A Simple yet Universal Strategy for Online Convex Optimization

Lijun Zhang [1 2] **Guanghui Wang** [1] **Jinfeng Yi** [3] **Tianbao Yang** [4]

Abstract

Recently, several universal methods have been proposed for online convex optimization, and attain minimax rates for multiple types of convex functions simultaneously. However, they need to design and optimize one surrogate loss for each type of functions, making it difficult to exploit the structure of the problem and utilize existing algorithms. In this paper, we propose a simple strategy for universal online convex optimization, which avoids these limitations. The key idea is to construct a set of experts to process the *original* online functions, and deploy a meta-algorithm over the *linearized* losses to aggregate predictions from experts. Specifically, the meta-algorithm is required to yield a second-order bound with excess losses, so that it can leverage strong convexity and exponential concavity to control the meta-regret. In this way, our strategy inherits the theoretical guarantee of *any* expert designed for strongly convex functions and exponentially concave functions, up to a double logarithmic factor. As a result, we can plug in off-the-shelf online solvers as black-box experts to deliver problem-dependent regret bounds. For general convex functions, it maintains the minimax optimality and also achieves a small-loss bound.

1. Introduction

Online convex optimization (OCO) has become a leading online learning framework, and is able to model various real-world problems such as online routing and spam filtering (Hazan, 2016). OCO can be seen as a structured repeated game, with the following protocol. At each round t, the online learner chooses $\mathbf{x}_t$ from a convex set $\mathcal{X}$. After committing to this choice, a convex cost function $f_t : \mathcal{X} \mapsto \mathbb{R}$ is revealed, and the loss incurred by the learner is $f_t(\mathbf{x}_t)$. The learner aims to minimize the cumulative loss over T rounds, i.e., $\sum_{t=1}^T f_t(\mathbf{x}_t)$, which is equivalent to minimizing the regret:

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

defined as the excess loss suffered by the learner compared to the minimum loss of any fixed choice.

In the literature, there are plenty of algorithms for OCO (Cesa-Bianchi & Lugosi, 2006; Shalev-Shwartz, 2011). For example, online gradient descent (OGD) with $O(1/\sqrt{t})$ step size achieves $O(\sqrt{T})$ regret for general convex functions (Zinkevich, 2003); OGD with $O(1/t)$ step size attains $O(\log T)$ regret for strongly convex functions (Shalev-Shwartz et al., 2007); online Newton step (ONS) enjoys $O(d \log T)$ regret for exponentially concave (abbr. exp-concave) functions, where d is the dimensionality (Hazan et al., 2007). Besides, there exist more powerful online algorithms such as ADAGRAD (Duchi et al., 2011) that are equipped with problem-dependent regret bounds (Srebro et al., 2010; Chiang et al., 2012; Kingma & Ba, 2015; Mukkamala & Hein, 2017; Reddi et al., 2018), which become tighter when the problem has special structures. Although we have rich theories for OCO, its application requires heavy domain knowledge: Users must know the type of functions in order to select an appropriate algorithm, and when dealing with strongly convex functions and exp-concave functions, they also need to estimate the moduli of strong convexity and exponential concavity.

The lack of universality of previous algorithms motivates the development of universal methods for OCO (Bartlett et al., 2008; Do et al., 2009). One milestone is MetaGrad of van Erven & Koolen (2016), which can handle general convex functions as well as exp-concave functions. Later, Wang et al. (2019) propose Maler, which further supports strongly convex functions explicitly. In a subsequent work, Wang et al. (2020b) develop UFO, which exploits smoothness to deliver small-loss regret bounds, i.e., regret bounds that depend on the minimal loss. However, the three aforementioned methods need to design one surrogate loss for each possible type of functions, which is both tedious and challenging. Furthermore, because they rely on surrogate losses,

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China [2]Peng Cheng Laboratory, Shenzhen, China [3]Frontis.AI, Beijing, China [4]Department of Computer Science, The University of Iowa, Iowa City, USA. Correspondence to: Lijun Zhang <zhanglj@lamda.nju.edu.cn>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

it is difficult to produce problem-dependent regret bounds, except the small-loss one.

To avoid the above limitations, we propose a simple yet universal strategy for online convex optimization. First, we create a set of experts to handle the uncertainty of the type of online functions and (possibly) the associated parameters. When facing unknown continuous variables, we discretize them by constructing a geometric series to cover the range of their values. The experts process the *original* online functions directly so that they can exploit the special structure of these functions. Second, we run a meta-algorithm to track the best expert on the fly, but use the *linearized* losses to measure the performance. To benefit from strong convexity and exponential concavity, we require the meta-algorithm to yield a second-order bound with excess losses, and choose Adapt-ML-Prod (Gaillard et al., 2014) as an example. Specifically, let ℓ_t and ℓ_t^i be the loss of the meta-algorithm and the i-th expert in the t-th round, respectively. The regret of the meta-algorithm satisfies

$$\sum_{t=1}^{T}(\ell_t - \ell_t^i) = O\left(\sqrt{\sum_{t=1}^{T}(\ell_t - \ell_t^i)^2}\right), \forall i \tag{2}$$

where for brevity we drop the dependence on the number of experts.

By incorporating existing methods for strongly convex functions, exp-concave functions and general convex functions as experts, we obtain a universal algorithm with the following properties.

- For strongly convex functions, our algorithm is agnostic to the modulus of strong convexity, at the price of maintaining $O(\log T)$ experts. More importantly, it inherits the regret bound of *any* expert designed for strongly convex functions, with a negligible additive factor of $O(\log\log T)$. As a result, we can deliver any problem-dependent or independent regret bound, without prior knowledge of strong convexity.
- For exp-concave functions, the above statements are also true.
- For general convex functions, the theoretical guarantee is a mix of the regret bound of the expert and the second-order bound of Adapt-ML-Prod. When the functions are convex and smooth, it yields a small-loss bound.

Compared to previous universal methods (van Erven & Koolen, 2016; Wang et al., 2019; 2020b), our algorithm has the following advantages.

- It decouples the loss used by the expert-algorithm and that by the meta-algorithm. In this way, we can directly utilize existing online algorithms as black-box subroutines, and do not need to design surrogate losses.
- For strongly convex functions and exp-concave functions, the regret bound of our algorithm achieves *the best of all worlds*, provided that both the domain and gradients are bounded.

2. Related Work

In this section, we review the related work in OCO, including traditional algorithms, universal algorithms, and parameter-free algorithms.

2.1. Traditional Algorithms

For general convex functions, the most popular algorithm is online gradient descent (OGD), which attains $O(\sqrt{T})$ regret by setting the step size as $\eta_t = O(1/\sqrt{t})$ (Zinkevich, 2003). For λ-strongly convex functions, the regret bound can be improved to $O(\frac{1}{\lambda}\log T)$ by running OGD with step size $\eta_t = O(1/[\lambda t])$ (Shalev-Shwartz et al., 2007). For α-exp-concave functions, online Newton step (ONS), with prior knowledge of the parameter α, achieves $O(\frac{d}{\alpha}\log T)$ regret, where d is the dimensionality (Hazan et al., 2007). These regret bounds for general convex functions, strongly convex functions, and exp-concave functions are known to be minimax optimal (Ordentlich & Cover, 1998; Abernethy et al., 2008), which means that they cannot be improved in the worst case. However, these bounds only exhibit the relationship with problem-independent quantities, such as the time horizon T and the dimensionality d, and thus do not reflect the property of the online problem at hand.

To exploit the structure of the problem, various problem-dependent (or data-dependent) regret bounds have been established recently (Srebro et al., 2010; Duchi et al., 2010; 2011; Chiang et al., 2012; Orabona et al., 2012; Tieleman & Hinton, 2012; Zeiler, 2012; Yang et al., 2014; Kingma & Ba, 2015; Mukkamala & Hein, 2017; Reddi et al., 2018; Wang et al., 2020a). The problem-dependent bounds reduce to the minimax rates in the worst case, but can be better under favorable conditions.

One well-known result is the small-loss bound which is very popular in the studies of online learning (Littlestone & Warmuth, 1994; Auer et al., 2002; Shalev-Shwartz, 2007; Luo & Schapire, 2015). When the functions are smooth and nonnegative, the regret for general convex functions, λ-strongly convex functions, and α-exp-concave functions can be upper bounded by $O(\sqrt{L_T^*})$, $O(\frac{1}{\lambda}\log L_T^*)$, and $O(\frac{d}{\alpha}\log L_T^*)$ respectively, where

$$L_T^* = \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \tag{3}$$

is the cumulative loss of the best point in $\mathcal{X}$ (Srebro et al., 2010; Orabona et al., 2012; Zhang et al., 2019; Wang et al., 2020b). These small-loss bounds could be much tighter when L_T^* is small, and still ensure the minimax optimality otherwise. Another problem-dependent guarantee is the gradient-variation bound for smooth functions (Chiang et al., 2012; Yang et al., 2014; Mohri & Yang, 2016), which replaces L_T^* in the upper bounds with the gradient variation:

$$V_T = \sum_{t=1}^{T} \max_{\mathbf{x}\in\mathcal{X}} \|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|^2. \tag{4}$$

Then, the regret bounds become smaller if the online functions evolve gradually.

Besides smoothness, it is also possible to exploit other structural properties of the functions, such as the sparsity of gradients. One representative work is ADAGRAD (Duchi et al., 2010; 2011), which incorporates knowledge of the geometry of the data observed in earlier iterations to perform more informative gradient-based learning. Let $\mathbf{g}_{1:T,j}$ be the vector obtained by concatenating the j-th element of the gradient sequence $\nabla f_1(\mathbf{x}_1), \ldots, \nabla f_T(\mathbf{x}_T)$. ADAGRAD achieves

$$O\left(\sum_{j=1}^{d} \|\mathbf{g}_{1:T,j}\|\right) \text{ and } O\left(\frac{1}{\lambda}\sum_{j=1}^{d} \log \|\mathbf{g}_{1:T,j}\|\right)$$

regret for general convex functions and λ-strongly convex functions, respectively (Duchi et al., 2010). These two bounds match the minimax rates in the worst case, and become tighter when gradients are sparse. Since the seminal work of ADAGRAD, a series of problem-dependent online algorithms have been developed, including RMSprop (Tieleman & Hinton, 2012; Mukkamala & Hein, 2017) and Adam (Kingma & Ba, 2015; Reddi et al., 2018).

Although there exist abundant algorithms and theories for OCO, how to choose them in practice is a nontrivial task. To ensure good performance, we not only need to know the type of functions, but also need to estimate the moduli of strong convexity and exponential concavity. The requirement of human intervention restricts the application of OCO to real-world problems, and motivates the development of universal algorithms for OCO.

2.2. Universal Algorithms

The first universal method for OCO is adaptive online gradient descent (AOGD) (Bartlett et al., 2008), which interpolates between the $O(\sqrt{T})$ regret of general convex functions and the $O(\log T)$ regret of strongly convex functions automatically. However, AOGD needs to know the modulus of strong convexity in each round, and does not support exp-concave functions. Do et al. (2009) develop a proximal extension of AOGD, but it suffers the same limitations.

The studies of universal methods are advanced by the MetaGrad algorithm of van Erven & Koolen (2016), which adapts to a much broader class of functions, including general convex functions and exp-concave functions. Under the framework of learning with expert advice (Cesa-Bianchi & Lugosi, 2006), MetaGrad is a two-layer algorithm consisting of a set of experts and a meta-algorithm. To handle exponential concavity, each expert minimizes one surrogate loss

$$\ell_{t,\eta}^{exp}(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 \left[(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t\right]^2 \tag{5}$$

parameterized by the step size η, where $\mathbf{g}_t = \nabla f_t(\mathbf{x}_t)$. MetaGrad maintains $O(\log T)$ experts to minimize (5) with different step sizes, and combines their predictions with a meta-algorithm. In this way, it attains $O(\frac{d}{\alpha}\log T)$ regret for α-exp-concave functions, without knowing the value of α. At the same time, MetaGrad also achieves an $O(\sqrt{T\log\log T})$ regret bound for general convex functions. Although we can treat strongly convex functions as exp-concave and obtain an $O(d\log T)$ regret bound, there exists an $O(d)$ gap from the minimax rate of strongly convex functions (Abernethy et al., 2008).

To deal with strongly convex functions, Wang et al. (2019) design another surrogate loss

$$\ell_{t,\eta}^{str}(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 G^2 \|\mathbf{x}_t - \mathbf{x}\|^2 \tag{6}$$

where G is an upper bound of the norm of gradients. Their proposed method, named as Maler, introduces additional $O(\log T)$ experts to optimize (6), and obtains $O(\frac{1}{\lambda}\log T)$ regret for λ-strongly convex functions. Similar to MetaGrad, it gets rid of the priori knowledge of strong convexity. Wang et al. (2019) further propose the following surrogate loss:

$$\ell_{t,\eta}^{con}(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 G^2 D^2 \tag{7}$$

where D is an upper bound of the diameter of $\mathcal{X}$, and obtain the optimal $O(\sqrt{T})$ regret for general convex functions. To exploit smoothness, Wang et al. (2020b) propose the following surrogate loss

$$\ell_{t,\eta}^{str,smo}(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 \|\mathbf{g}_t\|^2 \|\mathbf{x}_t - \mathbf{x}\|^2 \tag{8}$$

for strongly convex and smooth functions, reuse the surrogate loss in (5) for exp-concave and smooth functions, and introduce the following surrogate loss

$$\ell_{t,\eta}^{con,smo}(\mathbf{x}) = -\eta(\mathbf{x}_t - \mathbf{x})^\top \mathbf{g}_t + \eta^2 \|\mathbf{g}_t\|^2 D^2 \tag{9}$$

for convex and smooth functions. Under the smoothness condition, their algorithm, namely UFO, achieves $O(\frac{1}{\lambda}\log L_T^*)$, $O(\frac{d}{\alpha}\log L_T^*)$, and $O(\sqrt{L_T^*})$ regret bounds for λ-strongly convex functions, α-exp-concave functions, and general convex functions respectively, where L_T^* is defined in (3).

Besides supporting more types of functions, MetaGrad was also extended to avoid the knowledge of the Lipschitz constant G. Specifically, Mhammedi et al. (2019) first design a basic algorithm called MetaGrad+C, which requires an initial estimate of the Lipschitz constant, and then use a restarting scheme to set this parameter online. In this way, the final algorithm, named as MetaGrad+L, adapts to the Lipschitz hyperparameter automatically. Furthermore, Mhammedi et al. (2019) also remove the need to specify the number of rounds in advance. Recently, van Erven et al. (2021) propose a refined version of MetaGrad+L, which only restarts the meta-algorithm but not the experts.

From the above discussion, we observe that state-of-the-art universal methods (van Erven & Koolen, 2016; Wang et al., 2019; 2020b) all rely on the construction of surrogate losses, which brings two issues.

1. They have to design one surrogate loss for each possible type of functions, which is very challenging. That is because we need to ensure that the regret in terms of surrogate losses can be converted back to the regret of original functions.
2. Since all the experts receive the surrogate losses instead of the original functions, it is difficult to exploit the structure of the particular problem instance. Of course, one could use problem-dependent algorithms to optimize the surrogate losses, but they may not share the same structure with the original functions. Except the small-loss bound (Wang et al., 2020b), it is unclear how to generate other problem-dependent regret bounds.

2.3. Parameter-Free Algorithms

A parallel line of research is parameter-free online learning, which aims to design algorithms that avoid tuning parameters. The motivation is that traditional online algorithms, such as OGD (Zinkevich, 2003), require some prior knowledge about the comparator and/or gradients to set their parameters, e.g., the step size. To avoid this limitation, parameter-free algorithms adapt to both the comparator and the sequence of gradients automatically (Orabona, 2014; Orabona & Pál, 2016; Cutkosky & Boahen, 2016; 2017; Foster et al., 2017; Orabona & Pál, 2018; Cutkosky & Orabona, 2018; Cutkosky, 2019; Mhammedi & Koolen, 2020). The study of parameter-free algorithms mainly focuses on general convex functions, and is complementary to the development of universal algorithms.

3. Our Methods

In this section, we first introduce necessary assumptions and definitions, and then explain the motivation of our paper. Next, we elaborate the proposed strategy for OCO, and then present theoretical guarantees for strongly convex functions, exp-concave functions and general convex functions. Finally, we discuss parameter-free extensions. Due to space limitations, all the proofs are deferred to Appendix A.

3.1. Preliminaries

We start with two common assumptions of OCO (Zinkevich, 2003; van Erven & Koolen, 2016; Zhao et al., 2020; Zhang et al., 2021).

Assumption 3.1. The gradients of all functions are bounded by G, i.e.,

$$\max_{\mathbf{x}\in\mathcal{X}} \|\nabla f_t(\mathbf{x})\| \leq G, \ \forall t \in [T]. \tag{10}$$

Assumption 3.2. The diameter of the domain $\mathcal{X}$ is bounded by D, i.e.,

$$\max_{\mathbf{x},\mathbf{y}\in\mathcal{X}} \|\mathbf{x}-\mathbf{y}\| \leq D. \tag{11}$$

To facilitate presentations, we assume the values of G and D are known beforehand, and will return to this issue in Section 3.7.

Then, we introduce the definitions of strongly convex functions and exp-concave functions (Boyd & Vandenberghe, 2004; Cesa-Bianchi & Lugosi, 2006).

Definition 3.3. A function $f:\mathcal{X}\mapsto\mathbb{R}$ is λ-strongly convex if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\lambda}{2}\|\mathbf{y}-\mathbf{x}\|^2, \tag{12}$$

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

Definition 3.4. A function $f:\mathcal{X}\mapsto\mathbb{R}$ is α-exp-concave if $\exp(-\alpha f(\cdot))$ is concave over $\mathcal{X}$.

In the analysis, we will make use of the following property of exp-concave functions (Hazan et al., 2007, Lemma 3).

Lemma 3.5. *Suppose $f:\mathcal{X}\mapsto\mathbb{R}$ is α-exp-concave. Under Assumptions 3.1 and 3.2, we have*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\beta}{2}\langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle^2, \tag{13}$$

for all $\mathbf{x},\mathbf{y}\in\mathcal{X}$, where $\beta=\frac{1}{2}\min\{\frac{1}{4GD},\alpha\}$.

3.2. The Motivation

Similar to existing universal methods (van Erven & Koolen, 2016; Wang et al., 2019; 2020b), we follow the framework of learning with expert advice. Specifically, we construct a set of experts for each type of functions and use a meta-algorithm to aggregate their predictions. The difference is that our universal strategy adopts two novel ingredients:

(i) the experts process the *original* functions so that they can exploit the structure of the problem instance to deliver problem-dependent regret bounds;

(ii) the meta-algorithm chooses the *linearized* losses and makes use of second-order bounds to control the meta-regret.

In the following, we take strongly convex functions as an example to explain the motivation.

Let $\mathbf{x}_t$ and $\mathbf{u}_t$ be the output of the meta-algorithm and an expert in the t-th round, respectively. The regret of the meta-algorithm can be decomposed as the sum of the meta-regret and the expert-regret:

$$\begin{aligned}&\sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})\\ =&\underbrace{\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)}_{:=\text{meta-regret}} + \underbrace{\sum_{t=1}^T f_t(\mathbf{u}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})}_{:=\text{expert-regret}}.\end{aligned}$$

To bound the expert-regret, we can directly utilize theoretical guarantees of the expert-algorithm. So, the key is how to ensure small meta-regret.

Instead of using the original function, our meta-algorithm chooses the linearized loss

$$l_t(\mathbf{x}) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t\rangle \tag{14}$$

to measure the performance of the expert. From Definition 3.3, we have the following relationship between the meta-regret in terms of $f_t(\cdot)$ and the meta-regret in terms of $l_t(\cdot)$:

$$\begin{aligned}&\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t)\\ \overset{(12)}{\leq}&\sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u}_t\rangle - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}_t\|^2 \qquad (15)\\ \overset{(14)}{=}&\sum_{t=1}^T \big(l_t(\mathbf{x}_t) - l_t(\mathbf{u}_t)\big) - \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}_t\|^2.\end{aligned}$$

Since we require the meta-algorithm to yield a second-order bound in the form of (2), the meta-regret in terms of $l_t(\cdot)$ becomes

$$\begin{aligned}&\sum_{t=1}^T \big(l_t(\mathbf{x}_t) - l_t(\mathbf{u}_t)\big)\\ \overset{(2)}{=}&O\left(\sqrt{\sum_{t=1}^T \big(l_t(\mathbf{x}_t) - l_t(\mathbf{u}_t)\big)^2}\right) \qquad (16)\\ \overset{(14)}{=}&O\left(\sqrt{\sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{u}_t\rangle^2}\right).\end{aligned}$$

From Assumption 3.1, we further have

$$\begin{aligned}&\sum_{t=1}^T \big(l_t(\mathbf{x}_t) - l_t(\mathbf{u}_t)\big)\\ \overset{(10),(16)}{=}&O\left(\sqrt{G^2\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}_t\|^2}\right) \qquad (17)\\ =&O\left(\frac{G^2}{\lambda}\right) + \frac{\lambda}{2}\sum_{t=1}^T \|\mathbf{x}_t - \mathbf{u}_t\|^2.\end{aligned}$$

Combining (15) and (17), we have the following meta-regret

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{u}_t) = O\left(\frac{G^2}{\lambda}\right).$$

which is small and thus does not affect the optimality of the algorithm.

From the above discussions, we have the following conclusion: It is the second-order bound in (16) that makes it possible to exploit the negative term in (15), which arises from the strong convexity. Based on Lemma 3.5, we observe a similar phenomenon for exp-concave functions.

3.3. Our Universal Strategy

Our universal strategy for online convex optimization (USC) is summarized in Algorithm 1. We will consider three types of convex functions: strongly convex functions, exp-concave functions, and general convex functions.

Let $\mathcal{A}_{str}$ be the set of candidate algorithms designed for strongly convex functions, and $\mathcal{P}_{str}$ be the set of possible values of the modulus of strong convexity. Note that although the modulus of strong convexity is continuous, we can construct a finite set $\mathcal{P}_{str}$ to approximate its value, which will be elaborated in Section 3.4. For each algorithm $A \in \mathcal{A}_{str}$ and each $\lambda \in \mathcal{P}_{str}$, we create an expert $E(A, \lambda)$ by invoking algorithm A with parameter λ, and add it to the set $\mathcal{E}$ consisting of experts (Steps 3-8). Similarly, let $\mathcal{A}_{exp}$ be the set of algorithms designed for exp-concave functions, and $\mathcal{P}_{exp}$ be the set of values of the modulus of exponential concavity. For each algorithm $A \in \mathcal{A}_{exp}$ and each $\alpha \in \mathcal{P}_{exp}$, we instantiate an expert $E(A, \alpha)$ by running algorithm A with parameter α, and add it to $\mathcal{E}$ (Steps 9-14). General convex functions can be handled more easily, and we denote the set of algorithms designed for them by $\mathcal{A}_{con}$. Then, we simply create an expert $E(A)$ for each $A \in \mathcal{A}_{con}$ and add it to $\mathcal{E}$ (Steps 15-18).

Next, we deploy a meta-algorithm to track the best expert on the fly. Here, one can use any method that enjoys second-order bounds with excess losses (Koolen & Erven, 2015; Mhammedi et al., 2019). We choose Adapt-ML-Prod (Gaillard et al., 2014) because it is more simple and already

Algorithm 1 A Universal Strategy for Online Convex Optimization (USC)

```
1: Input: $\mathcal{A}_{str}$, $\mathcal{A}_{exp}$, $\mathcal{A}_{con}$, $\mathcal{P}_{str}$ and $\mathcal{P}_{exp}$
2: Initialize $\mathcal{E} = \emptyset$
3: for each algorithm $A \in \mathcal{A}_{str}$ do
4:     for each $\lambda \in \mathcal{P}_{str}$ do
5:         Create an expert $E(A, \lambda)$
6:         $\mathcal{E} = \mathcal{E} \cup E(A, \lambda)$
7:     end for
8: end for
9: for each algorithm $A \in \mathcal{A}_{exp}$ do
10:    for each $\alpha \in \mathcal{P}_{exp}$ do
11:        Create an expert $E(A, \alpha)$
12:        $\mathcal{E} = \mathcal{E} \cup E(A, \alpha)$
13:    end for
14: end for
15: for each algorithm $A \in \mathcal{A}_{con}$ do
16:    Create an expert $E(A)$
17:    $\mathcal{E} = \mathcal{E} \cup E(A)$
18: end for
19: for $t = 1$ to $T$ do
20:    Calculate the weight $p_t^i$ of each expert $E^i$ by (21)
21:    Receive $\mathbf{x}_t^i$ from each expert $E^i$ in $\mathcal{E}$
22:    Output the weighted average $\mathbf{x}_t = \sum_{i=1}^{|\mathcal{E}|} p_t^i \mathbf{x}_t^i$
23:    Observe the loss function $f_t(\cdot)$
24:    Send the required information of $f_t(\cdot)$ to each expert
       in $\mathcal{E}$
25: end for
```

satisfies our requirements. To simplify notations, we assume that experts in $\mathcal{E}$ are ordered, and use E^i to denote the i-th expert. In the t-th round, we denote by p_t^i the weight assigned to E^i, and $\mathbf{x}_t^i$ the prediction of E^i. The weights are determined according to Adapt-ML-Prod (Step 20). After receiving predictions from all experts in $\mathcal{E}$ (Step 21), USC submits the weighted average (Step 22):

$$\mathbf{x}_t = \sum_{i=1}^{|\mathcal{E}|} p_t^i \mathbf{x}_t^i. \tag{18}$$

Then, it observes the loss function $f_t(\cdot)$ and sends the required information to all experts so that they can update their predictions (Steps 23-24). If the expert is a first-order algorithm, we only need to send the gradient of $f_t(\cdot)$. Thus, USC may query the gradient *multiple* times in each round. It is worth to mention that allowing the online learner to observe multiple gradients does not affect the minimax rates in the full-information setting (Abernethy et al., 2008), where the leaner can observe the entire function. So, we can still use existing minimax rates to verify the optimality of USC.

Finally, we briefly explain how to calculate the weights of experts, i.e., p_t^i's. As we explained before, the meta-algorithm uses the linearized loss in (14) to measure the performance of each expert. In particular, the loss of E^i is given by

$$l_t^i := l_t(\mathbf{x}_t^i) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t^i - \mathbf{x}_t \rangle.$$

Under Assumptions 3.1 and 3.2, we have

$$|l_t^i| \leq \|\nabla f_t(\mathbf{x}_t)\| \|\mathbf{x}_t^i - \mathbf{x}_t\| \overset{(10),(11)}{\leq} GD.$$

Because Adapt-ML-Prod requires the loss to lie in $[0, 1]$, we normalize l_t^i in the following way

$$\ell_t^i = \frac{\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t^i - \mathbf{x}_t \rangle + GD}{2GD} \in [0, 1]. \tag{19}$$

Then, the loss of the meta-algorithm suffered in the t-th round becomes

$$\ell_t = \sum_{i=1}^{|\mathcal{E}|} p_t^i \ell_t^i \overset{(18),(19)}{=} \frac{1}{2}. \tag{20}$$

According to Adapt-ML-Prod (Gaillard et al., 2014), the weight of expert E^i is determined by

$$p_t^i = \frac{\eta_{t-1}^i w_{t-1}^i}{\sum_{j=1}^{|\mathcal{E}|} \eta_{t-1}^j w_{t-1}^j} \tag{21}$$

where

$$\eta_{t-1}^i = \min\left\{ \frac{1}{2}, \sqrt{\frac{\ln |\mathcal{E}|}{1 + \sum_{s=1}^{t-1} (\ell_s - \ell_s^i)^2}} \right\}, \ t \geq 1, \tag{22}$$

$$w_{t-1}^i = \left(w_{t-2}^i \big(1 + \eta_{t-2}^i (\ell_{t-1} - \ell_{t-1}^i)\big) \right)^{\frac{\eta_{t-1}^i}{\eta_{t-2}^i}}, \ t \geq 2.$$

In the beginning, we set $w_0^i = 1/|\mathcal{E}|$. As indicated by (22), Gaillard et al. (2014) use an adaptive way to set multiple time-varying learning rates.

3.4. Strongly Convex Functions

We present the regret bound of our strategy when encountering strongly convex functions. To apply USC in Algorithm 1, we need to specify $\mathcal{A}_{str}$, the set of candidate algorithms, and $\mathcal{P}_{str}$, the set of possible values of the modulus of strong convexity. To build $\mathcal{A}_{str}$, we can utilize any existing algorithm for online strongly convex optimization, such as

- OGD for strongly convex functions (SC-OGD) (Shalev-Shwartz et al., 2007);
- ADAGRAD for strongly convex functions (Duchi et al., 2010);
- Online extra-gradient descent (OEGD) for strongly convex and smooth functions (Chiang et al., 2012);
- SC-RMSProp (Mukkamala & Hein, 2017);

- SAdam (Wang et al., 2020a);
- S^{2}OGD for strongly convex and smooth functions (Wang et al., 2020b).

Chiang et al. (2012) only investigate OEGD for exp-concave functions and general convex functions, under the smoothness condition. In Appendix B, we extend OEGD to strongly convex functions and obtain a gradient-variation bound of order $O(\log V_T)$, which may be of independent interest.

We proceed to construct $\mathcal{P}_{str}$. Without loss of generality, we assume the unknown modulus λ is both lower bounded and upper bounded. In particular, we assume $\lambda \in [1/T, 1]$, because there is no need to explicitly consider the cases that $\lambda < 1/T$ and $\lambda > 1$, as explained below.

1. The regret bound of strongly convex functions exhibits an inverse dependence on λ. Thus, if $\lambda < 1/T$, the bound becomes at least $\Omega(T)$, which is meaningless. In this case, we cannot benefit from strong convexity and should treat these functions as general convex.
2. From Definition 3.3, we know that λ-strongly convex functions with $\lambda > 1$ are also 1-strongly convex. So, they can be handled as 1-strongly convex, and the resulting bound is optimal up to a constant (i.e., λ) factor.

Based on the interval $[1/T, 1]$, we set $\mathcal{P}_{str}$ to be an exponentially spaced grid with a ratio of 2:

$$\mathcal{P}_{str} = \left\{ \frac{1}{T}, \frac{2}{T}, \frac{2^2}{T}, \cdots, \frac{2^N}{T} \right\}, \ N = \lceil \log_2 T \rceil. \tag{23}$$

$\mathcal{P}_{str}$ can approximate λ well in the sense that for any $\lambda \in [1/T, 1]$, there must exist a $\widehat{\lambda} \in \mathcal{P}_{str}$ such that $\widehat{\lambda} \leq \lambda \leq 2\widehat{\lambda}$.

In the following, we denote by $R(A, \widehat{\lambda})$ the regret bound, predicted by theory, of expert $E(A, \widehat{\lambda})$ in Algorithm 1. Note that the expert $E(A, \widehat{\lambda})$ assumes the online functions are $\widehat{\lambda}$-strongly convex, which is true since $\widehat{\lambda} \leq \lambda$. Thus, the regret bound $R(A, \widehat{\lambda})$ is *valid*, and it is also *tight* because $\lambda \leq 2\widehat{\lambda}$. We have the following theoretical guarantee.

Theorem 3.6. *Under Assumptions 3.1 and 3.2, if the online functions are λ-strongly convex with $\lambda \in [1/T, 1]$, Algorithm 1 satisfies*

$$\begin{aligned} &\sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x}) \\ \leq &\min_{A \in \mathcal{A}_{str}} R(A, \widehat{\lambda}) + 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}} \right) + \frac{\Gamma^2 G^2}{2\lambda \ln |\mathcal{E}|} \\ = &\min_{A \in \mathcal{A}_{str}} R(A, \widehat{\lambda}) + O\left(\frac{\log \log T}{\lambda} \right) \end{aligned}$$

where $\widehat{\lambda} \in \mathcal{P}_{str}$, $\widehat{\lambda} \leq \lambda \leq 2\widehat{\lambda}$, and

$$\begin{aligned} \Gamma = &3 \ln |\mathcal{E}| + \ln \left(1 + \frac{|\mathcal{E}|}{2e} \big(1 + \ln(T+1)\big) \right) \\ \overset{(25)}{=} &O(\log \log T). \end{aligned} \tag{24}$$

Remark: To reveal the order of the upper bound, we assume the number of candidate algorithms is small, so $|\mathcal{A}_{str}|$, $|\mathcal{A}_{exp}|$ and $|\mathcal{A}_{con}|$ are all small constants. Thus,

$$\begin{aligned} |\mathcal{E}| = &|\mathcal{A}_{str}| \cdot |\mathcal{P}_{str}| + |\mathcal{A}_{exp}| \cdot |\mathcal{P}_{exp}| + |\mathcal{A}_{con}| \\ \overset{(23),(26)}{=} &O(\log T) \end{aligned} \tag{25}$$

which is used in (24). When both the domain and gradients are bounded, Theorem 3.6 shows that USC achieves *the best of all worlds* for strongly convex functions, up to an additive factor of $O(\log \log T)$.

Remark: The computational complexity of an expert per iteration is generally independent from T, but may depend on the dimensionality d (Duchi et al., 2010). To simplify discussions, we hide the dependence on d, and assume the complexity is $O(1)$ per iteration. Since USC maintains $|\mathcal{E}| = O(\log T)$ experts, its computational complexity is $O(\log T)$ per iteration, which is the same as that of previous methods (van Erven & Koolen, 2016; Wang et al., 2019; 2020b).

To be more concrete, we use the small-loss bound and the gradient-variation bound for smooth functions to give an example. To this end, we need an additional assumption (Srebro et al., 2010).

Assumption 3.7. All the online functions are nonnegative, and H-smooth over $\mathcal{X}$.

By using OEGD (Chiang et al., 2012) and S^{2}OGD (Wang et al., 2020b) as experts, we have the following corollary.

Corollary 3.8. *Under Assumptions 3.1, 3.2, and 3.7, if the online functions are λ-strongly convex with $\lambda \in [1/T, 1]$, we have*

$$\begin{aligned} &\sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x}) \\ = &O\left(\frac{1}{\lambda} \Big(\min(\log L_T^*, \log V_T) + \log \log T \Big) \right) \end{aligned}$$

where L_T^* and V_T are defined in (3) and (4) respectively, provided OEGD, S^{2}OGD $\in \mathcal{A}_{str}$.

3.5. Exp-concave Functions

We move to exp-concave functions, and use the following algorithms to build $\mathcal{A}_{exp}$:

- Online Newton step (ONS) (Hazan et al., 2007);

- ONS for exp-concave and smooth functions (Orabona et al., 2012);
- OEGD for exp-concave and smooth functions (Chiang et al., 2012).

Following the same arguments as in Section 3.4, we also assume the modulus of exponential concavity α lies in $[1/T, 1]$, and use the same geometric series to construct $\mathcal{P}_{exp}$ as

$$\mathcal{P}_{exp} = \left\{ \frac{1}{T}, \frac{2}{T}, \frac{2^2}{T}, \cdots, \frac{2^N}{T} \right\}, \ N = \lceil \log_2 T \rceil. \quad (26)$$

Then, for any $\alpha \in [1/T, 1]$, there must exist an $\widehat{\alpha} \in \mathcal{P}_{exp}$ such that $\widehat{\alpha} \le \alpha \le 2\widehat{\alpha}$.

We denote by $R(A, \widehat{\alpha})$ the regret bound of expert $E(A, \widehat{\alpha})$ in Algorithm 1. Similarly, $R(A, \widehat{\alpha})$ is both valid and tight. We have the following guarantee for exp-concave functions, which is analogous to Theorem 3.6.

Theorem 3.9. *Under Assumptions 3.1 and 3.2, if the online functions are α-exp-concave with $\alpha \in [1/T, 1]$, Algorithm 1 satisfies*

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \\
\le &\min_{A \in \mathcal{A}_{exp}} R(A, \widehat{\alpha}) + 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}} \right) + \frac{\Gamma^2}{2\beta \ln |\mathcal{E}|} \\
= &\min_{A \in \mathcal{A}_{exp}} R(A, \widehat{\alpha}) + O\left(\frac{\log \log T}{\alpha} \right)
\end{aligned}$$

where $\widehat{\alpha} \in \mathcal{P}_{exp}$, $\widehat{\alpha} \le \alpha \le 2\widehat{\alpha}$, $\beta = \frac{1}{2} \min\{\frac{1}{4GD}, \alpha\}$, and Γ is defined in (24).

Remark: Similar to the case of strongly convex functions, USC inherits the regret bound of *any* expert designed for exp-concave functions, with a negligible double logarithmic factor. By using ONS (Orabona et al., 2012) and OEGD (Chiang et al., 2012) as experts, we obtain the best of the small-loss bound and the gradient-variation bound, up to a double logarithmic factor.

Corollary 3.10. *Under Assumptions 3.1, 3.2, and 3.7, if the online functions are α-exp-concave with $\alpha \in [1/T, 1]$, we have*

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \\
=&O\left(\frac{1}{\alpha} \Big(d \min(\log L_T^*, \log V_T) + \log \log T \Big) \right)
\end{aligned}$$

where L_T^* and V_T are defined in (3) and (4) respectively, provided ONS, OEGD $\in \mathcal{A}_{exp}$.

3.6. General Convex Functions

Finally, we study general convex functions, and in this case, we have various algorithms to construct $\mathcal{A}_{con}$, such as OGD (Zinkevich, 2003), ADAGRAD (Duchi et al., 2011), OEGD for convex and smooth functions (Chiang et al., 2012), RMSprop (Tieleman & Hinton, 2012), ADADELTA (Zeiler, 2012), Adam (Kingma & Ba, 2015), AO-FTRL (Mohri & Yang, 2016), and SOGD (Zhang et al., 2019).

Let $R(A)$ be the regret bound of expert $E(A)$ in Algorithm 1. The theoretical guarantee of USC for general convex functions is stated below.

Theorem 3.11. *Under Assumptions 3.1 and 3.2, for any sequence of convex functions, Algorithm 1 satisfies*

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \\
\le &\min_{A \in \mathcal{A}_{con}} R(A) \\
&+ 4\Gamma G D + \frac{\Gamma D}{\sqrt{\ln |\mathcal{E}|}} \sqrt{4G^2 + \sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|^2} \\
= &\min_{A \in \mathcal{A}_{con}} R(A) + O\left(\sqrt{T \log \log T} \right)
\end{aligned}$$

where Γ is defined in (24).

Remark: The above theorem is weaker than those for strongly convex functions and exp-concave functions. That is because we cannot eliminate the regret of the meta-algorithm, and the final regret is the sum of the expert-regret and the meta-regret. Nevertheless, Theorem 3.11 still implies a small-loss bound for smooth functions, when SOGD (Zhang et al., 2019) is used as the expert.

Corollary 3.12. *Under Assumptions 3.1, 3.2, and 3.7, for any sequence of convex functions, we have*

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) = O\left(\sqrt{L_T^* \log \log T} \right)$$

where L_T^* is defined in (3), provided SOGD $\in \mathcal{A}_{con}$.

3.7. Parameter-Free Extensions

One limitation of USC is that it uses the product of G and D to normalize the linearized loss in (19) so that Adapt-ML-Prod can be applied. Besides, some experts in Algorithm 1 may also use G and D to tune their parameters. Generally speaking, it is easy to estimate D since it only ties to the domain, but difficult to evaluate G which depends on all the online functions. In this section, we discuss how to extend USC to avoid prior knowledge of G, provided that G is bounded.

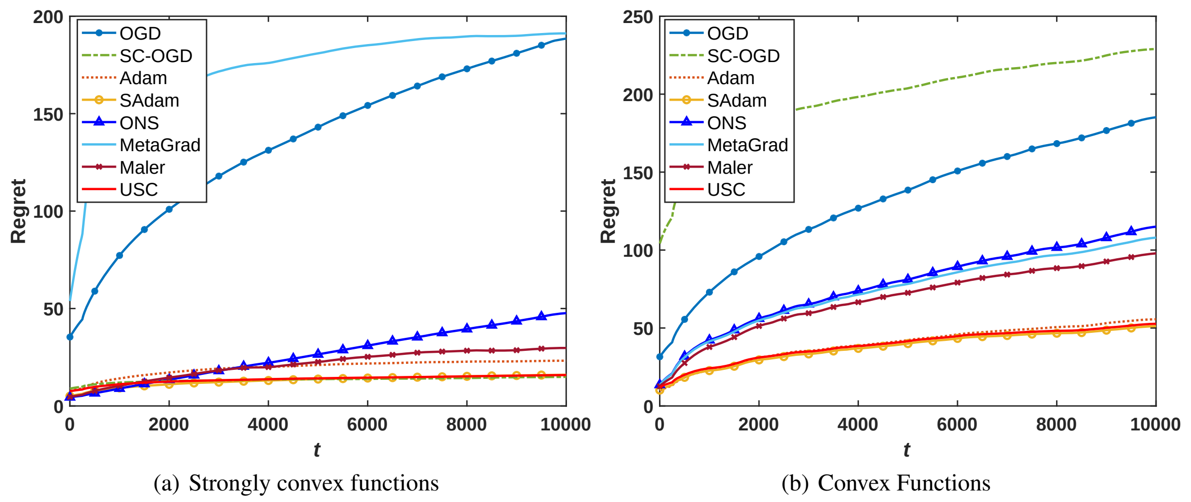

Figure 1. Regret of different methods versus the number of iterations.

For the meta-algorithm, we just need to replace Adapt-ML-Prod with more advanced methods which not only deliver second-order bounds but also adapt to the unknown loss range automatically, e.g., Squint+L (Mhammedi et al., 2019) and MsMwC-Master (Chen et al., 2021). In this way, we can directly use the (unnormalized) linearized loss in (14), and have the same guarantee for the meta-regret.

Next, we choose experts that do not need to know the value of G. For strongly convex functions, we can use OGD with step size $\eta_t = 1/(\lambda t)$ (Shalev-Shwartz et al., 2007), and AOGD (Bartlett et al., 2008). For exp-concave functions, we can choose the Exponentially Weighted Online Optimization (EWOO) algorithm (Hazan et al., 2007), and MetaGrad+L (Mhammedi et al., 2019). For general convex functions, we can use scale-free online learning (Orabona & Pál, 2018) and FreeGrad (Mhammedi & Koolen, 2020).

4. Preliminary Experiments

We conduct preliminary experiments to evaluate the proposed USC, and the detail is provided in Appendix C. We investigate both strongly convex functions and general convex functions, and present the results in Fig. 1. In both cases, the regret of USC is very close to that of the best expert, and smaller than MetaGrad and Maler.

5. Conclusion and Future Work

In this paper, we propose a simple strategy for universal OCO, namely USC, which can handle strongly convex functions, exp-concave functions and general convex functions simultaneously. To deal with the uncertainty of online functions, we construct a set of experts by running existing algorithms with different configurations, and combine them by a meta-algorithm that enjoys a second-order bound with excess losses. The key novelty is to let experts process original functions, and let the meta-algorithm use linearized losses. Thanks to the second-order bound of the meta-algorithm, USC attains *the best of all worlds* for strongly convex functions and exp-concave functions, up to a double logarithmic factor. For general convex functions, it maintains the minimax optimality and can achieve a small-loss bound.

There are several directions for future research. First, USC assumes that both the domain and gradients are bounded, i.e., Assumptions 3.1 and 3.2. We note that there exist online algorithms for unbounded domains or gradients (Chiang et al., 2012; Cutkosky & Boahen, 2016). Thus, a natural work is to design universal algorithms for the unbounded case. Second, USC is designed for the purpose of regret minimization, but regret itself may not be suitable for changing environments (Zhang, 2020; Cesa-Bianchi & Orabona, 2021). To address this limitation, recent developments in online learning have proposed new performance metrics including adaptive regret (Hazan & Seshadhri, 2007; Daniely et al., 2015) and dynamic regret (Zinkevich, 2003; Zhang et al., 2018). In the future, we will investigate how to modify USC to support those stronger notions of regret. Third, USC needs to fix the value of the time horizon T, which is then used to construct $\mathcal{P}_{str}$ and $\mathcal{P}_{exp}$. We will study how to turn USC into an *anytime* algorithm.

Acknowledgements

The authors would like to thank Sijia Chen for helping conduct experiments. This work was partially supported by NSFC (62122037, 61976112), JiangsuSF (BK20200064), and the major key project of PCL (PCL2021A12).

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 21st Annual Conference on Learning Theory*, pp. 415–423, 2008.

Auer, P., Cesa-Bianchi, N., and Gentile, C. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

Bartlett, P. L., Hazan, E., and Rakhlin, A. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems 20*, pp. 65–72, 2008.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Cesa-Bianchi, N. and Orabona, F. Online learning algorithms. *Annual Review of Statistics and Its Application*, 8 (1):165–190, 2021.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

Chen, L., Luo, H., and Wei, C.-Y. Impossible tuning made possible: A new expert algorithm and its applications. In *Proceedings of 34th Conference on Learning Theory*, pp. 1216–1259, 2021.

Chiang, C.-K., Yang, T., Lee, C.-J., Mahdavi, M., Lu, C.-J., Jin, R., and Zhu, S. Online optimization with gradual variations. In *Proceedings of the 25th Annual Conference on Learning Theory*, pp. 6.1–6.20, 2012.

Cutkosky, A. Artificial constraints and hints for unbounded online learning. In *Proceedings of the 32nd Conference on Learning Theory*, pp. 874–894, 2019.

Cutkosky, A. and Boahen, K. Online learning without prior information. In *Proceedings of the 30th Annual Conference on Learning Theory*, pp. 643–677, 2017.

Cutkosky, A. and Boahen, K. A. Online convex optimization with unconstrained domains and losses. In *Advances in Neural Information Processing Systems 29*, pp. 748–756, 2016.

Cutkosky, A. and Orabona, F. Black-box reductions for parameter-free online learning in Banach spaces. In *Proceedings of the 31st Conference On Learning Theory*, pp. 1493–1529, 2018.

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

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

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory*, pp. 257–269, 2010.

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

Foster, D. J., Kale, S., Mohri, M., and Sridharan, K. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems 30*, pp. 6020–6030, 2017.

Gaillard, P., Stoltz, G., and van Erven, T. A second-order bound with excess losses. In *Proceedings of the 27th Conference on Learning Theory*, pp. 176–196, 2014.

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

Hazan, E. and Seshadhri, C. Adaptive algorithms for online decision problems. *Electronic Colloquium on Computational Complexity*, 88, 2007.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Koolen, W. M. and Erven, T. V. Second-order quantile methods for experts and combinatorial games. In *Proceedings of the 28th Conference on Learning Theory*, pp. 1155–1175, 2015.

Littlestone, N. and Warmuth, M. K. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Luo, H. and Schapire, R. E. Achieving all with no parameters: Adanormalhedge. In *Proceedings of the 28th Conference on Learning Theory*, pp. 1286–1304, 2015.

Mhammedi, Z. and Koolen, W. M. Lipschitz and comparator-norm adaptivity in online learning. In *Proceedings of 33rd Conference on Learning Theory*, pp. 2858–2887, 2020.

Mhammedi, Z., Koolen, W. M., and Van Erven, T. Lipschitz adaptivity with multiple learning rates in online learning. In *Proceedings of the 32nd Conference on Learning Theory*, pp. 2490–2511, 2019.

Mohri, M. and Yang, S. Accelerating online convex optimization via adaptive prediction. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 848–856, 2016.

Mukkamala, M. C. and Hein, M. Variants of RMSProp and Adagrad with logarithmic regret bounds. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2545–2553, 2017.

Orabona, F. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems 27*, pp. 1116–1124, 2014.

Orabona, F. and Pál, D. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems 29*, pp. 577–585, 2016.

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

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

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

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018.

Shalev-Shwartz, S. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

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

Shalev-Shwartz, S., Singer, Y., and Srebro, N. Pegasos: primal estimated sub-gradient solver for SVM. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 807–814, 2007.

Srebro, N., Sridharan, K., and Tewari, A. Smoothness, low-noise and fast rates. In *Advances in Neural Information Processing Systems 23*, pp. 2199–2207, 2010.

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

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

van Erven, T., Koolen, W. M., and van der Hoeven, D. MetaGrad: Adaptation using multiple learning rates in online learning. *Journal of Machine Learning Research*, 22(161):1–61, 2021.

Wang, G., Lu, S., and Zhang, L. Adaptivity and optimality: A universal algorithm for online convex optimization. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, pp. 659–668, 2019.

Wang, G., Lu, S., Cheng, Q., Tu, W.-W., and Zhang, L. Sadam: A variant of Adam for strongly convex functions. In *International Conference on Learning Representations*, 2020a.

Wang, G., Lu, S., Hu, Y., and Zhang, L. Adapting to smoothness: A more universal algorithm for online convex optimization. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 6162–6169, 2020b.

Yang, T., Mahdavi, M., Jin, R., and Zhu, S. Regret bounded by gradual variation for online convex optimization. *Machine Learning*, 95:183–223, 2014.

Zeiler, M. D. Adadelta: An adaptive learning rate method. *ArXiv e-prints*, arXiv:1212.5701, 2012.

Zhang, L. Online learning in changing environments. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pp. 5178–5182, 2020. Early Career.

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

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

Zhang, L., Wang, G., Tu, W.-W., Jiang, W., and Zhou, Z.-H. Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. In *Advances in*

Neural Information Processing Systems 34, pp. 24968–24980, 2021.

Zhao, P., Zhang, Y.-J., Zhang, L., and Zhou, Z.-H. Dynamic regret of convex and smooth functions. In *Advances in Neural Information Processing Systems 33*, pp. 12510–12520, 2020.

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

A. Analysis

In this section, we present the analysis of all theorems.

A.1. Proof of Theorem 3.6

We first analyze the meta-regret of our strategy. According to the theoretical guarantee of Adapt-ML-Prod (Gaillard et al., 2014, Corollary 4), we have

$$\sum_{t=1}^{T} \ell_t - \sum_{t=1}^{T} \ell_t^i \leq \frac{\Gamma}{\sqrt{\ln |\mathcal{E}|}} \sqrt{1 + \sum_{t=1}^{T} (\ell_t - \ell_t^i)^2} + 2\Gamma$$

for all expert $E^i \in \mathcal{E}$, where Γ is given in (24). Combining with the definitions of ℓ_t^i and ℓ_t in (19) and (20), we arrive at

$$\begin{aligned} &\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle \\ \leq & 4\Gamma G D + \frac{\Gamma}{\sqrt{\ln |\mathcal{E}|}} \sqrt{4G^2D^2 + \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle^2} \\ \leq & 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}}\right) + \frac{\Gamma}{\sqrt{\ln |\mathcal{E}|}} \sqrt{\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle^2} \end{aligned} \tag{27}$$

where the last step follows from the basic inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$.

To utilize the property of strong convexity in (12), we proceed in the following way:

$$\begin{aligned} &\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle \\ \leq & 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}}\right) + \frac{\Gamma^2 G^2}{2\lambda \ln |\mathcal{E}|} + \frac{\lambda}{2G^2} \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle^2 \\ \leq & 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}}\right) + \frac{\Gamma^2 G^2}{2\lambda \ln |\mathcal{E}|} + \frac{\lambda}{2G^2} \sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t)\|^2 \|\mathbf{x}_t - \mathbf{x}_t^i\|^2 \\ \overset{(10)}{\leq} & 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}}\right) + \frac{\Gamma^2 G^2}{2\lambda \ln |\mathcal{E}|} + \frac{\lambda}{2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_t^i\|^2 \end{aligned} \tag{28}$$

where the first step follows from the basic inequality $2\sqrt{ab} \leq a + b$. According to Definition 3.3, the meta-regret in terms of $f_t(\cdot)$ is given by

$$\begin{aligned} \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^i) \overset{(12)}{\leq} & \sum_{t=1}^{T} \left(\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle - \frac{\lambda}{2} \|\mathbf{x}_t - \mathbf{x}_t^i\|^2 \right) \\ \overset{(28)}{\leq} & 2\Gamma G D \left(2 + \frac{1}{\sqrt{\ln |\mathcal{E}|}}\right) + \frac{\Gamma^2 G^2}{2\lambda \ln |\mathcal{E}|}. \end{aligned} \tag{29}$$

Next, we study the expert-regret. Let E^i be the expert $E(A, \widehat{\lambda})$ where $A \in \mathcal{A}_{str}$, $\widehat{\lambda} \in \mathcal{P}_{str}$, and $\widehat{\lambda} \leq \lambda \leq 2\widehat{\lambda}$. Since λ-strongly convex functions are also $\widehat{\lambda}$-strongly convex, expert $E(A, \widehat{\lambda})$ makes a right assumption, and the following inequality is true

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t^i) - \min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \leq R(A, \widehat{\lambda}). \tag{30}$$

Combining (29) and (30), we have

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \leq R(A, \widehat{\lambda}) + 2\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{E}|}}\right) + \frac{\Gamma^2 G^2}{2\lambda \ln|\mathcal{E}|}. \tag{31}$$

We complete the proof by noticing that (31) holds for any $A \in \mathcal{A}_{str}$.

A.2. Proof of Corollary 3.8

From the theoretical guarantee of OEGD for strongly convex and smooth functions in Theorem B.1, we have

$$R(\text{OEGD}, \widehat{\lambda}) = O\left(\frac{\log V_T}{\widehat{\lambda}}\right) \overset{\lambda \leq 2\widehat{\lambda}}{=} O\left(\frac{\log V_T}{\lambda}\right). \tag{32}$$

Similarly, from the regret bound of S^{2}OGD (Wang et al., 2020b, Theorem 1), we have

$$R(\text{S}^2\text{OGD}, \widehat{\lambda}) = O\left(\frac{\log L_T^*}{\widehat{\lambda}}\right) \overset{\lambda \leq 2\widehat{\lambda}}{=} O\left(\frac{\log L_T^*}{\lambda}\right). \tag{33}$$

We obtain the corollary by substituting (32) and (33) into Theorem 3.6.

A.3. Proof of Theorem 3.9

The analysis is similar to that of Theorem 3.6. To make use of the property of exponential concavity in (13), we change (28) as follows:

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle \\
\leq & 2\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{E}|}}\right) + \frac{\Gamma^2}{2\beta \ln|\mathcal{E}|} + \frac{\beta}{2}\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle^2.
\end{aligned} \tag{34}$$

According to Lemma 3.5, the meta-regret in terms of $f_t(\cdot)$ can be bounded by

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^i) \\
\overset{(13)}{\leq} & \sum_{t=1}^{T} \left(\langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle - \frac{\beta}{2} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle^2 \right) \\
\overset{(34)}{\leq} & 2\Gamma GD\left(2 + \frac{1}{\sqrt{\ln|\mathcal{E}|}}\right) + \frac{\Gamma^2}{2\beta \ln|\mathcal{E}|}.
\end{aligned}$$

The rest of the proof is identical to that of Theorem 3.6.

A.4. Proof of Corollary 3.10

From the theoretical guarantee of ONS for exp-concave and smooth functions (Orabona et al., 2012, Theorem 1), we have

$$R(\text{ONS}, \widehat{\alpha}) = O\left(\frac{d \log L_T^*}{\widehat{\alpha}}\right) \overset{\alpha \leq 2\widehat{\alpha}}{=} O\left(\frac{d \log L_T^*}{\alpha}\right). \tag{35}$$

Similarly, from the regret bound of OEGD (Chiang et al., 2012, Theorem 15), we have

$$R(\text{OEGD}, \widehat{\alpha}) = O\left(\frac{d \log V_T}{\widehat{\alpha}}\right) \overset{\alpha \leq 2\widehat{\alpha}}{=} O\left(\frac{d \log V_T}{\alpha}\right). \tag{36}$$

We obtain the corollary by substituting (35) and (36) into Theorem 3.9.

A.5. Proof of Theorem 3.11

From the first-order condition of convex functions, we bound the meta-regret by

$$\begin{aligned}\sum_{t=1}^T f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^i) \leq & \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle \\ \overset{(27)}{\leq} & 4\Gamma GD + \frac{\Gamma}{\sqrt{\ln|\mathcal{E}|}}\sqrt{4G^2D^2 + \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_t^i \rangle^2} \\ \leq & 4\Gamma GD + \frac{\Gamma}{\sqrt{\ln|\mathcal{E}|}}\sqrt{4G^2D^2 + \sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t)\|^2 \|\mathbf{x}_t - \mathbf{x}_t^i\|^2} \\ \overset{(11)}{\leq} & 4\Gamma GD + \frac{\Gamma D}{\sqrt{\ln|\mathcal{E}|}}\sqrt{4G^2 + \sum_{t=1}^T \|\nabla f_t(\mathbf{x}_t)\|^2}.\end{aligned}$$

We complete the proof by combining the above inequality with that of the expert-regret:

$$\sum_{t=1}^T f_t(\mathbf{x}_t^i) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x}) \leq R(A), \ \forall A \in \mathcal{A}_{con}.$$

A.6. Proof of Corollary 3.12

We need the self-bounding property of smooth functions (Srebro et al., 2010, Lemma 3.1).

Lemma A.1. *For a nonnegative and H-smooth function $f : \mathcal{X} \mapsto \mathbb{R}$, we have*

$$\|\nabla f(\mathbf{x})\| \leq \sqrt{4Hf(\mathbf{x})}, \ \forall \mathbf{x} \in \mathcal{X}. \tag{37}$$

Combining Lemma A.1 and Theorem 3.11, we have

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x}) \leq \min_{A\in\mathcal{A}_{con}} R(A) + 4\Gamma GD + \frac{\Gamma D}{\sqrt{\ln|\mathcal{E}|}}\sqrt{4G^2 + 4H\sum_{t=1}^T f_t(\mathbf{x}_t)}. \tag{38}$$

From the theoretical guarantee of SOGD for convex and smooth functions (Zhang et al., 2019, Theorem 2), we have

$$R(\text{SOGD}) = 8HD^2 + D\sqrt{2\delta + 8HL_*} \tag{39}$$

where $\delta > 0$ can be any small constant. Substituting (39) into (38), we obtain

$$\sum_{t=1}^T f_t(\mathbf{x}_t) - L_* \leq 8HD^2 + D\sqrt{2\delta + 8HL_*} + 4\Gamma GD + \frac{\Gamma D}{\sqrt{\ln|\mathcal{E}|}}\sqrt{4G^2 + 4H\sum_{t=1}^T f_t(\mathbf{x}_t)}. \tag{40}$$

To simplify the above inequality, we use the following lemma (Shalev-Shwartz, 2007, Lemma 19).

Lemma A.2. *Let $x, b, c \in \mathbb{R}_+$. Then,*

$$x - c \leq b\sqrt{x} \Rightarrow x - c \leq b^2 + b\sqrt{c}.$$

From (40), we have

$$\begin{aligned}&\left(\frac{G^2}{H} + \sum_{t=1}^T f_t(\mathbf{x}_t)\right) - \left(L_* + D\sqrt{2\delta + 8HL_*} + 4\Gamma GD + 8HD^2 + \frac{G^2}{H}\right) \\ \leq & \frac{\Gamma D\sqrt{4H}}{\sqrt{\ln|\mathcal{E}|}}\sqrt{\frac{G^2}{H} + \sum_{t=1}^T f_t(\mathbf{x}_t)},\end{aligned}$$

Lemma A.2 implies

$$
\begin{aligned}
&\left(\frac{G^2}{H}+\sum_{t=1}^T f_t(\mathbf{x}_t)\right)-\left(L_*+D\sqrt{2\delta+8HL_*}+4\Gamma GD+8HD^2+\frac{G^2}{H}\right)\\
\leq&\frac{4\Gamma^2D^2H}{\ln|\mathcal{E}|}+\frac{\Gamma D\sqrt{4H}}{\sqrt{\ln|\mathcal{E}|}}\sqrt{L_*+D\sqrt{2\delta+8HL_*}+4\Gamma GD+8HD^2+\frac{G^2}{H}}.
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&\sum_{t=1}^T f_t(\mathbf{x}_t)-\min_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^T f_t(\mathbf{x})=\sum_{t=1}^T f_t(\mathbf{x}_t)-L_*\\
\leq&\frac{\Gamma D\sqrt{4H}}{\sqrt{\ln|\mathcal{E}|}}\sqrt{L_*+D\sqrt{2\delta+8HL_*}+4\Gamma GD+8HD^2+\frac{G^2}{H}}+D\sqrt{2\delta+8HL_*}\\
&+4\Gamma GD+8HD^2+\frac{4\Gamma^2D^2H}{\ln|\mathcal{E}|}\\
=&O\left(\sqrt{L_*\log\log T}\right).
\end{aligned}
$$

B. Online extra-gradient descent (OEGD) for strongly convex and smooth functions

In this section, we extend the OEGD algorithm of Chiang et al. (2012) to strongly convex functions.

B.1. The algorithm

There are two sequences of solutions $\{\mathbf{x}_t\}_{t=1}^T$ and $\{\mathbf{u}_t\}_{t=1}^T$, where $\mathbf{u}_t$ is an auxiliary solution used to exploit the smoothness of the loss function.

Based on the property of strong convexity in (12), we set

$$
\mathcal{R}_t(\mathbf{x})=\frac{1}{2\eta_t}\|\mathbf{x}\|^2
$$

in Algorithm 1 of Chiang et al. (2012), where

$$
\eta_t=\frac{8G^2}{\lambda(\sum_{i=1}^{t-1}\|\nabla f_i(\mathbf{x}_i)-\nabla f_{i-1}(\mathbf{x}_{i-1})\|^2+G^2/\lambda+4G^2)}\overset{(10)}{\leq}\frac{8G^2}{\lambda(\sum_{i=1}^{t}\|\nabla f_i(\mathbf{x}_i)-\nabla f_{i-1}(\mathbf{x}_{i-1})\|^2+G^2/\lambda)}\leq 8, \tag{41}
$$

and obtain the following updating rules:

$$
\begin{aligned}
\mathbf{u}_{t+1}=&\Pi_{\mathcal{X}}\left[\mathbf{u}_t-\eta_t\nabla f_t(\mathbf{x}_t)\right],\\
\mathbf{x}_{t+1}=&\Pi_{\mathcal{X}}\left[\mathbf{u}_{t+1}-\eta_{t+1}\nabla f_t(\mathbf{x}_t)\right],
\end{aligned}
$$

where $\Pi_{\mathcal{X}}[\cdot]$ denotes the projection onto the nearest point in $\mathcal{X}$.

Theorem B.1. *Under Assumptions 3.1, 3.2, and 3.7, if the online functions are λ-strongly convex, we have*

$$
\sum_{t=1}^T f_t(\mathbf{x}_t)-\min_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^T f_t(\mathbf{x})\leq m\left(\frac{8G^2}{\lambda}+32G^2\right)\ln\left(\frac{2\lambda}{G^2}V_T+2\right)+\frac{D^2(8\lambda+3)}{32}=O\left(\frac{\log V_T}{\lambda}\right)
$$

where V_T is defined in (4), and

$$
m=\frac{\log(512H^2(1+4\lambda)+1)}{\log(\frac{2\lambda}{G^2}V_T+2)}+1=O(1). \tag{42}
$$

B.2. Proof of Theorem B.1

Based on Definition 3.3 and Lemma 5 of Chiang et al. (2012), we have

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_*) \overset{(12)}{\leq} \sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}_* \rangle - \frac{\lambda}{2} \sum_{t=1}^{T} \|\mathbf{x}_* - \mathbf{x}_t\|^2 \\
\leq & \underbrace{\sum_{t=1}^{T} \langle \nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1}), \mathbf{x}_t - \mathbf{u}_{t+1} \rangle}_{:=a} + \underbrace{\sum_{t=1}^{T} \frac{\|\mathbf{x}_* - \mathbf{u}_t\|^2 - \|\mathbf{x}_* - \mathbf{u}_{t+1}\|^2}{2\eta_t} - \frac{\lambda}{2}\|\mathbf{x}_t - \mathbf{x}_*\|^2}_{:=b} \\
& - \underbrace{\sum_{t=1}^{T} \frac{\|\mathbf{x}_t - \mathbf{u}_t\|^2 + \|\mathbf{x}_t - \mathbf{u}_{t+1}\|^2}{2\eta_t}}_{:=c}
\end{aligned} \tag{43}$$

where $\mathbf{x}_*$ is the best decision in hindsight. In the following, we upper bound the three terms above respectively. For term a, we have

$$\begin{aligned}
a \leq & \sum_{t=1}^{T} \eta_t \|\nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1})\|^2 \\
\overset{(41)}{\leq} & \sum_{t=1}^{T} \frac{8G^2}{\lambda(\sum_{i=1}^{t} \|\nabla f_i(\mathbf{x}_i) - \nabla f_{i-1}(\mathbf{x}_{i-1})\|^2 + G^2/\lambda)} \|\nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1})\|^2 \\
\leq & \frac{8G^2}{\lambda} \ln\left(\frac{\lambda}{G^2} \sum_{t=1}^{T} \|\nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1})\|^2 + 1 \right) \\
\leq & \frac{8G^2}{\lambda} \ln\left(\frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1 \right)
\end{aligned} \tag{44}$$

where the 1st inequality is derived from Lemma 6 of Chiang et al. (2012), the 3rd inequality follows from Lemma 11 of Hazan et al. (2007) for one dimension, and the 4th inequality is due to Lemma 12 of Chiang et al. (2012).

Similar to the proof of Lemma 14 of Chiang et al. (2012), we bound term b as

$$\begin{aligned}
b \overset{(11)}{\leq} & \frac{D^2}{2\eta_1} + \frac{1}{2} \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) \|\mathbf{x}_* - \mathbf{u}_t\|^2 - \frac{\lambda}{2} \sum_{t=1}^{T} \|\mathbf{x}_* - \mathbf{x}_t\|^2 \\
\overset{(10),(41)}{\leq} & \frac{D^2(4\lambda+1)}{16} + \frac{\lambda}{4} \sum_{t=1}^{T-1} \|\mathbf{x}_* - \mathbf{u}_{t+1}\|^2 - \frac{\lambda}{2} \sum_{t=1}^{T} \|\mathbf{x}_* - \mathbf{x}_t\|^2 \\
\leq & \frac{D^2(4\lambda+1)}{16} + \frac{\lambda}{2} \sum_{t=1}^{T} \|\mathbf{u}_{t+1} - \mathbf{x}_t\|^2 \\
\leq & \frac{D^2(4\lambda+1)}{16} + \frac{\lambda}{2} \sum_{t=1}^{T} \eta_t^2 \|\nabla f_t(\mathbf{x}_t) - \nabla f_{t-1}(\mathbf{x}_{t-1})\|^2 \\
\overset{(44)}{\leq} & \frac{D^2(4\lambda+1)}{16} + \frac{\lambda \eta_1}{2} \frac{8G^2}{\lambda} \ln\left(\frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1 \right) \\
\overset{(41)}{\leq} & \frac{D^2(4\lambda+1)}{16} + 32G^2 \ln\left(\frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1 \right)
\end{aligned} \tag{45}$$

where the 3rd inequality is based on Proposition 1 of Chiang et al. (2012), the 4th inequality is derived from Proposition 7 of Chiang et al. (2012), and the 5th inequality is based on the fact that η_t is non-increasing and (44).

For term c, based on the proof of Lemma 21 of Chiang et al. (2012), we have

$$\begin{aligned} c = \sum_{t=1}^{T} \frac{\|\mathbf{x}_t - \mathbf{u}_t\|^2}{2\eta_t} + \sum_{t=2}^{T+1} \frac{\|\mathbf{x}_{t-1} - \mathbf{u}_t\|^2}{2\eta_{t-1}} \geq & \sum_{t=2}^{T} \frac{\|\mathbf{x}_t - \mathbf{u}_t\|^2}{2\eta_{t-1}} + \sum_{t=2}^{T} \frac{\|\mathbf{x}_{t-1} - \mathbf{u}_t\|^2}{2\eta_{t-1}} \\ \geq & \sum_{t=2}^{T} \frac{\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2}{4\eta_{t-1}} \overset{(41)}{\geq} \frac{1}{32} \sum_{t=2}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2. \end{aligned} \tag{46}$$

Substituting (44), (45) and (46) into (43), we get

$$\begin{aligned} & \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_*) \\ \leq & \left(\frac{8G^2}{\lambda} + 32G^2\right) \ln\left(\frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1\right) - \frac{1}{32} \sum_{t=2}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + \frac{D^2(4\lambda + 1)}{16} \\ \leq & \left(\frac{8G^2}{\lambda} + 32G^2\right) \ln\left(\frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1\right) - \frac{1}{32} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + \frac{D^2(8\lambda + 3)}{32}. \end{aligned}$$

To simplify the above inequality, we make use of the following fact

$$\ln a \leq \frac{a}{b} + \ln b - 1, \ \forall a > 0, b > 0.$$

By setting

$$a = \frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1 \text{ and } b = \left(\frac{2\lambda}{G^2} V_T + 2\right)^m,$$

where $m > 0$ is a factor depending on V_T, we have

$$\begin{aligned} & \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_*) \\ \leq & \left(\frac{8G^2}{\lambda} + 32G^2\right) \left(\frac{\frac{2\lambda}{G^2} V_T + \frac{2\lambda H^2}{G^2} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + 1}{(\frac{2\lambda}{G^2} V_T + 2)^m} + m \ln\left(\frac{2\lambda}{G^2} V_T + 2\right) - 1\right) \\ & - \frac{1}{32} \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + \frac{D^2(8\lambda + 3)}{32} \\ \leq & \left[\frac{16H^2(1 + 4\lambda)}{(\frac{2\lambda}{G^2} V_T + 2)^m} - \frac{1}{32}\right] \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + \left(\frac{1}{(\frac{2\lambda}{G^2} V_T + 2)^{m-1}} - 1\right) \left(\frac{8G^2}{\lambda} + 32G^2\right) \\ & + \left(\frac{8G^2}{\lambda} + 32G^2\right) m \ln\left(\frac{2\lambda}{G^2} V_T + 2\right) + \frac{D^2(8\lambda + 3)}{32}. \end{aligned} \tag{47}$$

From our choice of m in (42), we have

$$m \geq \frac{\log(512H^2(1 + 4\lambda))}{\log(\frac{2\lambda}{G^2} V_T + 2)} \Rightarrow \frac{16H^2(1 + 4\lambda)}{(\frac{2\lambda}{G^2} V_T + 2)^m} - \frac{1}{32} \leq 0, \tag{48}$$

$$m \geq 1 \Rightarrow \frac{1}{(\frac{2\lambda}{G^2} V_T + 2)^{m-1}} - 1 \leq 0. \tag{49}$$

We complete the proof by combining (47), (48), and (49).

C. Experiments

In this part, we provide experimental details of Section 4.

Settings We consider the problem of online linear classification. In each round t, the learner chooses a linear classifier $\mathbf{x}_t \in \mathcal{X}$. After submitting the decision, the learner observes a small set of m examples $\{\mathbf{w}_t^{(i)}, y_t^{(i)}\}_{i=1}^m$, where $\mathbf{w}_t^{(i)} \in \mathbb{R}^d$ is the feature vector of the i-th example, and $y_t^{(i)} \in \{-1, +1\}$ is the corresponding label. Finally, the learner suffers a loss $f_t(\mathbf{x}_t)$ and updates the classifier. To demonstrate the universality of USC, we consider two different loss functions: the ℓ_2-regularized hinge-loss, i.e.,

$$f_t(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \max\left\{0, 1 - y_t^{(i)}\mathbf{x}^\top \mathbf{w}_t^{(i)}\right\} + \frac{\lambda}{2}\|\mathbf{x}\|^2$$

which is λ-strongly convex, and the standard hinge-loss, i.e.,

$$f_t(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \max\left\{0, 1 - y_t^{(i)}\mathbf{x}^\top \mathbf{w}_t^{(i)}\right\}$$

which is convex. We conduct both experiments on the a9a dataset (Chang & Lin, 2011), which contains 32561 examples and $d = 123$ features. During the learning process, we randomly sample $m = 10$ data points in each iteration. We configure $\lambda = 0.02$, the diameter of the decision set $D = 20$, and the time horizon $T = 10000$. We also estimate the value of G based on D and the gradient of the functions.

Algorithms We compare the performance of our proposed USC with existing universal methods MetaGrad (van Erven & Koolen, 2016) and Maler (Wang et al., 2019). The candidate algorithms for USC are constructed as follows.

- The algorithm set for convex functions $\mathcal{A}_{con}$ includes OGD with step size $\eta_t = \frac{G}{D\sqrt{t}}$ (Zinkevich, 2003), and Adam with hype-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and different step sizes in range $\{1, 10^{-1}, \ldots, 10^{-4}\}$ (Kingma & Ba, 2015).
- The algorithm set for exp-concave functions $\mathcal{A}_{exp}$ contains the ONS algorithm (Hazan et al., 2007). The set of possible parameters $\mathcal{P}_{exp}$ is constructed following (26).
- The algorithm set for strongly convex functions $\mathcal{A}_{str}$ consists of SC-OGD with step size $\eta_t = \frac{1}{\lambda t}$ (Shalev-Shwartz et al., 2007), and SAdam with hype-parameters $\beta_1 = 0.9$, $\beta_2 = 1 - \frac{0.9}{t}$, and step size $\alpha = \frac{0.1}{\lambda}$ (Wang et al., 2020a). The parameter set $\mathcal{P}_{str}$ is constructed as in (23).

Results All the experiments are repeated 5 times and the averaged results are recorded. We present the regret v.s. the number of iterations for optimizing strongly convex functions and convex functions in Fig. 1(a) and Fig. 1(b), respectively. Apart from universal methods, we also report the best performance of each candidate algorithm in USC. As can be seen, SC-OGD has the smallest regret for the strongly convex problem, while SAdam achieves the best performance for the convex problem. In both experiments, the performance of USC nearly matches that of the best expert, and is better than other universal methods such as MetaGrad and Maler.