Parameter-free, Dynamic, and Strongly-Adaptive Online Learning

Ashok Cutkosky [1 2]

Abstract

We provide a new online learning algorithm that for the first time combines several disparate notions of adaptivity. First, our algorithm obtains a "parameter-free" regret bound that adapts to the norm of the comparator and the squared norm of the size of the gradients it observes. Second, it obtains a "strongly-adaptive" regret bound, so that for any given interval of length N, the regret over the interval is $\tilde{O}(\sqrt{N})$. Finally, our algorithm obtains an optimal "dynamic" regret bound: for any sequence of comparators with path-length P, our algorithm obtains regret $\tilde{O}(\sqrt{PN})$ over intervals of length N. Our primary technique for achieving these goals is a new method of combining constrained online learning regret bounds that does not rely on an expert meta-algorithm to aggregate learners.

1. Online Learning and Strong-Adaptivity

Online learning is a popular way to analyze iterative optimization algorithms (Shalev-Shwartz, 2011; Zinkevich, 2003). Briefly, online learning is a game played between the learning algorithm and an environment over T rounds. In each round, the learning algorithm outputs a vector w_t in some convex space W and then the environment reveals a loss function ℓ_t and the learner suffers loss $\ell_t(w_t)$. The goal is typically to obtain small *regret*:

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

We can use this game to model the progress of an iterative machine learning training procedure: let w_t be the tth set of model parameters output by the training algorithm and ℓ_t be the loss on the tth minibatch. Then the regret simply measures how fast the training algorithm is able to converge to the optimal parameters $\mathring{w}$. More generally, online learning can be used to model many sequential decision making processes in which the characteristics of the ℓ_t may change from round to round. For example, ℓ_t may represent outcomes varying from the effects of a medical treatment to the score obtained by a game-playing robot.

In order to achieve tractable theoretical guarantees, in this paper we will make the common assumption that each ℓ_t is a convex function. With this assumption, we can set g_t to be a subgradient of ℓ_t at w_t and then obtain

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

All our results will come from upper bounding this linearized regret, so for simplicity in the rest of this paper we assume that each loss ℓ_t is linear and use the right hand side of (1) as the *definition* of the regret. This setting is often called *online linear optimization* (OLO), and many popular and successful algorithms have been designed an analyzed for this setting (Duchi et al., 2010; McMahan & Streeter, 2010; Zinkevich, 2003)

It is well-known that in online linear optimization, one cannot guarantee worst-case regret better than $O(DG\sqrt{T})$, where $G = \max \|g_t\|$ and $D = \sup_{x,y\in W} \|x - y\|$ is the diameter of the domain W, and this bound is achieved by online gradient descent (Abernethy et al., 2008). As a result, much of the work in online linear optimization is in designing *adaptive* algorithms (Hazan et al., 2008; Duchi et al., 2010; McMahan & Streeter, 2010). These algorithms should perform better in non-worst-case settings without sacrificing worst-case guarantees, and typically obtain bounds of the form:

$$R_T(\mathring{w}) \le D\sqrt{\sum_{t=1}^{T} \|g_t\|^2}$$

The above bound achieves adaptivity to non-worst-case sequences of losses g_t, but does not adapt to a non-worst-case comparison point $\mathring{w}$. In order to fix this, so-called "parameter-free" algorithms (Cutkosky & Orabona, 2018; Kempka et al., 2019; Cutkosky & Sarlos, 2019; van der Hoeven, 2019; Mhammedi & Koolen, 2020) obtain smaller

[1]Google Research [2]Boston University, Boston Massachusetts, USA. Correspondence to: Ashok Cutkosky <ashok@cutkosky.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

regret when $\mathring{w}$ is small:

$$R_T(\mathring{w}) \le \|\mathring{w}\|\sqrt{\sum_{t=1}^T \|g_t\|^2}$$

In a parallel direction, other works go beyond the minimax optimality of online gradient descent by considering harder notions of regret. In particular, one attractive goal is to obtain a so-called *strongly-adaptive* guarantee. Strongly-adaptive algorithms obtain:

$$\sup_{\mathring{w}\in W}\sum_{t=a}^b \langle g_t, w_t - \mathring{w}\rangle = \tilde{O}\left(DG\sqrt{b-a}\right)$$

for all intervals $[a, b] \subset [1, T]$ simultaneously. This guarantee was first obtained by (Daniely et al., 2015), and later (Jun et al., 2017) improved the logarithmic dependencies to a regret bound of $O\left(DG\sqrt{T\log(T)}\right)$. Intuitively, these algorithms provide robustness to environments that display some kind of shifting behavior, with losses on different intervals having very different statistics.

Finally, the notion of *dynamic* regret allows the comparison point $\mathring{w}$ to change:

$$R_T(\mathring{w}_1, \dots, \mathring{w}_T) = \sum_{t=1}^T \langle g_t, w_t - \mathring{w}_t\rangle$$

This setting has been studied in several previous works (often under the names "shifting" or "tracking" regret) (Herbster & Warmuth, 1998; Gyorgy et al., 2012; Gyorgy & Szepesvári, 2016; Zhang et al., 2018a), and there are several notions of performance in this setting (see Zhang et al. (2018b) for a description of many of them), but in this paper we focus on obtaining a regret bound in terms of the *path-length* $P = \sum_{t=1}^{T-1}\|\mathring{w}_{t+1} - \mathring{w}_t\|$. Online gradient descent already obtains regret $D\sqrt{T} + P\sqrt{T}$, but this can be improved to $\sqrt{DPT}$ (and no further) without any knowledge of P, as shown by Zhang et al. (2018a).

Given these disparate threads of research, it is natural to look for an approach that can obtain all three goals in a single algorithm. This is the main result of our paper: given any norm $\|\cdot\|$, we provide an algorithm such that for any interval $[a, b]$ the dynamic regret over that interval is:

$$\sum_{t=a}^b \langle g_t, w_t - \mathring{w}_t\rangle \le$$
$$O\left(\sqrt{\left(D^2 + D\sum_{t=a}^{b-1}\|\mathring{w}_{t+1} - \mathring{w}_t\|\right)\sum_{t=1}^T \|g_t\|^2\log(T)}\right)$$

Moreover, for the interval $[1, T]$ our algorithm maintains a non-dynamic "parameter-free" regret bound:

$$O\left(\|\mathring{w}\|\sqrt{\sum_{t=1}^T \|g_t\|^2 \log(\|\mathring{w}\|\sum_{t=1}^T\|g_t\|^2)}\right)$$

Note that in all cases, our bounds achieve adaptivity to the value of $\|g_t\|^2$, which we call a *second-order bound*, and so automatically perform better when the losses are less adversarial.

Previous algorithms obtain parts of this goal. For example, the algorithm of (Zhang et al., 2018a) obtains optimal dynamic regret with a second-order bound, but only for the interval $[a, b] = [1, T]$ (it is not strongly-adaptive). On the other hand, (Zhang et al., 2019a) does not consider the dynamic-regret setting, but obtains a strongly-adaptive algorithm with regret $D\sqrt{\max_{\|g_t\|}\sum_{t=a}^b\|g_t\|}$ for any interval $[a, b]$[1], which does not quite achieve a second-order bound. To the best of our knowledge, the only other algorithm to achieve strongly adaptive regret that also obtains a second-order bound is (Zhang et al., 2019b). However, their algorithm runs in $O(\log^2(T))$ time per update, while ours runs in $O(\log(T))$ time, which matches the best-known rate for any strongly-adaptive algorithm.

We are not aware of any algorithms that achieve the parameter-free regret bound while also achieving either strongly-adaptive or dynamic regret bounds. Further, we believe our result is the first to combine strongly-adaptive regret with the optimal dependence on path-length for dynamic regret. Interestingly, it turns out that this latter result is due almost entirely to an improvement in analysis rather than a novel property of our algorithm. We suspect that our same proof shows that essentially all known strongly-adaptive algorithms to-date also achieve the optimal dependence on the path-length in dynamic regret.

In order to obtain these improved results, our techniques are qualitatively somewhat different from previous approaches. The standard paradigm for designing strongly-adaptive methods, as outlined initially by (Daniely et al., 2015), is to instantiate $O(\log(T))$ different online learning algorithms, each dedicated to a different interval in $[1, T]$. The intervals are chosen cleverly so that any $[a, b]$ can be decomposed into a disjoint union of $\log(b - a)$ intervals. Then, a meta-algorithm is used to aggregate the $O(\log(T))$ outputs of the base algorithms in order to create a single method that never performs much worse than any of the base algorithms.

Our approach eschews the use of a meta-algorithm. Instead, we provide a generic technique that takes an online learner and any interval $[a, b]$ and provides a new learner

[1]Their algorithm technically considers the case of non-negative smooth losses, but with a little effort one can see they achieve this stated bound in our setting.

whose regret on any interval $I \supset [a, b]$ is at most an additive constant worse than the regret of the original learner on that interval, and whose regret on the interval $[a, b]$ is at most $\tilde{O}\left(\sqrt{\sum_{t=a}^{b} \|g_t\|^2}\right)$. Intuitively, on the interval $[a, b]$, we run a second online learner whose predictions are added to those of the first learner on that interval. The second learner's objective is in some sense to minimize the "residual" in the loss obtained by the first learner. This new approach is what allows our algorithm to maintain the parameter-free regret bound.

2. Setting, Notation, and General Strategy

We consider the online linear optimization game in which the learner outputs $w_t \in W$ for some convex domain W and then suffers loss $\langle g_t, w_t\rangle$ for some loss vector g_t. We define the dynamic regret over an interval $I = [a, b]$ as

$$R_I(\mathring{\mathbf{w}}) = \sum_{t\in I} \langle g_t, w_t - \mathring{w}_t\rangle$$

where $\mathring{\mathbf{w}}$ denotes the vector of comparison points $\mathring{w}_1, \dots, \mathring{w}_T$. We will occassionally refer to $R_I(\mathring{w})$ without a bold-font $\mathring{\mathbf{w}}$. This indicates a non-dynamic regret in which $\mathring{\mathbf{w}} = (\mathring{w}, \dots, \mathring{w})$. Given an interval $I = [a, b]$, we denote the comparator path-length over the interval as $P_I = \sum_{t=a}^{b-1} \|\mathring{w}_{t+1} - \mathring{w}_t\|$. We also use a compressed-sum notation to reduce clutter in formulas: $\|g\|^2_{a:b}$ indicates $\sum_{t=a}^{b} \|g_t\|^2$.

We assume that the domain W is a convex subset of $\mathbb{R}^d$, and let $\|\cdot\|$ indicate the standard Euclidean norm. Note that all of our presentation is essentially unchanged if W is a subset of a Hilbert space and $\|\cdot\|$ is the Hilbert space norm, but we restrict to $\mathbb{R}^d$ for ease of exposition. We will constrain g_t to satisfy $\|g_t\| \le 1$ for all t. We assume that our domain W is bounded, and use D to indicate the diameter of W with respect to this norm: $D = \sup_{x,y\in W} \|x - y\|$. Finally, we assume that the origin is in W. This assumption is without loss of generality: if the origin is not in W, we can simply choose some arbitrary point $w \in W$ and translate our coordinates to make w the origin.

A key component in our analysis is the existance so-called "parameter-free" algorithms (Mcmahan & Streeter, 2012; Orabona, 2014; Orabona & Pál, 2016; Foster et al., 2018; Cutkosky & Orabona, 2018). These are algorithms that obtain non-dynamic regret

$$R_{[1,T]}(\mathring{w}) \le \tilde{O}\left(\epsilon + \|\mathring{w}\|\sqrt{T\log(T\|\mathring{w}\|/\epsilon)}\right)$$

where $\epsilon > 0$ is an arbitrary user-specified constant. The key advantage of these algorithms is that their regret automatically adjusts to the value of $\|\mathring{w}\|$. In particular, if $\mathring{w} = 0$, the regret is bounded by the constant ϵ. Put another way, the *total loss* of the algorithm, $\sum_{t=1}^{T}\langle g_t, w_t\rangle$ cannot be larger than ϵ. In this paper, we will use algorithms that not only are parameter-free, but also achieve a second-order regret bound. Specifically, we have the following consequence of results in (Cutkosky & Sarlos, 2019; Cutkosky & Orabona, 2018):

Theorem 1. *For any convex set W, and any $\psi > 0$, there exists an algorithm that obtains (non-dynamic) regret*

$$\begin{aligned} R_{[1,T]}(\mathring{w}) &= \sum_{t=1}^{T}\langle g_t, w_t - \mathring{w}\rangle \\ &\le \psi + C\|\mathring{w}\|\sqrt{(1 + \|g_t\|^2_{1:T})\log\left(\frac{\|\mathring{w}\|(1 + \|g_t\|^2_{1:T})}{\psi}\right)} \\ &\quad + C\|\mathring{w}\|\log\left(\frac{\|\mathring{w}\|(1 + \|g_t\|^2_{1:T})}{\psi}\right) \end{aligned}$$

for some absolute constant C, where $\|g_t\|^2_{1:T}$ is notation for $\sum_{t=1}^{T}\|g_t\|^2$, so long as $\|g_t\| \le 1$ for all T

2.1. Overview of Approach

Our general approach bears many similarities to the pioneering work of (Daniely et al., 2015). We will construct a set of "representative intervals" S such that any interval $[a, b]$ can be written as a disjoint union of at most $\log_2(b-a)$ intervals in S. Then we will construct an algorithm such that, for each interval $I \in S$, the regret over the interval is $O(\sqrt{|I|})$, where $|I|$ is the length of the interval. Since each general interval $[a, b]$ can be constructed using a small number of intervals in S, this will be sufficient to show strongly-adaptive regret. The run-time of our algorithm will be proportional to the number of intervals in S, and so it is important that S be reasonably small. As outlined in Section 5, we use the geometric covering intervals suggested by (Daniely et al., 2015), which leads to $|S| = O(\log(T))$.

In order to build an algorithm that has low regret for each interval $I \in S$, we provide a generic procedure that takes an arbitrary algorithm and an interval I and produces a new algorithm whose regret on I is $O(\sqrt{|I|})$ without changing the regret on any other interval. By applying this construction for each interval in S, we develop our desired algorithm. We outline how to achieve this in Section 3. Our main Theorem is then presented in Theorem 6. Finally, we show that this approach to strongly-adaptive regret also implies optimal dynamic regret in Theorem 7.

3. Residual Learning

In this section we show how to take an algorithm $\mathcal{A}$ and an interval $[a, b]$ and produce a new algorithm that never has worse regret than $\mathcal{A}$, but is also guaranteed to have low regret on the given interval. To gain some intuition for how our technique works, let us for the moment ignore the constraint set W and allow our algorithm to be improper, in the sense that its outputs w_t may lie outside W. Then, one first idea is to have a second learner $\mathcal{B}$ run only on the interval $[a, b]$ on the losses $\ell_t(w) = \langle g_t, w - x_t \rangle$, where x_t is the tth output of $\mathcal{A}$. In some sense, $\mathcal{B}$ is attempting to minimize the residual error left by the first learner. If y_t is the output of $\mathcal{B}$, we might think to play $w_t = x_t + y_t$. Intuitively, if x_t is doing poorly, y_t can pick up the slack and vice versa. Unfortunately, this technique does not quite work because now $\mathcal{B}$'s comparison points are essentially $\mathring{w}_t + x_t$, which can have a much larger path-length than the original $\mathring{w}_t$. Note that in the case of non-dynamic regret and the fixed interval $[1, T]$, (Cutkosky, 2019) avoided this problem by assuming that $\mathcal{A}$ has a reasonable regret bound, which implies that the x_t have some nice properties. However, in our case we do not wish to assume anything about the loss of $\mathcal{A}$ on the interval $[a, b]$, and so we need a more delicate approach.

Our key idea to fix the problem with adding iterates is to introduce an extra dimension. $\mathcal{B}$ will output points in $W \times [0, 1] \subset \mathbb{R}^{d+1}$, and its loss at time point t is given by:

$$\ell_t(y, z) = \langle g_t, y \rangle - z \langle g_t, x_t \rangle$$

Notice that $\ell_t(y, z)$ is convex (in fact, linear) in (y, z). Then at time t, if (y_t, z_t) is the output of the second learner, we play the point $w_t = x_t + y_t - z_t x_t$. Using this method, ℓ_t has the following powerful properties:

$$\langle g_t, w_t - \mathring{w}_t \rangle = \ell_t(y_t, z_t) - \ell_t(\mathring{w}_t, 1) \quad (2)$$

$$\langle g_t, w_t - \mathring{w}_t \rangle = \langle g_t, x_t - \mathring{w} \rangle + \ell_t(y_t, z_t) - \ell_t(0, 0) \quad (3)$$

The first fact implies that the total regret over the interval $[a, b]$ is equal to the regret of $\mathcal{B}$ on its losses ℓ_t, which is under control as long as $\mathcal{B}$ obtains a good regret guarantee. The second fact implies that the regret over the interval $[a, b]$ is equal to the regret of $\mathcal{A}$ on the same interval plus the regret of $\mathcal{B}$ to the constant comparison point $(0, 0)$. This is where we invoke the existence of parameter-free algorithms. By setting $\mathcal{B}$ to be the algorithm specified by Theorem 1 that obtains constant regret with respect to the origin, we control this extra regret and so experience very little extra loss over the regret of $\mathcal{A}$.

Note that this intuition does not deal with the issue of enforcing $w_t \in W$. To address this issue, we take a brief detour to discuss a general method for applying contraints to online learning algorithms.

4. Adding Constraints

In order to remove the improperness from the approach outlined in the previous section, we need some way to enforce the constraint $w_t \in W$. Our strategy for this is a generalization of the reduction between unconstrained and constrained online learning proposed by (Cutkosky & Orabona, 2018). Their method takes the prediction w_t of an unconstrained algorithm and outputs the constrained prediction $\Pi(w_t) = \text{argmin}_{w \in W} \|w - w_t\|$. Then the original algorithm is provided with the surrogate loss function $\ell_t(w) = \langle g_t, w \rangle + \|g_t\| S(w)$, where $S(x) = \|x - \Pi(w)\|$. The additional term $\|g_t\| S(w)$ serves as a penalty for attempting to violate the constraints. This penalty is carefully designed in such a way that the regret of the unconstrained algorithm on the penalized losses is an upper bound on the regret of the constrained points on the true losses. We make two mild improvements upon their reduction. First, we design a better penalty term that allows us to remove an extra factor of two from the original regret analysis. Second, we observe that it is possible to *change the constraint set* at every round, so long as the comparison points $\mathring{w}_t$ always lie with in tth constraint. The algorithm and analysis are present in Algorithm 1 and Theorem 2.

Algorithm 1 Varying Constraints

Input: online learning algorithm $\mathcal{A}$. Sequence of domains $V_1, \dots, V_T$ contained $\mathbb{R}^d$.
for $t = 1 \dots T$ **do**
  Get $w_t \in V$ from $\mathcal{A}$.
  Define the function $\Pi_t(x) = \text{argmin}_{w \in V_t} \|w - x\|$.
  Output $\hat{w}_t = \Pi_t(w_t)$.
  Get loss g_t.
  Let $v_t = \frac{w_t - \hat{w}_t}{\|w_t - \hat{w}_t\|}$.
  Define the function $S_t(w) = \|w - \Pi_t(w)\|$.
  Define $\hat{\ell}_t(w)$ by:

$$\begin{cases} \langle g_t, w \rangle & \text{if } \langle g_t, w_t \rangle \geq \langle g_t, \hat{w}_t \rangle \\ \langle g_t, w \rangle - \langle g_t, v_t \rangle S_t(w) & \text{if } \langle g_t, w_t \rangle < \langle g_t, \hat{w}_t \rangle \end{cases}$$

  Compute $\hat{g}_t \in \partial \hat{\ell}_t(w_t)$.
  Send $\hat{g}_t$ to $\mathcal{A}$ as tth loss.
end for

Theorem 2. *The functions $\hat{\ell}_t$ defined in Algorithm 1 are convex functions defined on all of V and the gradients $\hat{g}_t$ sent to $\mathcal{A}$ by Algorithm 1 satisfy $\|\hat{g}_t\| \leq \|g_t\|$. Also, for all t and all $\mathring{w} \in V_t$ we have*

$$\langle g_t, \hat{w}_t - \mathring{w} \rangle \leq \hat{\ell}_t(w_t) - \hat{\ell}_t(\mathring{w}) \leq \langle \hat{g}_t, w_t - \mathring{w} \rangle$$

Proof. First, note depending on the values of w_t, $\hat{w}_t$ and g_t, $\hat{\ell}_t$ is either $\langle g_t, w \rangle$ for all w, or $\langle g_t, w \rangle - \langle g_t, v_t \rangle S(w)$ for all w. In the former case, we have $\hat{g}_t = g_t$ and so all claims in the statement are immediate. Let us focus on the latter case. We have from (Cutkosky & Orabona, 2018) Proposition 1 that $S_t(w)$ is a 1-Lipschitz convex function, so that to show that $\hat{\ell}_t$ is convex it suffices to show that $\langle g_t, v_t \rangle \leq 0$. To this end, observe that we can write $w_t = \hat{w}_t + \alpha v_t$ where $\alpha =$

$\|w_t - \mathring{w}_t\| = S(w_t) \geq 0$. Therefore, if $\langle g_t, w_t\rangle < \langle g_t, \mathring{w}_t\rangle$, we must have $\langle g_t, v_t\rangle < 0$ so that $\hat{\ell}_t$ is convex. Further, we have $\partial S_t(w_t) = \{v_t\}$ by Theorem 4 of (Cutkosky & Orabona, 2018). Therefore $\hat{g}_t = g_t - \langle g_t, v_t\rangle v_t$. Since $\|v_t\| = 1$, by triangle inequality we have $\|\hat{g}_t\| \leq 2\|g_t\|$. Then wse have that $\hat{g}_t$ is orthogonal projection of g_t onto subspace perpendicular to v_t, so that $\|\hat{g}_t\| \leq \|g_t\|$.

Next, notice that if $\langle \hat{g}_t, w_t\rangle < \langle g_t, \mathring{w}_t\rangle$, we have

$$\begin{aligned}\langle g_t, \mathring{w}_t\rangle &= \langle g_t, w_t\rangle + \langle g_t, \mathring{w}_t - w_t\rangle\\ &\langle g_t, w_t\rangle - \alpha\langle g_t, v_t\rangle\\ &= \hat{\ell}_t(w_t)\end{aligned}$$

This implies $\langle g_t, \mathring{w}_t - \mathring{w}\rangle = \hat{\ell}_t(w_t) - \hat{\ell}_t(\mathring{w}) \leq \langle \hat{g}_t, w_t - \mathring{w}\rangle$ as desired. □

Theorem 2 allows us to add arbitrary time-varying constraints to an online learning algorithm without damaging regret bounds, even if those bounds happen to depend on the values of $\|g_t\|$. Using this, we can fix the approach outlined in Section 3. Recall that given x_t, we have a second algorithm $\mathcal{B}$ that outputs (y_t, z_t) in response to losses

$$\ell_t(y, z) = \langle g_t, y\rangle - \langle g_t, x_t\rangle z$$

and our final output is $w_t = x_t + y_t - z_t x_t$. In order to guarantee that $w_t \in W$, we use the fact that $x_t \in W$ and define the set

$$V_t = \{(y, z) :\ x_t + y - z x_t \in W,\ z \in [0, 1], y \in W\}$$

Notice that V_t is a convex set, and $(0, 0)$ and $(w, 1)$ are both in V_t for all values of $w \in W$. Therefore we can simply constrain $\mathcal{B}$ to play within the set V_t, and by definition we will have $w_t = x_t + y_t - z_t x_t \in W$. Moreover, since $(0, 0)$ and $(\mathring{w}_t, 1)$ are both in V_t, we can still make use of the important properties (2) and (3). The full algorithm is described in Algorithm 2 and the analysis is in Theorem 3 below:

Theorem 3. *Let $\mathcal{A}$ be an online learner that outputs $x_t \in W$ in response to losses g_t. Let $I_1, \dots, I_K$ be some disjoint intervals in $[1, T]$. Let J be any interval such that for all k, either J contains I_k or is disjoint from I_k. Then given $\epsilon > 0$, Algorithm 2 guarantees regret:*

$$R_J(\mathring{\mathbf{w}}) = \sum_{t\in J} \langle g_t, w_t - \mathring{w}_t\rangle \leq R_J^{\mathcal{A}}(\mathring{\mathbf{w}}) + \epsilon$$

where $R_J^{\mathcal{A}}(\mathring{\mathbf{w}}) = \sum_{t\in J}\langle g_t, x_t - \mathring{w}_t\rangle$ is the regret of $\mathcal{A}$ on the interval J. Further, for each interval I_k, we have $R_{I_k}(\mathring{\mathbf{w}})$ bounded by:

$$\epsilon + O\left[(D + P_{I_k})\sqrt{\sum_{t\in I_k}\|g_t\|^2 \log\left(\frac{KD\sum_{t\in I_k}\|g_t\|^2}{\epsilon}\right)}\right]$$

where $D = \sup_{x,y\in W}\|x - y\|$ and $P_{[a,b]} = \sum_{t=a}^{b-1}|\mathring{w}_{t+1} - \mathring{w}_t\|$.

Algorithm 2 Residual Algorithm

Input: Intervals $I_1, \dots, I_K$, online learner $\mathcal{A}$, number $\epsilon > 0$.
Initialize algorithm $\mathcal{B}$ from Algorithm 1 using the algorithm of Theorem 1 as the base unconstrained algorithm, with $\psi = \epsilon/K$.
for $t = 1 \dots T$ **do**
  Get $x_t \in W$ from $\mathcal{A}$.
  if $t \notin I_k$ for some k **then**
    Output $w_t = x_t$.
    Get g_t and send it to $\mathcal{A}$.
  else
    If t is at the beginning of some I_k, reinitialize $\mathcal{B}$.
    Let k be such that $t \in I_k$, with $I_k = [a, b]$.
    Set $t' = t - a + 1$.
    Set $V_{t'} = \{(y, z) \in \mathbb{R}^d \times \mathbb{R} : x_t + y - z x_t \in W \text{ and } z \in [0, 1]\}$.
    Send $V_{t'}$ to $\mathcal{B}$ as t'th constraint set.
    Set (y_t, z_t) to the t'th output of $\mathcal{B}$.
    Output $w_t = x_t + y_{t'} - z_{t'} x_t \in W$.
    Get loss g_t.
    Send g_t to $\mathcal{A}$ as tth loss.
    Send $(g_t, -\langle g_t, x_t\rangle)$ to $\mathcal{B}$ as t'th loss.
  end if
end for

Proof. For the first statement, we break up the interval J into disjoint components that are either equal to some I_k, or have no intersection with any I_k. Observe that the total regret over J is just the sum of the regrets over these components. For any component C that has no intersection with I_k, we have $w_t = x_t$, where x_t is the output of $\mathcal{A}$. Therefore the regret obtained over the rounds that do not lie in any I_k is identical to the regret of $\mathcal{A}$ over these rounds.

Next, we compute the regret over any interval I_k. Let us write $I_k = [a, b]$. Observe that in I_k, $w_t = x_t + y_t - z_t x_t$ where x_t is the output of $\mathcal{A}$ and (y_t, z_t) is the $t - a + 1$st the output of $\mathcal{B}$ when running over I_k. Therefore we have

$$\begin{aligned}R_{I_k}(\mathring{\mathbf{w}}) &= \sum_{t\in I_k}\langle g_t, x_t + y_t - z_t x_t - \mathring{w}_t\rangle\\ &= \sum_{t\in I_k}\langle g_t, x_t - \mathring{w}_t\rangle + \sum_{t\in I_k}\langle g_t, y_t - z_t x_t\rangle\\ &= R_{I_k}^{\mathcal{A}}(\mathring{\mathbf{w}}) + \sum_{t=a}^{b}\langle g_t, y_t - 0\rangle + \langle -g_t, x_t\rangle(z_t - 0)\rangle\end{aligned}$$

Now, notice that $\sum_{t=a}^{b}\langle g_t, y_t - 0\rangle + \langle -g_t, x_t\rangle(z_t - 0)\rangle$ is the (non-dynamic) regret of $\mathcal{B}$ with respect to $(0, 0)$. Then, for all t, $(0, 0) \in V_t$ so that by Theorem 1 and Theorem

2, we have that the regret is at most ϵ/K. Therefore since there are only K total intervals I_k, summing up these regret bounds proves the first statement of the Theorem.

For the second statement, we again write $I_k = [a, b]$ and compute:

$$R_{I_k}(\mathring{\mathbf{w}}) = \sum_{t=a}^{b} \langle g_t, x_t + y_t - z_t x_t - \mathring{w}_t \rangle$$
$$= \sum_{t=a}^{b} \langle g_t, y_t - \mathring{w}_t \rangle + \langle -g_t, x_t \rangle (z_t - 1)$$

Therefore the regret is the dynamic regret of $\mathcal{B}$ with respect to the sequence of comparators $(\mathring{w}_a, 1), \dots, (\mathring{w}_b, 1)$. We analyze this dynamic regret separately in Theorem 4 below, from which the result follows. □

Theorem 4. *Suppose $V_1, \dots, V_T$ is a sequence of convex domains, all of which have bounded diameter D. Suppose $\mathring{w}_1, \dots, \mathring{w}_T$ is a sequence of points such that $\mathring{w}_t \in V_t$ for all t. Then if we run Algorithm 1 with base algorithm $\mathcal{A}$ equal to the algorithm of Theorem 1, the combined method achieves dynamic regret:*

$$\sum_{t=1}^{T} \langle g_t, \hat{w}_t - \mathring{w}_t \rangle$$
$$\le \epsilon + C\|\mathring{w}_T\| \sqrt{G \log\left(\frac{\|\mathring{w}_T\|G}{\epsilon}\right)} + C\|\mathring{w}\| \log\left(\frac{\|\mathring{w}_T\|G}{\epsilon}\right)$$
$$+ P_T + P_T C \sqrt{G \log\left((tD\epsilon^{-1} + 1)G\right)}$$
$$+ P_T C \log\left((tD\epsilon^{-1} + 1)G\right)$$
$$= O\Big[(\|\mathring{w}_T\| + P_T)\sqrt{G \log\left((tD\epsilon^{-1} + 1)G\right)}$$
$$+ (\|w_T\| + P_T) \log\left((tD\epsilon^{-1} + 1)G\right)\Big]$$

where $G = 1 + \sum_{t=1}^{T} \|g_t\|^2$, $P_T = \sum_{t=1}^{T-1} \|\mathring{w}_{t+1} - \mathring{w}_t\|$, and we follow the notation in the pseudocode to use $\hat{w}_t$ to indicate outputs of the algorithm.

Proof. Following the notation of Algorithm 1, let w_t be the ouputs of the unconstrained algorithm $\mathcal{A}$, and define $R^{\mathcal{A}}_{[1,T]}$ to be the regret of $\mathcal{A}$. Then we have

$$\sum_{t=1}^{T} \langle g_t, \hat{w}_t - \mathring{w}_t \rangle \le \sum_{t=1}^{T} \langle \hat{g}_t, w_t - \mathring{w}_t \rangle$$
$$= \sum_{t=1}^{T} \langle \hat{g}_t, w_t - \mathring{w}_T \rangle + \sum_{i=1}^{T-1} \sum_{t=1}^{T-i} \langle \hat{g}_t, -(\mathring{w}_{T-i} - \mathring{w}_{T-i+1}) \rangle$$
$$= R^{\mathcal{A}}_{[1,T]}(\mathring{w}_T) + P_T \max_t \left\| \sum_{i=1}^{t} \hat{g}_i \right\|$$

Next, we show that $\langle \hat{g}_t, w_t \rangle \ge \langle \hat{g}_t, \hat{w}_t \rangle$. There are two cases, either $\langle g_t, w_t \rangle \ge \langle g_t, \hat{w}_t \rangle$ or not. In the former case, we have $g_t = \hat{g}_t$ and so the statement follows. In the latter case, we write:

$$\langle \hat{g}_t, w_t - \hat{w}_t \rangle \ge \hat{\ell}_t(w_t) - \hat{\ell}_t(\hat{w}_t)$$
$$= \langle g_t, w_t \rangle - \langle g_t, v_t \rangle \|w_t - \hat{w}_t\| - \langle g_t, \hat{w}_t \rangle$$
$$\langle \hat{g}_t, w_t \rangle \ge \langle \hat{g}_t, \hat{w}_t \rangle + \langle g_t, w_t - \hat{w}_t \rangle - \langle g_t, v_t \rangle \|w_t - \hat{w}_t\|$$
$$\langle \hat{g}_t, w_t \rangle \ge \langle \hat{g}_t, \hat{w}_t \rangle + \langle g_t, w_t - \hat{w}_t \rangle - \left\langle g_t, \frac{(w_t - \hat{w}_t)}{\|w_t - \hat{w}_t\|} \right\rangle \|w_t - \hat{w}_t\|$$
$$\langle \hat{g}_t, w_t \rangle \ge \langle \hat{g}_t, \hat{w}_t \rangle$$

so that the statement still follows.

Now we have for any $X \ge 0$,

$$-Dt + X \left\| \sum_{i=1}^{t} \hat{g}_i \right\| \le \sum_{i=1}^{t} \langle \hat{g}_i, \hat{w}_i \rangle + X \left\| \sum_{i=1}^{t} \hat{g}_i \right\|$$
$$\le \sum_{i=1}^{t} \langle \hat{g}_i, w_i \rangle + X \left\| \sum_{i=1}^{t} \hat{g}_i \right\|$$
$$\le R^{\mathcal{A}}_{[1,T]}\left(-X \frac{\hat{g}_{1:T}}{\|\hat{g}_{1:T}\|}\right)$$
$$\le \epsilon + XC\sqrt{G \log\left(\frac{XG}{\epsilon}\right)} + CX \log\left(\frac{XG}{\epsilon}\right)$$
$$\left\| \sum_{i=1}^{t} \hat{g}_i \right\| \le \inf_X \frac{\epsilon + Dt}{X} + C\sqrt{G \log\left(\frac{XG}{\epsilon}\right)} + C \log\left(\frac{XG}{\epsilon}\right)$$

Set $X = \epsilon + Dt$ to obtain:

$$\left\| \sum_{i=1}^{t} \hat{g}_i \right\| \le 1 + C\sqrt{G \log\left((tD\epsilon^{-1} + 1)G\right)} + C \log\left((tD\epsilon^{-1} + 1)G\right)$$

And now put all these calculations together to prove the Theorem. □

5. Main Result

In this section we prove our main result by showing how to apply Theorem 3 to build a strongly-adaptive algorithm. Our approach uses the *geometric covering intervals* suggested by (Daniely et al., 2015). Let N be the smallest integer such that $T \le 2^N$. Note that $N = O(\log(T))$. For each $i = 0, 1, \dots, N$, we maintain a set of disjoint intervals S_i, such that $\bigcup_{I \in S_i} I = [1, T]$. The set S_i consists of all intervals of length 2^i starting at a multiple of 2^i. Notice that

S_i contains at most $O(T/2^i)$ intervals. Also, observe that any given index t is contained within at most one interval in each S_i. We write $S = \bigcup_{i\le N} S_i$

The key property of these intervals is the following, proved in Lemma 5 of (Daniely et al., 2015):

Proposition 5. *Let $[a, b]$ be any interval contained in $[1, T]$. Then $[a, b]$ can be written as a disjoint union of at most $O(\log_2(b-a))$ intervals such that each interval is in some S_i and no S_i contributes more than 2 intervals to the disjoint union.*

Using this Proposition, we build our algorithm in stages. Specifically, we will construct a sequence of algorithms $\mathcal{A}_N, \mathcal{A}_{N-1}, \dots, \mathcal{A}_1$ such that $\mathcal{A}_N$ is the algorithm provided by Theorem 4, and $\mathcal{A}_1$ is an algorithm that obtains the desired regret guarantees. Formally, we have the following Theorem:

Theorem 6. *Let τ be the time required to project to sets of the form V_t as defined in Algorithm 3. Then there exists an algorithm that runs in $O(d\log(T)\tau)$ time per round such that the dynamic regret $R_I(\mathring{\mathbf{w}})$ over any interval I is:*

$$O\left[(D+P_I)\log\left(T\sum_{t\in I}\|g_t\|^2\right)\sqrt{1+\sum_{t\in I}\|g_t\|^2}\right]$$

Moreover, the same algorithm achieves non-dynamic regret:

$$R_I(\mathring{w}) \le O\left[D(\log^2(T)+\sqrt{|I|\log(T)})\right]$$

where $|I|$ is the length of the interval I. Finally, the same algorithm also achieves non-dynamic regret bounding $R_{[1,T]}(\mathring{w})$ by:

$$O\left[\log(T)+\|\mathring{w}\|\sqrt{\sum_{t=1}^T\|g_t\|^2\log\left\|\mathring{w}\|\sum_{t=1}^T\|g_t\|^2\right)}\right]$$

Proof. Our first goal is to build an algorithm such that for any interval $J\in S$, we have the slightly better regret bound:

$$R_J(\mathring{\mathbf{w}}) \le O\left[(D+P_J)\left(\log\left(T+T\sum_{t\in J}\|g_t\|^2\right)\right.\right. \tag{4}$$
$$\left.\left.+\sqrt{(1+\sum_{t\in J}\|g_t\|^2)\log\left(T+T\sum_{t\in J}\|g_t\|^2\right)}\right)\right]$$

Suppose we have this result. Then recall that any inverval I can be written as a disjoint union of $O(\log(I))$ intervals in S. Further, the regret over the entire interval I is obtained by summing the regret obtained on each interval in the disjoint union. Then applying the Cauchy-Schwarz inequality yields the first statement of the Theorem. For the second statement, notice that since $\|g_t\| \le 1$, we must also have for any interval $J\in S$,

$$R_J(\mathring{\mathbf{w}}) \le O\left[(D+P_J)\left(\log(T|J|)+\sqrt{|J|\log(T|J|)}\right)\right]$$

Given any interval I, we write I as the disjoint union of intervals specified by Proposition 5 and sum the regret over the intervals, just as we did to obtain the first statement. However, now that we have abandoned dependence on $\|g_t\|$, we can bound the sum more efficiently than with Cauchy-Schwarz. Specifically, since no S_i contributes more than 4 intervals to the disjoint union, the total regret is bounded by:

$$O\left[4\sum_{k=1}^{1+\log_2(I)}(D+P_I)\left(\log\left(T2^k\right)+\sqrt{2^k\log\left(T2^k\right)}\right)\right]$$

which yields the desired expression. Therefore, it suffices to design an algorithm that achieves (4) for any $J\in S$.

Our construction builds a sequence of algorithms $\mathcal{A}_N, \dots, \mathcal{A}_1$ in such a way that $\mathcal{A}_i$ will satisfy (4) for any interval J in S_j for $j\ge i$. Thus $\mathcal{A}_1$ will satisfy the regret bound for any $J\in S$. To start, we set $\mathcal{A}_N$ to be the algorithm posited by Theorem 4, with each V_t set to be W and $\psi=1$. Now we build $\mathcal{A}_i$ from $\mathcal{A}_{i+1}$ inductively. Observe that every interval in S_i either disjoint from or completely contained within any interval in S_j for $j\ge i$. Therefore we apply the construction of Theorem 3 to $\mathcal{A}_{i-1}$ with the intervals in S_i and $\epsilon=1$ to obtain $\mathcal{A}_i$. Theorem 3 then implies the bound (4), which we have seen implies the first two claims of the Theorem. Further, since by Theorem 4, $\mathcal{A}_N$ obtains the bound:

$$R_{[1,T]}(\mathring{w}) \le O\left[\|\mathring{w}\|\sqrt{\sum_{t=1}^T\|g_t\|^2\log\left(\|\mathring{w}\|\sum_{t=1}^T\|g_t\|^2\right)}\right]$$

and since $N=\log(T)$, Theorem 3 also implies that $\mathcal{A}_1$ satisfies the last claim of the Theorem as well. $\square$

6. Optimal Dynamic Regret

The bound of Theorem 6 achieves regret *linear* in the path-length of the comparator P_I, but our goal is to obtain $O(\sqrt{P_I})$. This is the optimal rate, as shown by (Zhang et al., 2018a). In this section, we show that in fact a strongly-adaptive guarantee that is linear in P_I actually implies a strongly-adaptive guarantee that depends instead on $\sqrt{P_I}$:

Theorem 7. *The algorithm described by Theorem 6 also achieves for any interval:*

$$R_I(\mathring{\mathbf{w}}) \le \tilde{O}\left[P_I+D+\sqrt{D(P_I+D)\sum_{t\in I}\|g_t\|^2}\right]$$

Further, we have $P_I \le \sqrt{DP_I|I|}$.

Proof. First, we show that it is possible to break the interval I up into disjoint subintervals $I = J_1 \cup \cdots \cup J_K$ such that for each i, $P_{J_i} \le 2D$, and $K \le \frac{P_I + D}{D}$. We do this by explicitly building these subintervals via an iterative greedy construction. Let $I = [a, b]$. Our intervals will satisfy $J_i = [t_{i-1}, t_i]$ where $a = t_0 < t_1 < \cdots < t_K = b$. Let $J_1 = [a, t_1]$ where t_1 is the smallest index such that $P_{J_1} \ge D$. This implies $P_{[a,t_1-1]} < D$, and so therefore $P_{J_1} \le 2D$ since the diameter of W is D. Now, given t_{i-1}, let t_i be the smallest index such that $P_{[t_{i-1},t_i]} \ge D$, and set $J_i = [t_{i-1}, t_i]$. Then again we will have $P_{J_i} \le 2D$. If no such t_i exists, set $i = K$ and $t_i = b$. By this construction, $P_{J_i} \le 2D$ for all i (including $i = K$, for which $P_{J_K} \le D$). Further, we have that $P_{J_i} \ge D$ for all $i < K$ and $\sum_{i=1}^K P_{J_i} \le P_I$. Therefore:

$$P_I \ge \sum_{i=1}^K P_{J_i} \ge (K-1)D$$
$$\frac{P_I + D}{D} \ge K$$

Therefore we have a set of intervals J_i satisfiying the desired properties.

Now, on each of these intervals J_i, we have the regret bound

$$\begin{aligned} R_{J_i}(\mathring{\mathbf{w}}) &\le \tilde{O}\left((D + P_{J_i})\sqrt{1 + \sum_{t\in J_i} \|g_t\|^2}\right) \\ &= \tilde{O}\left(D\sqrt{1 + \sum_{t\in J_i} \|g_t\|^2}\right) \end{aligned}$$

So we sum over all i and use Cauchy-Schwarz:

$$\begin{aligned} R_I(\mathring{\mathbf{w}}) &= \sum_{i=1}^K R_{J_i}(\mathring{\mathbf{w}}) \\ &\le \tilde{O}\left(D\sqrt{K^2 + K\sum_{t\in I} \|g_t\|^2}\right) \\ &\le \tilde{O}\left(P_I + D + \sqrt{D(P_I + D)\sum_{t=1}^T \|g_t\|^2}\right) \end{aligned}$$

where in the last line we used $K \le \frac{P_I + D}{D}$. Finally, observe that $P_I \le D|I|$ to obtain $P_I \le \sqrt{DP_I|I|}$. □

This optimal dependence on the path-length P_I was previous achieved by (Zhang et al., 2018a). Their algorithm only achieves this regret for the entire interval $[1, T]$, while we obtain it for any sub-interval. Both our algorithm and that of (Zhang et al., 2018a) require $O(\log(T))$ time per update. However, we note that our proof of Theorem 7 is rather general, and we believe that in fact the same technique may be used to show that all known strongly-adaptive algorithms to-date also achieve the optimal dynamic regret. Interestingly, however, our Theorem 7 is not a generic reduction showing that all strongly-adaptive algorithms *must* also achieve optimal dynamic regret. Athough (Zhang et al., 2018b) show many interesting reductions between strongly-adaptive algorithms and various other restricted formulations of dynamic regret, to the best of our knowledge the question of whether strongly-adaptive algorithms must optimally adapt to the path-length is still unanswered.

7. Conclusion

We have introduced a new algorithm that combines several desirable notions of adaptivity. First, our algorithm obtains regret

$$R_T(\mathring{w}) \le \tilde{O}\left[\|\mathring{w}\|\sqrt{\sum_{t=1}^T \|g_t\|^2}\right] \quad (5)$$

which is the optimal adaptivity to the norm of $\mathring{w}$ and the gradients g_t. Intuitively, this is the same bound that would be obtained by an optimally tuned online gradient descent, but we do not require any manual tuning.

Secondly, our algorithm obtains strongly-adaptive regret: for any interval $[a, b]$, we have regret

$$\sum_{t=a}^b \langle g_t, w_t - \mathring{w}\rangle \le \tilde{O}\left[D\sqrt{\sum_{t=a}^b \|g_t\|^2}\right]$$

where D is the diameter of W. The only other algorithm we are aware of that achieves strong-adaptivity while also obtaining a bound that depends on $\sum_{t=1}^T \|g_t\|^2$ is that of (Zhang et al., 2019b), which incurs an extra factor of $O(\log(T))$ in the run-time due to running an instance of Metagrad (van Erven & Koolen, 2016) or Maler (Wang et al., 2019) as a subroutine.

Finally, our algorithm obtains optimal dynamic regret over any interval:

$$\sum_{t=a}^b \langle g_t, w_t - \mathring{w}_t\rangle \le \tilde{O}\left[\sqrt{(D^2 + DP)\sum_{t=1}^T \|g_t\|^2}\right]$$

where $P = \sum_{t=a}^{b-1} \|\mathring{w}_t - \mathring{w}_{t+1}\|$. In this case, we believe that our analysis can actually show that *all* known strongly-adaptive algorithms actually achieve optimal dependence on the path length P.

We believe our techniques exhibit a qualitative difference from prior approaches: instead of building our algorithm by using a sleeping-expert meta-algorithm to combine many base algorithms, we show a way to iteratively build up an algorithm with the desired regret bound. One might be able to view this new approach as in some sense bundling the meta-algorithm's operation into each of the individual base algorithms. We feel that our approach has tangible benefits: for example, it is not obvious how to use any prior sleeping-experts-based algorithm to maintain (5) while also obtaining strong-adaptivity.

Several intriguing open questions remain. First, can we replace the dependencies on D in the strongly-adaptive and dynamic regret bounds with some dependence on $\|\mathring{w}\|$ instead? Second, our regret bounds depend on $\log(T)$, even when the interval $[a, b]$ in question is much smaller than T. Is it possible to have a bound that depends only on $\log(b-a)$? Finally, is it possible to improve the $O(\log(T))$ runtime per update currently required by all known strongly-adaptive algorithms?

References

Abernethy, J., Bartlett, P. L., Rakhlin, A., and Tewari, A. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the nineteenth annual conference on computational learning theory*, 2008.

Cutkosky, A. Combining online learning guarantees. In *Proceedings of the Thirty-Second Conference on Learning Theory*, pp. 895–913, 2019.

Cutkosky, A. and Orabona, F. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pp. 1493–1529, 2018.

Cutkosky, A. and Sarlos, T. Matrix-free preconditioning in online learning. In *International Conference on Machine Learning*, pp. 1455–1464, 2019.

Daniely, A., Gonen, A., and Shalev-Shwartz, S. Strongly adaptive online learning. In *International Conference on Machine Learning*, pp. 1405–1411, 2015.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory (COLT)*, 2010.

Foster, D. J., Rakhlin, A., and Sridharan, K. Online learning: Sufficient statistics and the burkholder method. In *Conference on Learning Theory (COLT)*, 2018.

Gyorgy, A. and Szepesvári, C. Shifting regret, mirror descent, and matrices. In *International Conference on Machine Learning*, pp. 2943–2951, 2016.

Gyorgy, A., Linder, T., and Lugosi, G. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012.

Hazan, E., Rakhlin, A., and Bartlett, P. L. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pp. 65–72, 2008.

Herbster, M. and Warmuth, M. K. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

Jun, K.-S., Orabona, F., Wright, S., and Willett, R. Improved strongly adaptive online learning using coin betting. In *Artificial Intelligence and Statistics*, pp. 943–951, 2017.

Kempka, M., Kotlowski, W., and Warmuth, M. K. Adaptive scale-invariant online algorithms for learning linear models. In *International Conference on Machine Learning*, pp. 3321–3330, 2019.

Mcmahan, B. and Streeter, M. No-regret algorithms for unconstrained online convex optimization. In *Advances in neural information processing systems*, pp. 2402–2410, 2012.

McMahan, H. B. and Streeter, M. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, 2010.

Mhammedi, Z. and Koolen, W. M. Lipschitz and comparator-norm adaptivity in online learning. *Conference on Learning Theory*, 2020.

Orabona, F. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pp. 1116–1124, 2014.

Orabona, F. and Pál, D. Coin betting and parameter-free online learning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 577–585. Curran Associates, Inc., 2016.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

van der Hoeven, D. User-specified local differential privacy in unconstrained adaptive online learning. In *Advances in Neural Information Processing Systems*, pp. 14080–14089, 2019.

van Erven, T. and Koolen, W. M. Metagrad: Multiple learning rates in online learning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3666–3674. Curran Associates, Inc., 2016.

Wang, G., Lu, S., and Zhang, L. Adaptivity and optimality: A universal algorithm for online convex optimization. *arXiv preprint arXiv:1905.05917*, 2019.

Zhang, L., Lu, S., and Zhou, Z.-H. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems*, pp. 1323–1333, 2018a.

Zhang, L., Yang, T., Zhou, Z.-H., et al. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning*, pp. 5877–5886, 2018b.

Zhang, L., Liu, T.-Y., and Zhou, Z.-H. Adaptive regret of convex and smooth functions. In *International Conference on Machine Learning*, pp. 7414–7423, 2019a.

Zhang, L., Wang, G., Tu, W.-W., and Zhou, Z.-H. Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. *arXiv preprint arXiv:1906.10851*, 2019b.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003.