Adapting to Smoothness: A More Universal Algorithm for Online Convex Optimization

Guanghui Wang,[1] Shiyin Lu,[1] Yao Hu,[2] Lijun Zhang[1,*]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2]YouKu Cognitive and Intelligent Lab, Alibaba Group, Beijing 100102, China
{wanggh, lusy, zhanglj}@lamda.nju.edu.cn, yaoohu@alibaba-inc.com

Abstract

We aim to design universal algorithms for online convex optimization, which can handle multiple common types of loss functions simultaneously. The previous state-of-the-art universal method has achieved the minimax optimality for general convex, exponentially concave and strongly convex loss functions. However, it remains an open problem whether *smoothness* can be exploited to further improve the theoretical guarantees. In this paper, we provide an affirmative answer by developing a novel algorithm, namely UFO, which achieves $O(\sqrt{L_*})$, $O(d \log L_*)$ and $O(\log L_*)$ regret bounds for the three types of loss functions respectively under the assumption of smoothness, where L_* is the cumulative loss of the best comparator in hindsight, and d is dimensionality. Thus, our regret bounds are much tighter when the comparator has a small loss, and ensure the minimax optimality in the worst case. In addition, it is worth pointing out that UFO is the *first* to achieve the $O(\log L_*)$ regret bound for strongly convex and smooth functions, which is tighter than the existing small-loss bound by an $O(d)$ factor.

Introduction

Online Convex Optimization (OCO) has become a popular paradigm for modeling many real-world applications, including ad placement, portfolio management and web ranking (Shalev-Shwartz and others 2012). It is conducted in a repeated manner: In each round $t = 1, \ldots, T$, firstly a learner selects an action $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$, at the same time an adversary reveals a loss function $f_t : \mathcal{X} \mapsto \mathbb{R}$, and then the learner suffers a loss $f_t(\mathbf{x}_t)$. The learner's goal is to minimize *regret*, which is defined as the cumulative loss of the learner and that of the best action in hindsight (Hazan and others 2016):

$$R(T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}). \tag{1}$$

In the literature, various algorithms have been developed for minimizing the regret under OCO, based on different assumptions on the properties of loss functions, including general convexity (Zinkevich 2003), exponential concavity (abbr. exp-concavity) and strong convexity (Hazan, Agarwal, and Kale 2007). However, existing methods can only deal with one type of loss functions. Moreover, for exp-concave and strongly convex functions, they require prior knowledge of loss functions as inputs for parameter tuning. This lack of universality not only leaves a heavy burden to users, but also impedes the applications to board domains. To overcome this obstacle, recent advances in OCO have developed a series of *universal* algorithms, such as AOGD (Hazan, Rakhlin, and Bartlett 2008) and MetaGrad (van Erven and Koolen 2016), which are able to handle various types of loss functions simultaneously. Among them, the state-of-the-art method is Maler (Wang, Lu, and Zhang 2019), which can adapt to general convex, exp-concave and strongly convex functions, and achieve $O(\sqrt{T})$, $O(d \log T)$ and $O(\log T)$ regret bounds respectively. These bounds are known to be minimax optimal, as matching lower bounds have been established (Ordentlich and Cover 1998; Abernethy et al. 2008).

On the other hand, in a wide range of online learning tasks such as online least square and ℓ_2-regularized regressions, the loss functions enjoy the property of *smoothness*. While there do exist several algorithms that can exploit this property for one single type of loss functions, such as convex and smooth (Srebro, Sridharan, and Tewari 2010), or exp-concave and smooth (Orabona, Cesa-Bianchi, and Gentile 2012), it remains unclear whether universal algorithms can make use of smoothness to achieve better performance. In this paper, we provide an affirmative answer by developing a novel algorithm, named UFO, which can automatically attain tighter bounds for smooth functions, and thus achieves broader universality.

Following previous work, our proposed method adopts the classic Learning with Experts Algorithm (LEA) (Cesa-Bianchi and Lugosi 2006). The basic idea is to maintain multiple algorithms with different learning rates in parallel as experts, and employ a meta-algorithm to identify the best on the fly. To exploit smoothness, the most straightforward idea is to use algorithms for smooth functions as experts. However, it does not provide tight results, because of the following technical challenges:

*Lijun Zhang is the corresponding author.
Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

- The current state-of-the-art method for smooth and strongly convex functions (Orabona, Cesa-Bianchi, and Gentile 2012) is suboptimal in the worst case, as there exists a large $O(d)$ gap.
- In existing universal methods, the experts are performed on a series of surrogate loss functions instead of f_t, and thus the smoothness of f_t can not be exploited.

To address these problems, we first propose the Smooth and Strongly convex Online Gradient Descent (S^2OGD) algorithm, and show that it enjoys an $O(\log L_*)$ *small-loss* regret bound, where $L_* = \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$ is the cumulative loss of the the best action in hindsight. Thus, this bound matches the minimax optimal bound in the worst case, and automatically becomes tighter whenever L_* is small, i.e., $L_* = o(T)$. Then, we develop a Universal algorithm For Online smooth and convex optimization (UFO), which follows from the basic LEA, while employing carefully designed novel surrogate loss functions and expert algorithms to exploit smoothness. Theoretical analysis shows that, under the assumption of smoothness, UFO achieves $O(\sqrt{L_*})$, $O(d\log L_*)$ and $O(\log L_*)$ small-loss regret bounds for general convex, exp-concave, and strongly convex functions, respectively.

Notation. Throughout the paper, we use $\|\cdot\|$ to denote the ℓ_2-norm. The weighted ℓ_2-norm associated with a positive semidefinite matrix $A \in \mathbb{R}^{d\times d}$ is defined as $\|\mathbf{x}\|_A^2 = \mathbf{x}^\top A\mathbf{x}$. Given a positive semidefinite matrix B, the B-weighted projection $\Pi_{\mathcal{X}}^B[\mathbf{x}]$ of $\mathbf{x}$ onto $\mathcal{X}$ is defined as $\Pi_{\mathcal{X}}^B[\mathbf{x}] = \operatorname{argmin}_{\mathbf{y}\in\mathcal{X}} \|\mathbf{y}-\mathbf{x}\|_B^2$. For the sake of clarity, we denote the gradient of f_t at $\mathbf{x}_t$ as $\mathbf{g}_t$, i.e., $\mathbf{g}_t = \nabla f_t(\mathbf{x}_t)$. The best action in hindsight is denoted as $\mathbf{x}_* = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$, and the d-dimension identity matrix is denoted as I_d.

Related Work

For general convex functions, the classic Online Gradient Descent (OGD) (Zinkevich 2003) with step size proportional to $O(1/\sqrt{t})$ (referred to as convex OGD) attains an $O(\sqrt{T})$ regret bound. If the loss functions are strongly convex, OGD with step size on the order of $O(1/t)$ (referred to as strongly convex OGD) achieves a regret bound of $O(\log T)$. For exp-concave functions, the Online Newton Step (ONS) (Hazan, Agarwal, and Kale 2007) enjoys an $O(d\log T)$ regret bound.

While the above bounds are minimax optimal, tighter results are attainable if the loss functions are smooth. Specifically, for general convex and smooth functions, Srebro, Sridharan, and Tewari (2010) show that OGD with a constant step size attains an $O(\sqrt{L})$ regret bound, where L is an upper bound of L_*. However, this method requires the modulus of smoothness as well as L as inputs to tune the step size, which are typically unavailable in practice. To tackle this problem, Zhang, Liu, and Zhou (2019) propose the Scale-free Online Gradient Descent (SOGD) algorithm, which is a special case of the Scale-free Online Mirror Descent (SOMD) algorithm (Orabona and Pál 2018). SOGD achieves the $O(\sqrt{L_*})$ regret bound, and is parameter-free to the modulus of smoothness and L. For exp-concave and smooth functions, Orabona, Cesa-Bianchi, and Gentile (2012) prove that an $O(d\log L_*)$ regret bound can be achieved by employing the ONS algorithm. Although this result also implies an $O(d\log L_*)$ regret bound for strongly convex and smooth functions, there still exists an $O(d)$ gap from the $\Omega(\log T)$ lower bound in the worst case. Aside from achieving small-loss bounds under the smoothness assumption, there are studies work on the variation bounds (Hazan and Kale. 2010; Chiang et al. 2012). For convex and smooth functions, Chiang et al. (2012) propose the extra-gradient descent algorithm, which attains an $O(\sqrt{D_T})$ regret bound, where $D_T = \sum_{t=2}^T \max_{\mathbf{x}\in\mathcal{X}} \|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|^2$ measures the variation in gradients of loss functions. Thus, the regret bounds automatically become tighter than $O(\sqrt{T})$ when D_T is small. They also develop a variant of ONS which achieves $O(d\log(D_T))$ variation bound. In this paper, we mainly focus on the small-loss bound, and it is an interesting direction to investigate whether the variation bounds can also be obtained by universal methods.

To cope with different types of loss functions simultaneously, Hazan, Rakhlin, and Bartlett (2008) develop the adaptive online gradient descent (AOGD), which can deal with general convex and strongly convex functions. Later, Do, Le, and Foo (2009) extend AOGD to the proximal setting. Both algorithms can attain $O(\sqrt{T})$ and $O(\log T)$ regret bounds for convex and strongly convex functions, respectively. However, they require the curvature of f_t as input in each round t, and fail to achieve a logarithmic regret bound for exp-concave functions. Another cornerstone is the multiple eta gradient (MetaGrad) (van Erven and Koolen 2016), which can automatically adapt to convex and exp-concave functions, and guarantees the corresponding minimax optimal bounds. However, Metagrad treats strongly convex functions as exp-concave functions, and thus suffers the suboptimal $O(d\log T)$ regret bound for strongly convex functions. This limitation is addressed by Wang, Lu, and Zhang (2019), who propose the multiple sub-algorithms and learning rates (Maler). Maler achieves minimal optimal regret bounds for general convex, exp-concave and strongly convex functions.

In this paper, we are devoted to designing algorithms that can adapt to the structure of the loss functions. A parallel line of research considers adapting to the structure inherent in data, such as sparsity (Duchi, Hazan, and Singer 2011; Tieleman and Hinton 2012; Kingma and Ba 2015; Reddi, Kale, and Kumar 2018). For general convex functions, these algorithms are able to achieve regret bounds which are tighter than $O(\sqrt{T})$ when the gradients are sparse.

In the definition of regret, we compare the performance of the learner with that of a fixed action. However, in some cases the best action may drift over time. To address this problem, recently some more stringent performance metrics are proposed. One of them is dynamic regret (Zinkevich 2003; Hall and Willett 2013; Zhang, Lu, and Zhou 2018; Lu and Zhang 2019), which is defined as the difference be-

tween the cumulative loss of the learner with that of any sequence of comparators. Another is adaptive regret (Hazan and Seshadhri 2007; Daniely, Gonen, and Shalev-Shwartz 2015; Jun et al. 2017; Wang, Zhao, and Zhang 2018), which is the maximum "local" regret over any contiguous time interval. Zhao et al. (2019) and Zhang, Liu, and Zhou (2019) show how to exploit smoothness to improve dynamic and adaptive regrets, respectively.

Main Results

In this section, we firstly provide some assumptions and definitions, then investigate how to utilize smoothness to improve the regret bound when the loss functions are strongly convex, and finally present our universal algorithm for multiple types of smooth functions and its theoretical guarantees.

Preliminary

Following previous work (Orabona, Cesa-Bianchi, and Gentile 2012; Wang, Lu, and Zhang 2019), we introduce the following assumptions and definitions (Boyd and Vandenberghe 2004).

Assumption 1. *The domain $\mathcal{X}$ is convex, and its diameter is bounded by D, i.e.,*

$$\max_{\mathbf{x}_1,\mathbf{x}_2\in\mathcal{X}} \|\mathbf{x}_1-\mathbf{x}_2\| \le D.$$

Assumption 2. *The gradients of all loss functions are bounded by G, i.e.,*

$$\max_{\mathbf{x}\in\mathcal{X}} \|f_t(\mathbf{x})\| \le G, \forall t \in \{1,\dots,T\}.$$

Definition 1. *A function $f:\mathcal{X}\mapsto\mathbb{R}$ is convex if*

$$f(\mathbf{x}_1) \ge f(\mathbf{x}_2) + \nabla f(\mathbf{x}_2)^\top(\mathbf{x}_1-\mathbf{x}_2), \forall \mathbf{x}_1,\mathbf{x}_2\in\mathcal{X}. \quad (2)$$

Definition 2. *A function $f:\mathcal{X}\mapsto\mathbb{R}$ is λ-strongly convex if $\forall \mathbf{x}_1,\mathbf{x}_2\in\mathcal{X}$,*

$$\begin{aligned} f(\mathbf{x}_1) \ge & f(\mathbf{x}_2) + \nabla f(\mathbf{x}_2)^\top(\mathbf{x}_1-\mathbf{x}_2) \\ & + \frac{\lambda}{2}\|\mathbf{x}_1-\mathbf{x}_2\|^2. \end{aligned} \quad (3)$$

Definition 3. *A function $f:\mathcal{X}\mapsto\mathbb{R}$ is α-exp-concave if $\exp(-\alpha f(\mathbf{x}))$ is concave.*

Definition 4. *A function $f:\mathcal{X}\mapsto\mathbb{R}$ is h-smooth if*

$$\|\nabla f(\mathbf{x}_1)-\nabla f(\mathbf{x}_2)\| \le h\|\mathbf{x}_1-\mathbf{x}_2\|, \ \forall \mathbf{x}_1,\mathbf{x}_2\in\mathcal{X}.$$

Finally, we review some useful properties.

Lemma 1 (Hazan, Agarwal, and Kale (2007), Lemma 3)**.** *Suppose Assumptions 1 and 2 hold, and $f:\mathcal{X}\mapsto\mathbb{R}$ is α-exp-concave. Then, the following holds: $\forall\beta \le \frac{1}{2}\min\{\frac{1}{4GD},\alpha\}$,*

$$\begin{aligned} f(\mathbf{x}_1) \ge & f(\mathbf{x}_2) + (\mathbf{x}_1-\mathbf{x}_2)^\top\nabla f(\mathbf{x}_2) \\ & + \frac{\beta}{2}\left((\mathbf{x}_1-\mathbf{x}_2)^\top\nabla f(\mathbf{x}_2))\right)^2, \ \forall \mathbf{x}_1,\mathbf{x}_2\in\mathcal{X}. \end{aligned} \quad (4)$$

Lemma 2 (Srebro, Sridharan, and Tewari (2010), Lemma 3.1)**.** *For an h-smooth and nonnegative function $f:\mathcal{X}\mapsto\mathbb{R}$, we have*

$$\|\nabla f(\mathbf{x})\| \le \sqrt{4hf(\mathbf{x})}, \forall \mathbf{x}\in\mathcal{X}. \quad (5)$$

Algorithm 1 Smooth and Strongly Convex OGD (S^2OGD)

Input: Parameters δ and γ

1: $\mathbf{x}_1 = \mathbf{0}$
2: **for** $t = 1,\dots,T$ **do**
3: Submit $\mathbf{x}_t$ and then receive f_t
4: Suffer loss $f_t(\mathbf{x}_t)$, and observe $\mathbf{g}_t$
5: Update

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}^{I_d}[\mathbf{x}_t - \alpha_t\mathbf{g}_t]$$

where

$$\alpha_t = \frac{\gamma}{\delta + \sum_{i=1}^t \|\mathbf{g}_i\|^2}.$$

6: **end for**

Smooth and Strongly Convex OGD

In this section, we propose a novel algorithm for smooth and strongly convex functions. The proposed algorithm is built upon the SOGD algorithm (Zhang, Liu, and Zhou 2019), which is designed for smooth and convex functions. In SOGD, the action in each round t is updated by the following projected gradient descent step

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}^{I_d}[\mathbf{x}_t - \alpha_t\mathbf{g}_t]$$

where the step size α_t is configured as

$$\begin{aligned} \alpha_t &= \frac{\gamma}{\sqrt{\delta + \sum_{i=1}^t \|\mathbf{g}_i\|^2}} \\ &= \frac{\gamma}{\sqrt{\delta + t\left(\frac{1}{t}\sum_{i=1}^t \|\mathbf{g}_i\|^2\right)}} \end{aligned} \quad (6)$$

and $\delta,\gamma > 0$ are constant parameters. Similarly to convex OGD, the step size of SOGD decreases in general on the order of $O(1/\sqrt{t})$, but is adjusted by the average of past gradients. This enables SOGD to automatically adapt to smoothness, and achieve the $O(\sqrt{L_*})$ regret bound.

For strongly convex and smooth functions, mimicking the behavior of strongly convex OGD, where the step size decreases on the order of $O(1/t)$, we modify the step size of SOGD as

$$\begin{aligned} \alpha_t &= \frac{\gamma}{\delta + t\left(\frac{1}{t}\sum_{i=1}^t \|\mathbf{g}_i\|^2\right)} \\ &= \frac{\gamma}{\delta + \sum_{i=1}^t \|\mathbf{g}_i\|^2} \end{aligned} \quad (7)$$

so that its step size decays approximately proportional to $O(1/t)$, which is similar to that in the strongly convex OGD. The new algorithm, named Smooth and Strongly convex Online Gradient Descent (S^2OGD), is summarized in Algorithm 1. For S^2OGD, we prove the following theorem.

Theorem 1. *Suppose Assumptions 1 and 2 hold, and all loss functions are λ-strongly convex. Let $\gamma = \frac{G^2}{\lambda}$ and $\delta = G^2$. Then, S^2OGD guarantees the following regret bound:*

$$R(T) \le \lambda D^2 + \frac{G^2}{2\lambda}\ln\left(\frac{1}{G^2}\sum_{t=1}^T \|\mathbf{g}_t\|^2 + 1\right).$$

Moreover, if all loss functions are also nonnegative and h-smooth, S^{2}OGD enjoys

$$R(T) \leq \frac{G^2}{2\lambda} \ln \left(\frac{8h}{G^2} \sum_{t=1}^{T} f_t(\mathbf{x}_*) + \mu \right) + \lambda D^2$$

where

$$\mu = \frac{8\lambda h D^2}{G^2} + \frac{4h}{\lambda} \ln \frac{4h + \lambda e}{e\lambda} + 2$$

is a constant.

Remark 1. Theorem 1 implies that S^{2}OGD guarantees an $O(\log L_*)$ small-loss regret bound, which reduces to the minimax optimal $O(\log T)$ regret bound in the worst case, and automatically becomes tighter when L_* is small. We note that this is the *first* time that such type of regret bound is achieved. Compared to $O(d \log L_*)$, i.e., the existing state-of-the-art small-loss regret bound (Orabona, Cesa-Bianchi, and Gentile 2012), our result is tighter by an $O(d)$ factor, which is thus of independent interest.

A Parameter-Free Algorithm for Smooth and Strongly Convex Optimization

While S^{2}OGD successfully achieves the $O(\log L_*)$ regret bound, it requires the modulus of strong convexity λ as input to tune the step size. In this section, we propose a novel algorithm which ensures the $O(\log L_*)$ regret bound while being parameter-free to λ. Then, in the next section, we extend the proposed algorithm to support more types of functions. Our proposed algorithm is inspired by Maler (Wang, Lu, and Zhang 2019), so we first briefly review some intuition behind this algorithm below.

Review of Maler To deal with strongly convex functions, Maler introduces the following surrogate loss function:

$$s_t^\eta(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 G^2 \|\mathbf{x}_t - \mathbf{x}\|^2$$

where $\eta \in (0, \frac{1}{5DG}]$ is a constant. It can be easily seen that s_t^η is $2\eta^2 G^2$-strongly convex. Thus, by applying strongly convex OGD on s_t^η, we obtain

$$\sum_{t=1}^{T} s_t^\eta(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} s_t^\eta(\mathbf{x}) \leq O(\log T). \tag{8}$$

On the other hand, we have

$$\begin{aligned}
R(T) &\overset{(2)}{\leq} \sum_{t=1}^{T} \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}_*) \\
&= \frac{-\sum_{t=1}^{T} s_t^\eta(\mathbf{x}_*)}{\eta} + \eta V_T^s \\
&= \frac{\sum_{t=1}^{T} s_t^\eta(\mathbf{x}_t) - \sum_{t=1}^{T} s_t^\eta(\mathbf{x}_*)}{\eta} + \eta V_T^s \\
&\leq \frac{\sum_{t=1}^{T} s_t^\eta(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} s_t^\eta(\mathbf{x})}{\eta} + \eta V_T^s \\
&\overset{(8)}{\leq} \frac{O(\log T)}{\eta} + \eta V_T^s
\end{aligned}$$

Algorithm 2 A Parameter-free algorithm for Strongly convex and Smooth functions (PASS)

Input: Learning rates $\eta_1, \eta_2, \ldots,$ prior weights $\pi_1^{\eta_1, \hat{s}}, \pi_1^{\eta_2, \hat{s}}, \ldots$
for $t = 1, \ldots, T$ **do**
  Get predictions $\mathbf{x}_t^{\eta, \hat{s}}$ from Algorithm 3 for all η
  Play

$$\mathbf{x}_t = \frac{\sum_\eta \pi_t^{\eta, \hat{s}} \eta \mathbf{x}_t^{\eta, \hat{s}}}{\sum_\eta \eta \pi_t^{\eta, \hat{s}}}$$

  Observe gradient $\mathbf{g}_t$ and send it to all experts
  Update weights:

$$\pi_{t+1}^{\eta, \hat{s}} = \frac{\pi_t^{\eta, \hat{s}} e^{-\hat{s}_t^\eta(\mathbf{x}_t^{\eta, \hat{s}})}}{\sum_\eta \pi_t^{\eta, \hat{s}} e^{-\hat{s}_t^\eta(\mathbf{x}_t^{\eta, \hat{s}})}}$$

  for all η
end for

where $V_T^s = \sum_{t=1}^{T} G^2 \|\mathbf{x}_t - \mathbf{x}_*\|^2$, the first equality is derived from the definition of s_t^η, and the second equality is due to the fact that $\forall t \in \{1, \ldots, T\}$, $s_t^\eta(\mathbf{x}_t) = 0$.
By configuring η as

$$\eta^* = \sqrt{\frac{O(\log T)}{V_T^s}}$$

we have

$$R(T) \leq \sum_{t=1}^{T} \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}_*) \leq O(\sqrt{V_T^s \log T})$$

which, combining with Definition 2, automatically reduces to $O(\log T)$ for strongly convex functions. However, the optimal η^* can not be obtained, since it depends on $\mathbf{x}_*$ and the whole learning history. To resolve this problem, Maler maintains multiple strongly convex OGD as experts, each of which runs on s_t^η with a different η, and then utilizes a meta-algorithm to identify the expert with the best η adaptively. Specifically, to deal with strongly convex functions, Maler keeps $C = \lceil \frac{1}{2} \log_2 T \rceil + 1$ experts. In the t-th round, each expert receives s_t^η as the loss function, and runs strongly convex OGD to output an action $\mathbf{x}_t^{\eta, s}$. Maler then calculates the final action $\mathbf{x}_t$ according to the Titled Exponential Weighted Algorithm (TEWA) [1]:

$$\mathbf{x}_t = \frac{\sum_\eta \pi_t^{\eta, s} \eta \mathbf{x}_t^{\eta, s}}{\sum_\eta \eta \pi_t^{\eta, s}} \tag{9}$$

where $\pi_t^{\eta, s} \propto \exp(-\sum_{i=1}^{t} s_i^\eta(\mathbf{x}_i^{\eta, s}))$. Theoretical analysis shows that, in this way, we can successfully achieve an $O(\sqrt{V_T^s \log T} + \log T)$ regret bound.

[1] Here we only provide the main idea of Maler on how to deal with strongly convex functions. We refer to Wang, Lu, and Zhang (2019) for the exact weighting technique.

Algorithm 3 Strongly convex expert algorithm (S^3OGD)

1: $\mathbf{x}_1^{\eta,\hat{s}} = \mathbf{0}$, $\delta = 2\eta^2 G^2$
2: **for** $t = 1, \ldots, T$ **do**
3: Send $\mathbf{x}_t^{\eta,\hat{s}}$ to the meta-algorithm
4: Receive gradient $\mathbf{g}_t$ from the meta-algorithm
5: Update

$$\mathbf{x}_{t+1}^{\eta,\hat{s}} = \Pi_{\mathcal{X}}^{I_d}\left[\mathbf{x}_t^{\eta,\hat{s}} - \alpha_t^\eta \nabla \hat{s}_t(\mathbf{x}_t^{\eta,\hat{s}})\right]$$

where

$$\nabla \hat{s}_t^\eta(\mathbf{x}) = \eta \mathbf{g}_t + 2\eta^2 \|\mathbf{g}_t\|^2 (\mathbf{x} - \mathbf{x}_t)$$

and

$$\alpha_t^\eta = \frac{1}{\delta + \sum_{i=1}^t \lambda_i^\eta}.$$

6: **end for**

Our Algorithm Following Maler, to exploit the smoothness of strongly convex loss functions, one may consider directly replacing the strongly convex OGD with our S^2OGD, and hope to obtain an $O(\sqrt{V_T^s \log L_*})$ regret bound, which can reduce to $O(\log L_*)$ for strongly convex functions. However, this procedure can not achieve our goal due to the following issues.

- Since the expert algorithm is performed on s_t^η, in the derived regret bound, the small-loss part is related to s_t^η rather than f_t.
- Although s_t^η is $2\eta^2 G^2$-smooth, it could be negative, which makes the self-bounding property in Lemma 2 inapplicable.

Thus, under this method, one can only hope to obtain a regret bound on the order of $O(\sqrt{V_T^s \log(\sum_{t=1}^T \|\nabla s_t^\eta(\mathbf{x}_t^{\eta,s})\|^2)}$, which does not lead to a small-loss regret bound with respect to f_t.

To tackle this problem, we introduce a new surrogate loss function:

$$\hat{s}_t^\eta(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 \|\mathbf{g}_t\|^2 \|\mathbf{x}_t - \mathbf{x}\|^2 \tag{10}$$

which is a lower bound of s_t^η. We first observe two important properties of $\hat{s}_t^\eta$.

Lemma 3. *Suppose Assumptions 1 and 2 hold. Then, we have* $\forall \eta \in (0, \frac{1}{5DG}], \mathbf{x} \in \mathcal{X}$,

$$\|\nabla \hat{s}_t^\eta(\mathbf{x})\|^2 \leq 2\eta^2 \|\mathbf{g}_t\|^2.$$

Lemma 4. *Let* $\lambda_t^\eta = 2\eta^2 \|\mathbf{g}_t\|^2$. *We have* $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$,

$$\hat{s}_t^\eta(\mathbf{x}_1) \geq \hat{s}_t^\eta(\mathbf{x}_2) + \nabla \hat{s}_t^\eta(\mathbf{x}_2)^\top (\mathbf{x}_1 - \mathbf{x}_2) + \frac{\lambda_t^\eta}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2.$$

Lemma 3 reveals that, if we can derive the gradient-dependent $O(\sqrt{V_T^s \log \sum_{t=1}^T \|\nabla \hat{s}_t^\eta(\mathbf{x}_t^{\eta,\hat{s}})\|^2})$ regret bound, we can obtain a regret bound of $O(\sqrt{V_T^s \log \sum_{t=1}^T \|\mathbf{g}_t\|^2})$. This bound, together with the self-bounding property in Lemma 2, can further lead to the desired small-loss regret bound. However, the new surrogate loss brings another challenge: Since the overall "strongly convex" parameter of $\hat{s}_1^\eta, \ldots, \hat{s}_T^\eta$, i.e.,

$$\lambda^\eta = \min_{t \in \{1,\ldots,T\}} \lambda_t^\eta$$

is unavailable and even might be zero, S^2OGD can not be applied. To handle this issue, inspired by Hazan, Rakhlin, and Bartlett (2008), we propose to perform the following update rule on $\hat{s}_t^\eta$:

$$\mathbf{x}_{t+1}^{\eta,\hat{s}} = \Pi_{\mathcal{X}}^{I_d}[\mathbf{x}_t^{\eta,\hat{s}} - \alpha_t^\eta \nabla \hat{s}_t^\eta(\mathbf{x}_t^{\eta,\hat{s}})]$$

where

$$\alpha_t^\eta = \frac{1}{\delta + \sum_{i=1}^t \lambda_i^\eta} \tag{11}$$

and $\lambda_t^\eta = 2\eta^2 \|\mathbf{g}_t\|^2$ is the curvature of $\hat{s}_t^\eta$, which is defined in Lemma 3. We name this algorithm Surrogate loss S^2OGD (S^3OGD), summarized in Algorithm 3. The main advantage of S^3OGD is that it does not require λ^η to tune the step size, and can achieve the following regret bound (Hazan, Rakhlin, and Bartlett 2008):

$$\begin{aligned}&\sum_{t=1}^T \hat{s}_t(\mathbf{x}_t^{\eta,\hat{s}}) - \sum_{t=1}^T \hat{s}_t(\mathbf{x}) \\ \leq& O\left(\sum_{t=1}^T \frac{\|\nabla \hat{s}_t^\eta(\mathbf{x}_t^{\eta,\hat{s}})\|^2}{\delta + \sum_{i=1}^t \lambda_i^\eta}\right).\end{aligned} \tag{12}$$

For general strongly convex cases, it only leads to an $O(\log T)$ regret bound. Nevertheless, thanks to the nice properties of $\hat{s}_t^\eta$, i.e., Lemma 3 and Lemma 4, for an expert runing on $\hat{s}_t^\eta$, we can derive the following tighter regret bound, which is on the order of $O(\log(\sum_{t=1}^T \|\mathbf{g}_t\|^2))$.

Theorem 2. *Suppose Assumptions 1 and 2 hold. Let* $\delta = 2\eta^2 G^2$. *Then, for an expert that runs S^3OGD on* $\hat{s}_t^\eta$, *we have* $\forall \mathbf{x} \in \mathcal{X}$,

$$\begin{aligned}&\sum_{t=1}^T \hat{s}_t(\mathbf{x}_t^{\eta,\hat{s}}) - \sum_{t=1}^T \hat{s}_t(\mathbf{x}) \\ \leq& 1 + \ln\left(\frac{1}{G^2}\sum_{t=1}^T \|\mathbf{g}_t\|^2 + 1\right).\end{aligned} \tag{13}$$

Based on Theorem 2, we propose our Parameter-free algorithm for Strongly convex and Smooth functions (PASS), which is summarized in Algorithm 2. In each round t, we keep $\lceil \frac{1}{2}\log_2 T \rceil + 1$ experts, each of which runs S^3OGD on $\hat{s}_t^\eta$ with a different η, and outputs an action, denoted as $\mathbf{x}_t^{\eta,\hat{s}}$ (Step 3). For expert $i \in \{0, \ldots, \lceil \frac{1}{2}\log_2 T \rceil\}$, we configure

$$\pi_1^{\eta_i,\hat{s}} = \frac{1}{C},\ \eta_i = \frac{2^{-i}}{5GD}.$$

Then, PASS updates $\mathbf{x}_t$ by TWEA (Step 4):

$$\mathbf{x}_t = \frac{\sum_\eta \pi_t^{\eta,\hat{s}} \eta \mathbf{x}_t^{\eta,\hat{s}}}{\sum_\eta \eta \pi_t^{\eta,\hat{s}}}.$$

Finally, the algorithm observes $\mathbf{g}_t$ (Step 5) and updates the weights of experts according to their losses on $\hat{s}_t^\eta$ (Step 6).

We obtain the following regret bound for our PASS algorithm:

Theorem 3. *Suppose Assumptions 1 and 2 hold, and all loss functions are λ-strongly convex, h-smooth and nonnegative.Then, PASS achieves the following regret bound:*

$$\begin{aligned}&R(T)\\ \leq& m\ln\left(\frac{8h}{G^2}\sum_{t=1}^{T}f_t(\mathbf{x}_*)+\frac{8hn}{G^2}+\frac{8mh}{G^2}\ln\left(\frac{8mh+eG^2}{eG^2}\right)\right.\\ &\left.+2\right)+n\\ =&O\left(\frac{1}{\lambda}\log L_*\right)\end{aligned}$$

where

$$m=\left(10GD+\frac{9G^2}{2\lambda}\right)$$

and

$$n=\left(10GD+\frac{9G^2}{2\lambda}\right)(\ln(\log_2 T+3)+1).$$

Remark 2. Theorem 3 indicates that PASS achieves the $O(\frac{1}{\lambda}\log L_*)$ small-loss regret bound for smooth and λ-strongly convex functions, which becomes much tighter whenever L_* is small. Compared to S^{2}OGD, PASS doesn't require the modulus of strong convexity to tune parameters, and is thus *parameter-free* to λ.

A Universal Algorithm for Online Smooth Optimization

In this section, we extend PASS to support more types of loss functions.

To adapt to exp-concave functions, we employ the following surrogate loss function proposed by van Erven and Koolen (2016):

$$\ell_t^\eta(\mathbf{x})=-\eta(\mathbf{x}_t-\mathbf{x})^\top\mathbf{g}_t+\eta^2\left((\mathbf{x}_t-\mathbf{x}_*)^\top\mathbf{g}_t\right)^2. \quad (14)$$

We introduce the following two lemmas about ℓ_t^η. The first lemma implies that ℓ_t^η is also exp-concave for small η, and the second lemma reflects a direct connection between the gradients of ℓ_t^η and $\mathbf{g}_t$.

Lemma 5 (Wang, Lu, and Zhang 2019)**.** *For $\eta\in(0,\frac{1}{5DG}]$, ℓ_t^η is 1-exp-concave.*

Lemma 6. *Suppose Assumptions 1 and 2 hold. Then, $\forall\eta\in(0,\frac{1}{5GD}]$, $\forall\mathbf{x}\in\mathcal{X}$,*

$$\|\nabla\ell_t^\eta(\mathbf{x})\|^2\leq 2\eta^2\|\mathbf{g}_t\|^2.$$

The two lemmas motivate us to adopt ONS as expert algorithm on ℓ_t^η, and derive the following regret bound:

$$\begin{aligned}&R(T)\\ \leq&\sum_{t=1}^{T}\mathbf{g}_t^\top(\mathbf{x}_t-\mathbf{x}_*)\\ \leq&O\left(\sqrt{V_T^\ell d\log\left(\sum_{t=1}^{T}\|\mathbf{g}_t\|^2\right)}+d\log\left(\sum_{t=1}^{T}\|\mathbf{g}_t\|^2\right)\right)\end{aligned}$$

where $V_T^\ell=\sum_{t=1}^{T}\left(\mathbf{g}_t^\top(\mathbf{x}_t-\mathbf{x}_*)\right)^2$. When the loss functions are exp-concave and smooth, the above regret bound can reduce to $O(d\log L_*)$ by exploiting Lemma 1 and Lemma 2.

For convex functions, Wang, Lu, and Zhang (2019) propose the following convex surrogate loss function:

$$c_t(\mathbf{x})=-\eta^c(\mathbf{x}_t-\mathbf{x})^\top\mathbf{g}_t+(\eta^c)^2G^2D^2 \quad (15)$$

where $\eta^c=\frac{1}{2GD\sqrt{T}}$. A naive approach is to directly apply algorithms for convex and smooth functions such as SOGD on $c_t(\mathbf{x})$ as experts. However, due to the constant term (the second term of c_t), incorporating this approach into LEA only gives to an $O(\sqrt{T})$ regret bound, and fails to achieve the desired small-loss bound. To address this problem, we design a new surrogate loss:

$$c_t^\eta(\mathbf{x})=-\eta(\mathbf{x}_t-\mathbf{x})^\top\mathbf{g}_t+\eta^2\|\mathbf{g}_t\|^2D^2 \quad (16)$$

where the second term is dependent on $\mathbf{g}_t$. We then employ SOGD on c_t^η as experts. In this way, we are able to obtain

---

Algorithm 4 Universal algorithm For Online smooth optimization (UFO)

1: **Input:** Learning rates $\eta_1,\eta_2,\ldots$, prior weights $\pi_1^{\eta_1,\hat{s}},\pi_1^{\eta_2,\hat{s}},\ldots,\pi_1^{\eta_1,\ell},\pi_1^{\eta_2,\ell},\ldots$, and $\pi_1^{\eta_1,c},\pi_1^{\eta_2,c},\ldots$
2: **for** $t=1,\ldots,T$ **do**
3: Get predictions $\mathbf{x}_t^{\eta,c}$, $\mathbf{x}_t^{\eta,\ell}$ and $\mathbf{x}_t^{\eta,\hat{s}}$ from Algorithms 5, 6 and 3 for all η
4: Update

$$\mathbf{x}_t=\frac{\sum_\eta(\pi_t^{\eta,\hat{s}}\eta\mathbf{x}_t^{\eta,\hat{s}}+\pi_t^{\eta,\ell}\eta\mathbf{x}_t^{\eta,\ell}+\pi_t^{\eta,c}\eta\mathbf{x}_t^{\eta,c})}{\sum_\eta(\pi_t^{\eta,\hat{s}}\eta+\pi_t^{\eta,\ell}\eta+\pi_t^{\eta,c}\eta)}$$

5: Observe gradient $\mathbf{g}_t$ and send it to all experts
6: Update weights:

$$\pi_{t+1}^{\eta,c}=\frac{\pi_t^{\eta,c}e^{-c_t^\eta(\mathbf{x}_t^{\eta,c})}}{\Phi_t}\text{ for all }\eta$$

$$\pi_{t+1}^{\eta,\hat{s}}=\frac{\pi_t^{\eta,\hat{s}}e^{-\hat{s}_t^\eta(\mathbf{x}_t^{\eta,\hat{s}})}}{\Phi_t}\text{ for all }\eta$$

$$\pi_{t+1}^{\eta,\ell}=\frac{\pi_t^{\eta,\ell}e^{-\ell_t^\eta(\mathbf{x}_t^{\eta,\ell})}}{\Phi_t}\text{ for all }\eta$$

where

$$\Phi_t=\sum_\eta\left(\pi_t^{\eta,\hat{s}}e^{-\hat{s}_t^\eta(\mathbf{x}_t^{\eta,\hat{s}})}+\pi_t^{\eta,\ell}e^{-\ell_t^\eta(\mathbf{x}_t^{\eta,\ell})}+\pi_t^{\eta,c}e^{-c_t^\eta(\mathbf{x}_t^{\eta,c})}\right)$$

7: **end for**

Algorithm 5 Convex expert algorithm (SOGD)

1: $\mathbf{x}_1^{\eta,c} = \mathbf{0}, \delta = \eta^2, \gamma = \frac{D}{\sqrt{2}}$
2: **for** $t = 1, \ldots, T$ **do**
3: Send $\mathbf{x}_t^{\eta,c}$ to Algorithm 4
4: Receive gradient $\mathbf{g}_t$ from Algorithm 4
5: Update

$$\mathbf{x}_{t+1}^{\eta,c} = \Pi_{\mathcal{X}}^{I_d}\left[\mathbf{x}_t^{\eta,c} - \frac{\gamma \nabla c_t^{\eta}(\mathbf{x}_t^{\eta,c})}{\sqrt{\delta + \sum_{i=1}^{t} \|\nabla c_i^{\eta}(\mathbf{x}_i^{\eta,c})\|^2}}\right]$$

where $\nabla c_t^{\eta}(\mathbf{x}) = \mathbf{g}_t$.
6: **end for**

Algorithm 6 Exp-concave expert algorithm (ONS)

1: $\mathbf{x}_1^{\eta,\ell} = \mathbf{0}$, $\beta = \frac{1}{2}\min\left\{\frac{1}{4G^{\ell}D}, 1\right\} = \frac{25}{56}$, where $G^{\ell} = \frac{7}{25D}$, $\Sigma_1 = \frac{1}{\beta^2 D^2} I_d$
2: **for** $t = 1, \ldots, T$ **do**
3: Send $\mathbf{x}_t^{\eta,\ell}$ to Algorithm 4
4: Receive gradient $\mathbf{g}_t$ from Algorithm 4
5: Update

$$\Sigma_{t+1} = \Sigma_t + \nabla \ell_t^{\eta}\left(\mathbf{x}_t^{\eta,\ell}\right)\left(\nabla \ell_t^{\eta}\left(\mathbf{x}_t^{\eta,\ell}\right)\right)^{\top}$$

$$\mathbf{x}_{t+1}^{\eta,\ell} = \Pi_{\mathcal{X}}^{\Sigma_{t+1}}\left[\mathbf{x}_t^{\eta,\ell} - \frac{1}{\beta}\Sigma_{t+1}^{-1}\nabla \ell_t^{\eta}\left(\mathbf{x}_t^{\eta,\ell}\right)\right]$$

where $\nabla \ell_t^{\eta}(\mathbf{x}_t^{\eta,\ell}) = \eta \mathbf{g}_t + 2\eta^2 \mathbf{g}_t \mathbf{g}_t^{\top}(\mathbf{x}_t^{\eta,\ell} - \mathbf{x}_t)$.
6: **end for**

an $O(\sqrt{\sum_{t=1}^{T} \|\mathbf{g}_t\|^2})$ regret bound, which attains $O(\sqrt{L_*})$ under the assumption of general convexity and smoothness.

Our Universal algorithm For Online smooth optimization (UFO) is summarized in Algorithm 4, which is an extension of PASS algorithm by incorporating more types of experts. Similar to PASS, our proposed UFO follows from the LEA, while utilizing different types of algorithms as experts to deal with multiple types of loss functions. Each expert algorithm is associated with a carefully designed surrogate loss. Specifically, in each round t, we maintain three types of experts:

- Strongly convex experts. For strongly convex functions, similarly to Pass, we keep $C = \lceil \frac{1}{2}\log_2 T \rceil + 1$ strongly convex experts, each of which runs S^{3}OGD (Algorothm 3) on $\hat{s}_t^{\eta}$ to output an action $\mathbf{x}_t^{\eta,\hat{s}}$. For each expert $i \in \{0, \ldots, C-1\}$, we configure its learning rate and prior weight as

$$\eta_i = \frac{2^{-i}}{5GD}, \ \pi_1^{\eta,\hat{s}} = \frac{1}{3C}.$$

- Exp-concave experts. To handle exp-concave functions, we also maintain $C = \lceil \frac{1}{2}\log_2 T \rceil + 1$ experts, each of which receives ℓ_t^{η} as the loss function, and runs the standard Online Newton Step (ONS) (Algorithm 6) to output an action, denoted as $\mathbf{x}_t^{\eta,\ell}$. For each expert $i \in \{0, \ldots, C-1\}$, we set

$$\eta_i = \frac{2^{-i}}{5GD}, \ \pi_1^{\eta_i,\ell} = \frac{1}{3C}.$$

- Convex experts. To deal with general convex functions, we maintain $C = \lceil \frac{1}{2}\log_2 T \rceil + 1$ convex experts. Each expert receives c_t^{η} as the loss function, and runs SOGD (Algorithm 5) to output an action, denoted by $\mathbf{x}_t^{\eta,c}$. For each $i \in \{0, \ldots, C-1\}$, we let

$$\eta_i = \frac{2^{-i}}{5GD}, \ \pi_1^{\eta,c} = \frac{1}{3C}.$$

In each round t, UFO firstly receives the outputs of all experts (Step 3), then submits the following action (Step 4):

$$\mathbf{x}_t = \frac{\sum_{\eta}(\pi_t^{\eta,\hat{s}}\eta\mathbf{x}_t^{\eta,\hat{s}} + \pi_t^{\eta,\ell}\eta\mathbf{x}_t^{\eta,\ell} + \pi_t^{\eta,c}\eta\mathbf{x}_t^{\eta,c})}{\sum_{\eta}(\pi_t^{\eta,\hat{s}}\eta + \pi_t^{\eta,\ell}\eta + \pi_t^{\eta,c}\eta)} \tag{17}$$

which is the weighted sum of the outputs of experts titled by their own η. After the gradient $\mathbf{g}_t$ is observed (Step 5), UFO updates the weights based on the historical performance of the experts (Step 6) on their own surrogate loss functions.

For UFO, we can derive the following theoretical guarantees.

Theorem 4. *Suppose Assumptions 1 and 2 hold, and all loss functions are h-smooth and nonnegative. Then, for general convex, α-exp-concave, and λ-strongly convex loss functions, the regret of UFO algorithm is upper bounded by $O(\sqrt{hL_*})$, $O(\frac{d}{\alpha}\log(hL_*))$ and $O(\frac{1}{\lambda}\log(hL_*))$, respectively.* [2]

Remark 3. Theorem 4 implies that, under the assumption of smoothness, for general convex, α-exp-concave and λ-strongly convex loss functions, UFO attains the minimax $O(\sqrt{T})$, $O(\frac{d}{\alpha}\log T)$, and $O(\frac{1}{\lambda}\log T)$ regret bounds in the worst case, and achieves tighter bounds whenever $L_* = o(T)$.

Conclusion and Future Work

In this paper, we propose a universal algorithm for online smooth optimization, which can achieve small loss regret bounds for general convex, exp-concave and strongly convex functions simultaneously, under the assumption of smoothness. The proposed algorithm, named UFO, follows the LEA to deal with the uncertainty of the loss function type, and utilizes carefully designed surrogate loss functions and expert algorithms to make use of the smoothness. We show that, under the smoothness assumption, the regret bound of UFO for the three types of loss functions attain $O(\sqrt{L_*})$, $O(d\log L_*)$ and $O(\log L_*)$ respectively, which match the minimax optimal results in the worst case, and become much tighter as long as $L_* = o(T)$. Finally, we note it is the first time that the $O(\log L_*)$ regret bound is attained for strongly convex and smooth functions. In the future, we will investigate our algorithm to achieve broader universality.

[2] Due to the page limitation, we only provide the order of regret bounds. The details can be found in the full version.

Acknowledgment

This work was partially supported by NSFC (61976112, 61921006), NSFC-NRF Joint Research Project (61861146001), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Abernethy, J.; Bartlett, P. L.; Rakhlin, A.; and Tewari, A. 2008. Optimal stragies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory*, 415–423.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Chiang, C.-K.; Yang, T.; Lee, C.-J.; Mahdavi, M.; Lu, C.-J.; Jin, R.; and Zhu, S. 2012. Online optimization with gradual variations. In *Proceedings of the 25th Annual Conference on Learning Theory*, 415–423.

Daniely, A.; Gonen, A.; and Shalev-Shwartz, S. 2015. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning*, 1405–1411.

Do, C. B.; Le, Q. V.; and Foo, C.-S. 2009. Proximal regularization for online and batch learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 257–264.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.

Hall, E. C., and Willett, R. M. 2013. Dynamical models and tracking regret in online convex programming. In *Proceedings of the 30th International Conference on Machine Learning*, 579–587.

Hazan, E.; Agarwal, A.; and Kale, S. 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69:169–192.

Hazan, E., and Kale., S. 2010. Fast rates in statistical and online learning. *Machine Learning* 165–188.

Hazan, E., et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2(3-4):157–325.

Hazan, E., and Seshadhri, C. 2007. Adaptive algorithms for online decision problems. In *Electronic Colloquium on Computational Complexity*.

Hazan, E.; Rakhlin, A.; and Bartlett, P. L. 2008. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 21*, 65–72.

Jun, K.-S.; Orabona, F.; Wright, S.; and Willett, R. 2017. Improved strongly adaptive online learning using coin betting. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 943–951.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. *Proceedings of 3rd International Conference on Learning Representations*.

Lu, S., and Zhang, L. 2019. More adaptive algorithms for tracking the best expert. *arXiv preprint*.

Orabona, F., and Pál, D. 2018. Scale-free online learning. *Theoretical Computer Science* 716:50–69.

Orabona, F.; Cesa-Bianchi, N.; and Gentile, C. 2012. Beyond logarithmic bounds in online learning. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 823–831.

Ordentlich, E., and Cover, T. M. 1998. The cost of achieving the best portfolio in hindsight. *Mathematics of Operations Research* 23(4):960–982.

Reddi, S. J.; Kale, S.; and Kumar, S. 2018. On the convergence of adam and beyond. In *Proceedings of 6th International Conference on Learning Representations*.

Shalev-Shwartz, S., et al. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* 4(2):107–194.

Srebro, N.; Sridharan, K.; and Tewari, A. 2010. Smoothness, low noise and fast rates. In *Advances in neural information processing systems 22*, 2199–2207.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 26–31.

van Erven, T., and Koolen, W. M. 2016. Metagrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29*, 3666–3674.

Wang, G.; Lu, S.; and Zhang, L. 2019. Adaptivity and optimality: A universal algorithm for online convex optimization. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*.

Wang, G.; Zhao, D.; and Zhang, L. 2018. Minimizing adaptive regret with one gradient per iteration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2762–2768.

Zhang, L.; Liu, T.-Y.; and Zhou, Z.-H. 2019. Adaptive regret of convex and smooth functions. In *Proceedings of the 36th International Conference on Machine Learning*, 7414–7423.

Zhang, L.; Lu, S.; and Zhou, Z.-H. 2018. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31*, 1323–1333.

Zhao, P.; Zhang, Y.-J.; Zhang, L.; and Zhou, Z.-H. 2019. Dynamic regret of convex and smooth functions. *arXiv preprint*.

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