_{t \leq T} \left(\sum_{i=1}^t \|g_i\|_* / G_t \right)^p + 2|\dot{w}| \right] + \min_{q \in [0,1]} \left[G \frac{\|\dot{w}\|^{1+\frac{1-q}{p}}}{k^{\frac{1-q}{p}}} \left(\sum_{t=1}^T \|g_t\| / G \right)^q \right] \\
 & + R_T^{A_B}(\dot{w} / \|\dot{w}\|)
 \end{aligned}$$

where $G = \max_t \|g_t\|_*$ and $h_T = \max(\mathfrak{g}, G)$.

As an important special case, when W is a Hilbert space we can obtain $R_T^{A_B}(z) \leq 2^{3/2} \sqrt{\sum_{t=1}^T \|g_t\|^2}$ via standard Adagrad-style analysis (which we reproduce in Section D for completeness).

Appendix B. Proof of Theorem 1

We restate Theorem 1 below for reference:

Theorem 1 *The regret of Algorithm 1 is bounded by:*

$$\begin{aligned}
 R_T(\dot{w}, h_T) \leq & \epsilon + |\dot{w}| \max \left(8h_T \left(\ln \frac{16|\dot{w}|h_T \exp(\alpha/4h_T^2) \left(1 + \frac{\sum_{t=1}^T g_t^2}{\alpha}\right)^{4.5}}{\epsilon} - 1 \right), \right. \\
 & \left. 2 \sqrt{\sum_{t=1}^T g_t^2 \ln \left(\frac{4 \left(\sum_{t=1}^T g_t^2\right)^{10} \exp(\alpha/2h_T^2) \dot{w}^2}{\epsilon^2} + 1 \right)} \right)
 \end{aligned}$$

Proof First, we recall the connection between wealth and regret. If we can prove $\text{Wealth}_T \geq f\left(-\sum_{t=1}^T g_t\right)$ for some function f , then we have:

$$R_T(\dot{w}) \leq \epsilon - \dot{w} \sum_{t=1}^T g_t - f\left(-\sum_{t=1}^T g_t\right) \leq \sup_X \epsilon + X\dot{w} - f(X) = \epsilon + f^*(\dot{w})$$

where f^* is the Fenchel conjugate of f . Thus it suffices to prove a lower-bound on the wealth of our algorithm.

Define $\text{Wealth}(\hat{v})$ as the wealth of an algorithm that uses betting fraction \hat{v} on every round. Then we have the recursions:

$$\begin{aligned}\log(\text{Wealth}_T) &= \log(\epsilon) + \sum_{t=1}^T \log(1 - v_t g_t) \\ \log(\text{Wealth}(\hat{v})) &= \log(\epsilon) + \sum_{t=1}^T \log(1 - \hat{v} g_t)\end{aligned}$$

Now suppose we choose v_t via an online learning algorithm on the losses $-\log(1 - v g_t)$, obtaining regret $R_T^v(\hat{v})$. Subtracting the log-wealth equations and exponentiating, we have

$$\text{Wealth}_T \geq \frac{\text{Wealth}(\hat{v})}{\exp(R_T^v(\hat{v}))}$$

Choose $\hat{v} = \frac{\sum_{t=1}^T g_t}{2 \sum_{t=1}^T g_t^2 + 2h_T |\sum_{t=1}^T g_t|} \in [-1/2h_T, 1/2h_T]$. Then, using $\log(1 + x) \geq x - x^2$ for $|x| \leq 1/2$, we have

$$\log(\text{Wealth}(\hat{v})) \geq \log(\epsilon) + \frac{|\sum_{t=1}^T g_t|^2}{4 \sum_{t=1}^T g_t^2 + 4h_T |\sum_{t=1}^T g_t|}$$

which implies

$$\text{Wealth}_T \geq \epsilon \frac{\exp\left(\frac{|\sum_{t=1}^T g_t|^2}{4 \sum_{t=1}^T g_t^2 + 4h_T |\sum_{t=1}^T g_t|}\right)}{\exp(R_T^v(\hat{v}))}$$

Now it remains to compute $R_T^v(\hat{v})$. In the standard ONS bound, this is $O(\log(T))$. However, our setting is slightly more subtle because we have the shrinking domains $S_t = [-1/2h_t, 1/2h_t]$. It turns out that this has essentially zero effect on the analysis, but we recapitulate the argument in Section C for completeness (see Lemma 10). The final result is that

$$R_T^v(\hat{v}) \leq \frac{\alpha}{4h_T^2} + 4.5 \log\left(\frac{\alpha + \sum_{t=1}^T g_t^2}{\alpha}\right)$$

from which we obtain

$$\text{Wealth}_T \geq \epsilon \frac{\exp\left(\frac{|\sum_{t=1}^T g_t|^2}{4 \sum_{t=1}^T g_t^2 + 4h_T |\sum_{t=1}^T g_t|}\right)}{\exp(\alpha/4h_T^2) \left(\frac{\alpha + \sum_{t=1}^T g_t^2}{\alpha}\right)^{4.5}}$$

Set $a = \frac{\epsilon}{\exp(1/4h_T^2) \left(\frac{\alpha + \sum_{t=1}^T g_t^2}{\alpha}\right)^{4.5}}$, $b = \frac{\alpha}{4h_T}$ and $c = \frac{\sum_{t=1}^T g_t^2}{h_T}$. Then we can write

$$\text{Wealth}_T \geq a \exp\left(b \frac{(\sum_{t=1}^T g_t)^2}{|\sum_{t=1}^T g_t| + c}\right)$$

Algorithm 5 ONS with shrinking domains

Require: $\tau, \beta > 0$

- 1: **Initialize:** Interval $S_1 \subset \mathbb{R}$, $v_1 = 0 \in S_1$
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Play v_t
 - 4: Receive z_t
 - 5: Receive interval $S_{t+1} \subset S_t$
 - 6: Set $A_t = \tau + \sum_{i=1}^t z_i^2$
 - 7: $v_{t+1} = \Pi_{S_{t+1}}\left(v_t - \frac{z_t}{\beta A_t}\right)$, where $\Pi_{S_{t+1}}(x)$ is the projection of x to S_{t+1} (i.e. a truncation).
 - 8: **end for**
-

so if we define $f = a \exp\left(b \frac{x^2}{|x|+c}\right)$, we have

$$R_T(\dot{w}) \leq \epsilon + f^*(\dot{w})$$

We recall the computation of f^* in Lemma 11, to obtain:

$$\begin{aligned} R_T(\dot{w}) &\leq \epsilon + |\dot{w}| \max\left(\frac{2}{b} \left(\ln \frac{2|\dot{w}|}{ab} - 1\right), \sqrt{\frac{c}{b} \ln\left(\frac{c\dot{w}^2}{a^2b} + 1\right)} - a\right) \\ &\leq \epsilon + |\dot{w}| \max\left(8h_T \left(\ln \frac{16|\dot{w}|h_T \exp(\alpha/4h_T^2) \left(\frac{\alpha + \sum_{t=1}^T g_t^2}{\alpha}\right)^{4.5}}{\epsilon} - 1\right), \right. \\ &\quad \left. 2 \sqrt{\sum_{t=1}^T g_t^2 \ln\left(\frac{4 \left(\sum_{t=1}^T g_t^2\right)^{10} \exp(\alpha/2G_T^2) \dot{w}^2}{\epsilon^2} + 1\right)}\right) \end{aligned}$$

■

Appendix C. 1D ONS with shrinking domains

Essentially all of the analysis here is identical to the classical procedure (e.g. see (Hazan et al., 2007)), but we recall it here to verify that the shrinking domains have little effect.

Theorem 6 For any $\hat{v} \in S_T$,

$$\sum_{t=1}^T \left(z_t(v_t - \hat{v}) - \frac{\beta}{2} [z_t(v_t - \hat{v})]^2 \right) \leq \frac{\beta\tau}{2} \hat{v}^2 + \frac{2}{\beta} \sum_{t=1}^T \frac{z_t^2}{A_t}.$$

Proof Define $x_{t+1} = v_t - \frac{z_t}{\beta A_t}$ so that $v_{t+1} = \Pi_{S_{t+1}}(x_{t+1})$ for $t < T$. We make the definition $v_{T+1} = x_{T+1}$ for ease of analysis later. Then, we have

$$x_{t+1} - \hat{v} = v_t - \hat{v} - \frac{z_t}{\beta A_t},$$

that implies

$$A_t(x_{t+1} - \dot{v}) = A_t(v_t - \dot{v} - \frac{z_t}{\beta A_t}) = A_t(v_t - \dot{v}) - \frac{1}{\beta} z_t,$$

and

$$\begin{aligned} A_t(x_{t+1} - \dot{v})^2 &= (A_t(v_t - \dot{v}) - \frac{1}{\beta} z_t)(x_{t+1} - \dot{v}) \\ &= A_t(v_t - \dot{v})(x_{t+1} - \dot{v}) - \frac{1}{\beta} z_t(x_{t+1} - \dot{v}) \\ &= A_t(v_t - \dot{v})(x_{t+1} - \dot{v}) - \frac{1}{\beta} z_t(v_t - \dot{v} - \frac{z_t}{\beta A_t}) \\ &= A_t(v_t - \dot{v})(x_{t+1} - \dot{v}) - \frac{1}{\beta} z_t(v_t - \dot{v}) + \frac{z_t^2}{\beta^2 A_t} \\ &= A_t(v_t - \dot{v})(v_t - \dot{v} - \frac{z_t}{\beta A_t}) - \frac{1}{\beta} z_t(v_t - \dot{v}) + \frac{z_t^2}{\beta^2 A_t} \\ &= A_t(v_t - \dot{v})^2 - \frac{2}{\beta} (v_t - \dot{v}) z_t + \frac{z_t^2}{\beta^2 A_t} \end{aligned}$$

We now use the definition of $\Pi_{S_{t+1}}$, and the assumption that $\dot{v} \in S_T \subset S_{t+1}$ to have:

$$(x_{t+1} - \dot{v})^2 \geq (v_{t+1} - \dot{v})^2$$

from which we conclude:

$$z_t(v_t - \dot{v}) \leq \frac{\beta A_t}{2} (v_t - \dot{v})^2 - \frac{\beta}{2} A_t (v_{t+1} - \dot{v})^2 + \frac{2z_t^2}{\beta A_t}$$

Summing over $t = 1, \dots, T$, we have

$$\begin{aligned} \sum_{t=1}^T z_t(v_t - \dot{v}) &\leq \frac{\beta}{2} A_1 (v_1 - \dot{v})^2 + \frac{\beta}{2} \sum_{t=2}^T (A_t - A_{t-1}) (v_t - \dot{v})^2 \\ &\quad - \frac{\beta (v_{T+1} - \dot{v})^2}{2A_T} + \sum_{t=1}^T \frac{2z_t^2}{\beta A_t} \\ &\leq \frac{\beta}{2} A_1 (v_1 - \dot{v})^2 + \frac{\beta}{2} \sum_{t=2}^T z_t^2 (v_t - \dot{v})^2 + \sum_{t=1}^T \frac{2z_t^2}{\beta A_t} \\ &= \frac{\beta}{2} \tau \dot{v}^2 + \frac{\beta}{2} \sum_{t=1}^T [z_t(v_t - \dot{v})]^2 + \sum_{t=1}^T \frac{2z_t^2}{\beta A_t} \end{aligned}$$

■

Next we need to bound the sum $\sum_{t=1}^T \frac{2z_t^2}{\beta A_t}$, which is easy thanks to the concavity of log:

Lemma 7

$$\sum_{t=1}^T \frac{z_t^2}{A_t} \leq \log \left(1 + \frac{\sum_{t=1}^T z_t^2}{\tau} \right)$$

Proof Since $\log(x)$ is concave and $\frac{d}{dx} \log(x) = \frac{1}{x}$, we have $\log(a+b) - \log(a) \geq \frac{b}{a+b}$. Therefore for any K we have

$$\begin{aligned} \log\left(\frac{\tau + \sum_{t=1}^{K+1} z_t^2}{\tau}\right) - \log\left(\frac{\tau + \sum_{t=1}^K z_t^2}{\tau}\right) &\geq \frac{z_{K+1}^2}{\tau + \sum_{t=1}^{K+1} z_t^2} \\ &= \frac{z_{K+1}^2}{A_{K+1}} \end{aligned}$$

Summing this identity over all $K < T$ proves the result. \blacksquare

Here are three lemmas copied over (with occasional mild modification) from (Cutkosky and Orabona, 2018):

Lemma 8 For $-1 < x \leq 2$, we have

$$\ln(1+x) \leq x - \frac{2 - \ln(3)}{4} x^2.$$

Lemma 9 Define $\ell_t(v) = -\ln(1 - g_t v)$. Let $|\hat{v}|, |v| \leq \frac{1}{2G_t}$ and $|g_t| \leq G_t$. Then

$$\ell_t(v) - \ell_t(\hat{v}) \leq \ell'_t(v)(v - \hat{v}) - \frac{2 - \ln(3)}{2} \frac{1}{2} [\ell'_t(v)(v - \hat{v})]^2.$$

Proof We have

$$\ln(1 - g_t \hat{v}) = \ln(1 - g_t v + g_t(v - \hat{v})) = \ln(1 - g_t v) + \ln\left(1 + \frac{g_t(v - \hat{v})}{1 - g_t v}\right).$$

Now, observe that since $1 - g_t \hat{v} \geq 0$ and $1 - g_t v \geq 0$, $1 + \frac{g_t(v - \hat{v})}{1 - g_t v} \geq 0$ as well so that $\frac{g_t(v - \hat{v})}{1 - g_t v} \geq -1$. Further, since $|\hat{v} - v| \leq \frac{1}{G_t}$ and $1 - g_t v \geq 1/2$, $\frac{g_t(v - \hat{v})}{1 - g_t v} \leq 2$. Therefore, by Lemma 8 we have

$$\ln(1 - g_t \hat{v}) \leq \ln(1 - g_t v) + \frac{g_t(v - \hat{v})}{1 - g_t v} - \frac{2 - \ln(3)}{4} \frac{[g_t(v - \hat{v})]^2}{(1 - g_t v)^2}.$$

Using the fact that $\ell'_t(v) = \frac{g_t}{1 - g_t v}$ finishes the proof. \blacksquare

Lemma 10 Define $\ell_t(v) : [-1/2h_t, 1/2h_t] \rightarrow \mathbb{R}$ as $\ell_t(v) = -\ln(1 - g_t v)$, where $|g_t| \leq h_t$. If we run ONS in Algorithm 5 with $\beta = \frac{2 - \ln(3)}{2}$, $\tau = 4\alpha$, and $S_t = [-1/2h_t, 1/2h_t]$, then for all $\hat{v} \in S_T$,

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq \frac{\alpha}{4h_T^2} + 4.5 \log\left(\frac{\alpha + \sum_{t=1}^T g_t^2}{\alpha}\right)$$

Proof By Lemma 9, we have:

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq \sum_{t=1}^T \ell'_t(v_t)(v_t - \hat{v}) - \frac{2 - \ln(3)}{2} \frac{1}{2} [\ell'_t(v_t)(v_t - \hat{v})]^2$$

Algorithm 6 Adaptive Gradient Descent

- 1: **Initialize:** Unit ball B in some Hilbert space, $w_1 = 0 \in B$, $\lambda = \sqrt{2}$
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Play w_t
 - 4: Receive g_t
 - 5: Set $\eta_t = \frac{\lambda}{\sqrt{\sum_{s=1}^t \|g_s\|^2}}$
 - 6: Set $w_{t+1} = \prod_B(w_t - \eta_t g_t) // \prod_B(x) = \operatorname{argmin}_{y \in B} \|y - x\|$.
 - 7: **end for**
-

Then set $z_t = \ell'_t(v_t)$ and use Theorem 6 to obtain

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq 2\beta\alpha\hat{v}^2 + \frac{2}{\beta} \sum_{t=1}^T \frac{z_t^2}{A_t}.$$

Next, apply Lemma 7:

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq \frac{\beta\alpha}{2h_T^2} + \frac{2}{\beta} \log \left(1 + \frac{\sum_{t=1}^T z_t^2}{4\alpha} \right)$$

Now we observe that $|z_t| = \left| \frac{g_t}{1-g_t v_t} \right| \leq 2|g_t|$ so that $\sum_{t=1}^T z_t^2 \leq 4 \sum_{t=1}^T g_t^2$, yielding

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq \frac{\beta\alpha}{2h_T^2} + \frac{2}{\beta} \log \left(\frac{\alpha + \sum_{t=1}^T g_t^2}{\alpha} \right)$$

Finally, numerically evaluate β to conclude the bound. ■

Lemma 11 (Lemma 19 of (Cutkosky and Orabona, 2018)) *Let $f(x) = a \exp(b \frac{x^2}{|x|+c})$, where $a, b > 0$ and $c \geq 0$. Then*

$$f^*(\theta) \leq |\theta| \max \left(\frac{2}{b} \left(\ln \frac{2|\theta|}{ab} - 1 \right), \sqrt{\frac{c}{b} \ln \left(\frac{c\theta^2}{a^2b} + 1 \right)} - a \right).$$

Appendix D. Adaptive Unit-Ball Algorithm in Hilbert Spaces

Here we briefly recall some classic analysis of adaptive mirror descent algorithms. For simplicity, we only consider the Hilbert space case, rather than a more general smooth Banach space. More details and more generality can be found in (Duchi et al., 2010; McMahan, 2014; Hazan et al., 2008).

Theorem 12 *Algorithm 6 guarantees*

$$R_T(\hat{w}) \leq 2^{3/2} \sqrt{\sum_{t=1}^T \|g_t\|^2}$$

for all $\hat{w} \in B$.

Proof

$$\begin{aligned}
 \|w_{t+1} - \hat{w}\|^2 &\leq \|w_t - \eta_t g_t - \hat{w}\|^2 \\
 &= \|w_t - \hat{w}\|^2 + 2\eta_t \langle g_t, w_t - \hat{w} \rangle + \eta_t^2 \|g_t\|^2 \\
 \langle g_t, w_t - \hat{w} \rangle &\leq \frac{\|w_t - \hat{w}\|^2 - \|w_{t+1} - \hat{w}\|^2}{2\eta_t} + \frac{\eta_t}{2} \|g_t\|^2 \\
 R_T(\hat{w}) &\leq \frac{\|w_1 - \hat{w}\|^2}{2\eta_1} - \frac{\|w_{T+1} - \hat{w}\|^2}{\eta_T} + \sum_{t=2}^T \frac{\|w_t - \hat{w}\|^2}{2} (\eta_t^{-1} - \eta_{t-1}^{-1}) + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\
 &\leq \frac{2}{\eta_T} + \lambda \sqrt{\sum_{t=1}^T \|g_t\|^2} \\
 &\leq \left(\frac{2}{\lambda} + \lambda\right) \sqrt{\sum_{t=1}^T \|g_t\|^2} \\
 &= 2^{3/2} \sqrt{\sum_{t=1}^T \|g_t\|^2}
 \end{aligned}$$

where we have used the identity $\sum_{t=1}^T \frac{\|g_t\|^2}{\sqrt{\sum_{i=1}^T \|g_i\|^2}} \leq 2\sqrt{\sum_{t=1}^T \|g_t\|^2}$, which holds by concavity of the square root. ■