Revisiting Smoothed Online Learning

Lijun Zhang[1,2], **Wei Jiang**[1], **Shiyin Lu**[1], **Tianbao Yang**[3]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Peng Cheng Laboratory, Shenzhen, Guangdong, China
[3]Department of Computer Science, The University of Iowa, Iowa City, IA 52242, USA
{zhanglj, jiangw, lusy}@lamda.nju.edu.cn, tianbao-yang@uiowa.edu

Abstract

In this paper, we revisit the problem of smoothed online learning, in which the online learner suffers both a hitting cost and a switching cost, and target two performance metrics: competitive ratio and dynamic regret with switching cost. To bound the competitive ratio, we assume the hitting cost is known to the learner in each round, and investigate the simple idea of balancing the two costs by an optimization problem. Surprisingly, we find that minimizing the hitting cost alone is $\max(1, \frac{2}{\alpha})$-competitive for α-polyhedral functions and $1 + \frac{4}{\lambda}$-competitive for λ-quadratic growth functions, both of which improve state-of-the-art results significantly. Moreover, when the hitting cost is both convex and λ-quadratic growth, we reduce the competitive ratio to $1 + \frac{2}{\sqrt{\lambda}}$ by minimizing the weighted sum of the hitting cost and the switching cost. To bound the dynamic regret with switching cost, we follow the standard setting of online convex optimization, in which the hitting cost is convex but hidden from the learner before making predictions. We modify Ader, an existing algorithm designed for dynamic regret, slightly to take into account the switching cost when measuring the performance. The proposed algorithm, named as Smoothed Ader, attains an optimal $O(\sqrt{T(1+P_T)})$ bound for dynamic regret with switching cost, where P_T is the path-length of the comparator sequence. Furthermore, if the hitting cost is accessible in the beginning of each round, we obtain a similar guarantee without the bounded gradient condition, and establish an $(\sqrt{T(1+P_T)})$ lower bound to confirm the optimality.

1 Introduction

Online learning is the process of making a sequence of predictions given knowledge of the answer to previous tasks and possibly additional information [Shalev-Shwartz, 2011]. While the traditional online learning aims to make the prediction as accurate as possible, in this paper, we study smoothed online learning (SOL), where the online learner incurs a switching cost for changing its predictions between rounds [Cesa-Bianchi et al., 2013]. SOL has received lots of attention recently because in many real-world applications, a change of action usually brings some additional cost. Examples include the dynamic right-sizing for data centers [Lin et al., 2011], geographical load balancing [Lin et al., 2012], real-time electricity pricing [Kim and Giannakis, 2014], video streaming [Joseph and de Veciana, 2012], spatiotemporal sequence prediction [Kim et al., 2015], multi-timescale control [Goel et al., 2017], and thermal management [Zanini et al., 2010].

Specifically, SOL is performed in a sequence of consecutive rounds, where at round t the learner is asked to select a point $\mathbf{x}_t$ from the decision set $\mathcal{X}$, and suffers a hitting cost $f_t(\mathbf{x}_t)$. Depending on the performance metric, the learner *may* be allowed to observe $f_t(\cdot)$ when making decisions, which is different from the traditional online learning in which $f_t(\cdot)$ is revealed to the learner after submitting the decision [Cesa-Bianchi and Lugosi, 2006]. Additionally, the learner also incurs a switching cost $m(\mathbf{x}_t, \mathbf{x}_{t-1})$ for changing decisions between successive rounds. The switching cost $m(\mathbf{x}_t, \mathbf{x}_{t-1})$

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

could be any distance function, such as the ℓ_2-norm distance $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$ and the squared ℓ_2-norm distance $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2/2$ [Goel et al., 2019]. In the literature, there are two performance metrics for SOL: competitive ratio and dynamic regret with switching cost.

Competitive ratio is popular in the community of online algorithms [Borodin and El-Yaniv, 1998]. It is defined as the worst-case ratio of the total cost incurred by the online learner and the offline optimal cost:

$$\frac{\sum_{t=1}^{T} \left(f_t(\mathbf{x}_t) + m(\mathbf{x}_t, \mathbf{x}_{t-1}) \right)}{\min_{\mathbf{u}_0,\mathbf{u}_1,\ldots,\mathbf{u}_T \in \mathcal{X}} \sum_{t=1}^{T} \left(f_t(\mathbf{u}_t) + m(\mathbf{u}_t, \mathbf{u}_{t-1}) \right)}. \tag{1}$$

When focusing on the competitive ratio, the learner can observe $f_t(\cdot)$ before picking $\mathbf{x}_t$. The problem is still nontrivial due to the coupling created by the switching cost. On the other hand, dynamic regret with switching cost is a generalization of dynamic regret—a popular performance metric in the community of online learning [Zinkevich, 2003]. It is defined as the difference between the total cost incurred by the online learner and that of an arbitrary comparator sequence $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$:

$$\sum_{t=1}^{T} \left(f_t(\mathbf{x}_t) + m(\mathbf{x}_t, \mathbf{x}_{t-1}) \right) - \sum_{t=1}^{T} \left(f_t(\mathbf{u}_t) + m(\mathbf{u}_t, \mathbf{u}_{t-1}) \right). \tag{2}$$

Different from previous work [Chen et al., 2018, Goel et al., 2019], we did not introduce the minimization operation over $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T$ in (2). The reason is that we want to bound (2) by certain regularities of the comparator sequence, such as the path-lengtht

Our analysis of the naive approach and the greedy algorithm is very simple. In contrast, both OBD and Greedy OBD rely on intricate geometric arguments.

While both OBD and R-OBD are equipped with sublinear dynamic regret with switching cost, they are unsatisfactory in the following aspects:

The regret of OBD depends on an upper bound of the path-length instead of the path-length itself [Chen et al., 2018, Corollary 11], making it nonadaptive.

The regret of R-OBD is adaptive but it uses the squared ℓ_2-norm to measure the switching an upper bound of the path-length of the comparator sequence, i.e., $P_T(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T) \leq L$. Chen et al. [2018, Corollary 11] have proved that

$$\sum_{t=1}^{T} \Big(f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \Big) - \min_{P_T(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T) \leq L} \sum_{t=1}^{T} \Big(f_t(\mathbf{u}_t) + \|\mathbf{u}_t - \mathbf{u}_{t-1}\| \Big) = O\left(\sqrt{TL}\right) \tag{5}$$

leading to sublinear regret when $L = o(T)$. However, the upper bound is nonadaptive because it depends on L instead of the actual path-length P_T.

Later, Goel and Wierman [2019] demonstrate that OBD is $3 + O(\frac{1}{\lambda})$-competitive for λ-strongly convex functions, when the switching cost is set to be the squared ℓ_2-norm distance. In a subsequent work, Goel et al. [2019, Theorem 4] propose Regularized OBD (R-OBD), which improves the competitive ratio to $\frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4}{\lambda}}$, matching the lower bound of strongly convex functions exactly [Goel et al., 2019, Theorem 1]. R-OBD includes (4) as a special case, which also enjoys the optimal competitive ratio for strongly convex functions. Furthermore, Goel et al. [2019, Theorem 6] have analyzed the dynamic regret of R-OBD and the following result can be extracted from that paper

$$\sum_{t=1}^{T} \Big(f_t(\mathbf{x}_t) + \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 \Big) - \sum_{t=1}^{T} \Big(f_t(\mathbf{u}_t) + \frac{1}{2}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|^2 \Big) = O\left(\sqrt{T \sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|^2}\right)$$

Compared with (5), this bound is adaptive because the upper bound depends on the switching cost of the comparator sequence. However, it chooses the squared ℓ_2-norm as the switching cost, which may not be suitable for general convex functions. When the hitting cost is both quasiconvex and λ-quadratic growth, Goel et al. [2019, Theorem 3] have demonstrated that their Greedy OBD algorithm attains an $O(1/\sqrt{\lambda})$ competitive ratio, as $\lambda \to 0$.

Lin et al. [2020] have analyzed the naive approach which ignores the switching cost and simply minimizes the hitting cost in each round, i.e., the greedy algorithm with $\gamma = 0$. It is a bit surprising that this naive approach is $1 + \frac{2}{\alpha}$-competitive for α-polyhedral functions and $\max(1 + \frac{6}{\lambda}, 4)$-competitive for λ-quadratic growth functions, without any convexity assumption [Lin et al., 2020, Lemma 1]. Argue et al. [2020a] have investigated a hybrid setting in which the hitting cost is both λ-strongly convex and H-smooth, but the switching cost is the ℓ_2-norm distance instead of the squared one. They develop Constrained OBD, and establish a $4 + 4\sqrt{2H/\lambda}$ competitive ratio. However, their analysis relies on a strong condition that the hitting cost is non-negative over the whole space, i.e., $\min_{\mathbf{x} \in \mathbb{R}^d} f_t(\mathbf{x}) = 0$, as opposed to the usual condition $\min_{\mathbf{x} \in \mathcal{X}} f_t(\mathbf{x}) = 0$.

Finally, we note that SOCO is closely related to convex body chasing (CBC) [Friedman and Linial, 1993, Antoniadis et al., 2016, Bansal et al., 2018, Argue et al., 2019, Bubeck et al., 2019, 2020]. In this problem, the online learner receives a sequence of convex bodies $\mathcal{X}_1, \ldots, \mathcal{X}_T \subseteq \mathbb{R}^d$ and must select one point from each body, and the goal is to minimize the total movement between consecutive output points. Apparently, we can treat CBC as a special case of SOCO by defining the hitting cost $f_t(\cdot)$ as the indicator function of $\mathcal{X}_t$, which means that the domains of hitting costs are allowed to be different. On the other hand, we can also formulate a d-dimensional SOCO problem as a $d+1$-dimensional CBC problem [Lin et al., 2020, Proposition 1]. For the general setting of CBC, the competitive ratio exhibits a polynomial dependence on the dimensionality, and the state-of-the-art result is $O(\min(d, \sqrt{d \log T}))$ [Argue et al., 2020b, Sellke, 2020], which nearly match the $\Omega(\sqrt{d})$ lower bound [Friedman and Linial, 1993]. Our paper aims to derive dimensionality-independent competitive ratios and sublinear dynamic regret for SOCO, under appropriate conditions.

2.2 Dynamic regret

Recently, dynamic regret has attained considerable interest in the community of online learning [Zhang, 2020]. The motivation of dynamic regret is to deal with changing environments, in which the optimal decision may change over time. It is defined as the difference between the cumulative loss of the learner and that of a sequence of comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$:

$$\text{D-Regret}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t). \tag{6}$$

In the general form of dynamic regret, $\mathbf{u}_1, \ldots, \mathbf{u}_T$ could be an *arbitrary* sequence [Zinkevich, 2003, Hall and Willett, 2013, Zhang et al., 2018a, 2020, Cutkosky, 2020, Zhao et al., 2020a], and in the restricted form, they are chosen as the minimizers of online functions, i.e., $\mathbf{u}_t \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} f_t(\mathbf{x})$ [Jadbabaie et al., 2015, Besbes et al., 2015, Yang et al., 2016, Mokhtari et al., 2016, Zhang et al., 2017, 2018b, Wan et al., 2021, Zhao and Zhang, 2021]. While it is well-known that sublinear dynamic regret is unattainable in the worst case, one can bound the dynamic regret in terms of some regularities of the comparator sequence. An instance is given by Zinkevich [2003], who introduces the notion of path-length defined in (3) to measure the temporal variability of the comparator sequence, and derives an $O(\sqrt{T}(1 + P_T))$ bound for the dynamic regret of OGD. Later, Zhang et al. [2018a] develop adaptive learning for dynamic environment (Ader), which achieves the optimal $O(\sqrt{T(1 + P_T)})$ dynamic regret. In this paper, we show that a small change of Ader attains the same bound for dynamic regret with switching cost.

3 Competitive ratio

In this section, we focus on competitive ratio. Without loss of generality, we assume the hitting cost is non-negative, since the competitive ratio can only improve if this is not the case.

3.1 Polyhedral functions

We first introduce the definition of polyhedral functions.

Definition 1 *A function $f(\cdot) : \mathcal{X} \mapsto \mathbb{R}$ with minimizer $\mathbf{v}$ is α-polyhedral if*

$$f(\mathbf{x}) - f(\mathbf{v}) \geq \alpha \|\mathbf{x} - \mathbf{v}\|, \ \forall \mathbf{x} \in \mathcal{X}. \tag{7}$$

We note that polyhedral functions have been used for stochastic network optimization [Huang and Neely, 2011] and geographical load balancing [Lin et al., 2012].

Following Chen et al. [2018], we set the switching cost as $m(\mathbf{x}_t, \mathbf{x}_{t-1}) = \|\mathbf{x}_t - \mathbf{x}_{t-1}\|$. Intuitively, we may expect that the switching cost should be taken into consideration when making decisions. However, our analysis shows that minimizing the hitting cost alone yields the tightest competitive ratio so far. Specifically, we consider the following naive approach that ignores the switching cost and selects

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}} f_t(\mathbf{x}). \tag{8}$$

)) : $\mathcal{X} \mapsto \mathbb{R}$ *with minimizer* $\mathbf{v}_t$ k

Definition 2 *A function $f(\cdot) : \mathcal{X} \mapsto \mathbb{R}$ with minimizer $\mathbf{v}$ is λ-quadratic growth if*

$$f(\mathbf{x}) - f(\mathbf{v}) \geq \frac{\lambda}{2}\|\mathbf{x} - \mathbf{v}\|^2, \ \forall \mathbf{x} \in \mathcal{X}. \tag{9}$$

The quadratic growth condition has been exploited by the optimization community [Drusvyatskiy and Lewis, 2018, Necoara et al., 2019] to establish linear convergence, and this condition is weaker than strong convexity [Hazan and Kale, 2011].

Following Goel et al. [2019], we set the switching cost as $m(\mathbf{x}_t, \mathbf{x}_{t-1}) = \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2/2$. We also consider the naive approach in (8) and have the following theoretical guarantee.

Theorem 2 *Suppose each $f_t(\cdot) : \mathcal{X} \mapsto \mathbb{R}$ with minimizer $\mathbf{v}_t$ is λ-quadratic growth. We have*

$$\sum_{t=1}^{T}\left(f_t(\mathbf{x}_t) + \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2\right) \leq \left(1 + \frac{4}{\lambda}\right)\sum_{t=1}^{T}\left(f_t(\mathbf{u}_t) + \frac{1}{2}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|^2\right),$$

for all $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$, where we assume $\mathbf{x}_0 = \mathbf{u}_0$.

Remark: The above theorem implies that the naive approach achieves a competitive ratio of $1 + \frac{4}{\lambda}$, which matches the lower bound of this algorithm [Goel et al., 2019, Theorem 5]. Furthermore, it is also much better than the $\max(1 + \frac{6}{\lambda}, 4)$ ratio established by Lin et al. [2020] for (8). Similar to the case of polyhedral functions, it seems safe to ignore the switching cost here.

3.3 Convex and quadratic growth functions

When $f_t(\cdot)$ is both quasiconvex and λ-quadratic growth, Goel et al. [2019] have established an $O(1/\sqrt{\lambda})$ competitive ratio for Greedy OBD. Inspired by this result, we introduce convexity to further improve the competitive ratio. In this case, the switching cost plays a role in deriving tighter competitive ratios. Specifically, we choose the greedy algorithm with $\gamma > 0$ to select $\mathbf{x}_t$, i.e.,

$$\mathbf{x}_t = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}}\left(f_t(\mathbf{x}) + \frac{\gamma}{2}\|\mathbf{x} - \mathbf{x}_{t-1}\|^2\right). \tag{10}$$

The theoretical guarantee of (10) is stated below.

Theorem 3 *Suppose the domain $\mathcal{X}$ is convex, and each $f_t(\cdot) : \mathcal{X} \mapsto \mathbb{R}$ with minimizer $\mathbf{v}_t$ is λ-quadratic growth and convex. By setting $\gamma = \lambda/(\lambda + \sqrt{\lambda})$, we have*

$$\sum_{t=1}^{T}\left(f_t(\mathbf{x}_t) + \frac{1}{2}\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2\right) \leq \left(1 + \frac{2}{\sqrt{\lambda}}\right)\sum_{t=1}^{T}\left(f_t(\mathbf{u}_t) + \frac{1}{2}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|^2\right),$$

for all $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$, where we assume $\mathbf{x}_0 = \mathbf{u}_0$.

Remark: The above theorem shows that the competitive ratio is improved to $1 + \frac{2}{\sqrt{\lambda}}$ under the additional convexity condition. According to the lower bound of strongly convex functions [Goel et al., 2019, Theorem 1], the ratio in Theorem 3 is optimal up to constant factors. Compared with Greedy OBD [Goel et al., 2019, Theorem 3], our assumption is slightly stronger, since we require convexity instead of quasiconvexity. However, our algorithm and analysis are much simpler, and the constants in our bound are much smaller.

4 Dynamic regret with switching cost

When considering dynamic regret with switching cost, we adopt the common assumptions of online convex optimization (OCO) [Shalev-Shwartz, 2011].

Assumption 1 *All the functions f_t's are convex over their domain $\mathcal{X}$.*

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

$$\max_{\mathbf{x} \in \mathcal{X}}\|\nabla f_t(\mathbf{x})\| \leq G, \ \forall t \in [T]. \tag{11}$$

Algorithm 1 SAder: Meta-algorithm

Require: A step size β, and a set $\mathcal{H}$ containing step sizes for experts

1: Activate a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking the expert-algorithm for each step size $\eta \in \mathcal{H}$
2: Sort step sizes in ascending order $\eta_1 \leq \eta_2 \leq \cdots \leq \eta_N$, and set $w_1^{\eta_i} = \frac{C}{i(i+1)}$
3: **for** $t = 1, \ldots, T$ **do**
4: Receive $\mathbf{x}_t^\eta$ from each expert E^η
5: Output the weighted average $\mathbf{x}_t = \sum_{\eta \in \mathcal{H}} w_t^\eta \mathbf{x}_t^\eta$
6: Observe the loss function $f_t(\cdot)$
7: Update the weight of each expert by (14)
8: Send gradient $\nabla f_t(\mathbf{x}_t)$ to each expert E^η
9: **end for**

Assumption 3 *The diameter of the domain $\mathcal{X}$ is bounded by D, i.e.,*

$$\max_{\mathbf{x}, \mathbf{x}' \in \mathcal{X}} \|\mathbf{x} - \mathbf{x}'\| \leq D. \tag{12}$$

Assumption 2 implies that the hitting cost is Lipschitz continuous, so it is natural to set the switching cost as $m(\mathbf{x}_t, \mathbf{x}_{t-1}) = \|\mathbf{x}_t - \mathbf{x}_{t-1}\|$.

4.1 The standard setting

We first follow the standard setting of OCO in which the learner can not observe the hitting cost when making predictions, and develop an algorithm based on Ader [Zhang et al., 2018a]. Specifically, we demonstrate that a small change of Ader, which modifies the loss of the meta-algorithm to take into account the switching cost of experts, is sufficient to minimize the dynamic regret with switching cost. Our proposed method is named as Smoothed Ader (SAder), and stated below.[2]

Meta-algorithm The meta-algorithm is similar to that of Ader [Zhang et al., 2018a, Algorithm 3], and summarized in Algorithm 1. The inputs of the meta-algorithm are its own step size β, and a set $\mathcal{H}$ of step sizes for experts. In Step 1, we active a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking the expert-algorithm for each $\eta \in \mathcal{H}$. In Step 2, we set the initial weight of each expert. Let η_i be the i-th smallest step size in $\mathcal{H}$. The weight of E^{η_i} is chosen as

$$w_1^{\eta_i} = \frac{C}{i(i+1)}, \text{ and } C = 1 + \frac{1}{|\mathcal{H}|}. \tag{13}$$

In each round, the meta-algorithm receives a set of predictions $\{\mathbf{x}_t^\eta | \eta \in \mathcal{H}\}$ from all experts (Step 4), and outputs the weighted average (Step 5):

$$\mathbf{x}_t = \sum_{\eta \in \mathcal{H}} w_t^\eta \mathbf{x}_t^\eta$$

where w_t^η is the weight assigned to expert E^η. After observing the loss function, the weights of experts are updated according to the exponential weighting scheme (Step 7) [Cesa-Bianchi and Lugosi, 2006]:

$$w_{t+1}^\eta = \frac{w_t^\eta e^{-\beta \ell_t(\mathbf{x}_t^\eta)}}{\sum_{\eta \in \mathcal{H}} w_t^\eta e^{-\beta \ell_t(\mathbf{x}_t^\eta)}} \tag{14}$$

where

$$\ell_t(\mathbf{x}_t^\eta) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t^\eta - \mathbf{x}_t \rangle + \|\mathbf{x}_t^\eta - \mathbf{x}_{t-1}^\eta\|. \tag{15}$$

When $t = 1$, we set $\mathbf{x}_0^\eta = \mathbf{0}$, for all $\eta \in \mathcal{H}$. As can be seen from (15) the switching cost $\|\mathbf{x}_t^\eta - \mathbf{x}_{t-1}^\eta\|$ of expert E^η to measure its performance. This is the *only* modification made to Ader. In the last step, we send the gradient $\nabla f_t(\mathbf{x}_t)$ to each expert E^η so that they can update their own predictions.

[2]In a concurrent work, Zhao et al. [2021] independently develop a similar algorithm for OCO with memory.

Algorithm 2 SAder: Expert-algorithm

Require: The step size η

1: Let $\mathbf{x}_1^\eta$ be any point in $\mathcal{X}$

2: **for** $t = 1, \ldots, T$ **do**

3: Submit $\mathbf{x}_t^\eta$ to the meta-algorithm

4: Receive gradient $\nabla f_t(\mathbf{x}_t)$ from the meta-algorithm

5:
$$\mathbf{x}_{t+1}^\eta = \Pi_{\mathcal{X}}\left[\mathbf{x}_t^\eta - \eta \nabla f_t(\mathbf{x}_t)\right]$$

6: **end for**

Expert-algorithm The expert-algorithm is the same as that of Ader [Zhang et al., 2018a, Algorithm 4], which is OGD over the linearized loss or the surrogate loss

$$s_t(\mathbf{x}) = \langle \nabla f_t(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle. \tag{16}$$

For the sake of completeness, we present its procedure in Algorithm 2. The input of the expert is its step size η. In Step 3 of Algorithm 2, each expert submits its prediction $\mathbf{x}_t^\eta$ to the meta-algorithm, and receives the gradient $\nabla f_t(\mathbf{x}_t)$ in Step 4. Then, in Step 5, it performs gradient descent

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

to get the prediction for the next round. Here, $\Pi_{\mathcal{X}}[\cdot]$ denotes the projection onto the nearest point in $\mathcal{X}$.

We have the following theoretical guarantee.

Theorem 4 *Set*

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\sqrt{\frac{D^2}{T(G^2+2G)}} \;\middle|\; i = 1, \ldots, N \right\} \tag{17}$$

where

$$N = \left\lceil \frac{1}{2}\log_2(1+2T) \right\rceil + 1, \textit{ and } \beta = \frac{2}{(2G+1)D}\sqrt{\frac{2}{5T}}$$

in Algorithm 1. Under Assumptions 1, 2 and 3, for any *comparator sequence* $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$, *SAder satisfies*

$$\begin{aligned}
&\sum_{t=1}^T \left(f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \right) - \sum_{t=1}^T f_t(\mathbf{u}_t) \\
\leq& \frac{3}{2}\sqrt{T(G^2+2G)\left(D^2 + 2D\sum_{t=1}^T \|\mathbf{u}_t - \mathbf{u}_{t-1}\|\right)} + (2G+1)D\sqrt{\frac{5T}{8}}\left[1 + 2\ln(k+1)\right] \\
=&O\left(\sqrt{T(1+P_T)} + \sqrt{T}(1+\log\log P_T)\right) = O\left(\sqrt{T(1+P_T)}\right)
\end{aligned} \tag{18}$$

where we define $\mathbf{x}_0 = \mathbf{0}$, *and*

$$k = \left\lfloor \frac{1}{2}\log_2\left(1 + \frac{2P_T}{D}\right) \right\rfloor + 1. \tag{19}$$

Remark: Theorem 4 shows that SAder attains an $O(\sqrt{T(1+P_T)})$ bound for dynamic regret with switching cost, which is on the same order as that of Ader for dynamic regret. From the $\Omega(\sqrt{T(1+P_T)})$ lower bound of dynamic regret [Zhang et al., 2018a, Theorem 2], we know that our upper bound is optimal up to constant factors. Compared with the regret bound of OBD in (5) [Chen et al., 2018], the advantage of SAder is that its regret depends on the path-length P_T directly, and thus becomes tighter when focusing on comparator sequences with smaller path-lengths. Finally, note that in (18), we did not minus the switching cost of the comparator sequence, i.e., P_T, that is because it is always smaller than $\sqrt{DT(1+P_T)}$ and does not affect the order.

Algorithm 3 Lookahead SAder: Meta-algorithm

Require: A step size β, and a set $\mathcal{H}$ containing step sizes for experts
1: Activate a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking the expert-algorithm for each step size $\eta \in \mathcal{H}$
2: Sort step sizes in ascending order $\eta_1 \leq \eta_2 \leq \cdots \leq \eta_N$, and set $w_0^{\eta_i} = \frac{C}{i(i+1)}$
3: **for** $t = 1, \ldots, T$ **do**
4: Observe the loss function $f_t(\cdot)$ and send it to each expert E^η
5: Receive $\mathbf{x}_t^\eta$ from each expert E^η
6: Update the weight of each expert by (20)
7: Output the weighted average $\mathbf{x}_t = \sum_{\eta \in \mathcal{H}} w_t^\eta \mathbf{x}_t^\eta$
8: **end for**

Algorithm 4 Lookahead SAder: Expert-algorithm

Require: The step size η
1: **for** $t = 1, \ldots, T$ **do**
2: Receive the loss $f_t(\cdot)$ from the meta-algorithm
3: Solve the optimization problem in (21) to obtain $\mathbf{x}_t^\eta$
4: Submit $\mathbf{x}_t^\eta$ to the meta-algorithm
5: **end for**

4.2 The lookahead setting

It is interesting to investigate whether we can do better if the hitting cost is available before predictions. In this case, we propose a *lookahead* version of SAder, and demonstrate that the regret bound remains on the same order, but Assumption 2 can be dropped. That is, the gradient of the function could be unbounded, and thus the function could also be unbounded.

Meta-algorithm We design a lookahead version of Hedge, and summarize it in Algorithm 3. Compared with Algorithm 1, we make the following modifications.

- In the t-th round, the meta-algorithm first sends $f_t(\cdot)$ to all experts so that they can also benefit from the prior knowledge of $f_t(\cdot)$ (Step 4).
- After receiving the prediction from experts (Step 5), the meta-algorithm makes use of $f_t(\cdot)$ to determine the weights of experts (Step 6):

$$w_t^\eta = \frac{w_{t-1}^\eta e^{-\beta \ell_t(\mathbf{x}_t^\eta)}}{\sum_{\eta \in \mathcal{H}} w_{t-1}^\eta e^{-\beta \ell_t(\mathbf{x}_t^\eta)}} \tag{20}$$

where $\ell_t(\mathbf{x}_t^\eta)$ is defined in (15).

Expert-algorithm To exploit the hitting cost of the current round, we choose an instance of the greedy algorithm in (4) as the expert-algorithm, and summarize it in Algorithm 4. The input of the expert is its step size η. After receiving $f_t(\cdot)$ (Step 2), the expert solves the following optimization problem to obtain $\mathbf{x}_t^\eta$ (Step 3):

$$\min_{\mathbf{x} \in \mathcal{X}} \quad f_t(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_{t-1}^\eta\|^2. \tag{21}$$

We have the following theoretical guarantee of the lookahead SAder.

Theorem 5 *Set*

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\sqrt{\frac{D^2}{T}} \;\middle|\; i = 1, \ldots, N \right\} \tag{22}$$

where

$$N = \left\lceil \frac{1}{2}\log_2(1 + 2T) \right\rceil + 1, \ \textit{and}\ \beta = \frac{1}{D}\sqrt{\frac{2}{T}}$$

in Algorithm 3. Under Assumptions 1 and 3, for any *comparator sequence* $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{X}$*, the lookahead SAder satisfies*

$$\sum_{t=1}^{T} \Big(f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \Big) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$
$$\leq \frac{3}{2}\sqrt{T\Big(D^2 + 2D\sum_{t=1}^{T}\|\mathbf{u}_t - \mathbf{u}_{t-1}\|\Big)} + D\sqrt{\frac{T}{2}}\,[1 + 2\ln(k+1)]$$
$$= O\Big(\sqrt{T(1+P_T)} + \sqrt{T}(1 + \log\log P_T)\Big) = O\Big(\sqrt{T(1+P_T)}\Big)$$

where $\mathbf{x}_0 = 0$*, and* k *is defined in (19).*

Remark: Similar to SAder, the lookahead SAder also achieves an $O(\sqrt{T(1+P_T)})$ bound for dynamic regret with switching cost. In the lookahead setting, we do not need Assumption 2 any more, and the constants in Theorem 5 are independent from G.

Lower Bound To show the optimality of Theorem 5, we provide the lower bound of dynamic regret with switching cost under the lookahead setting.

Theorem 6 *For any online algorithm with lookahead ability and any* $\tau \in [0, TD]$*, there exists a sequence of functions* $f_1, \ldots, f_T$ *and a sequence of comparators* $\mathbf{u}_1, \ldots, \mathbf{u}_T$ *satisfying Assumptions 1 and 3 such that (i) the path-length of* $\mathbf{u}_1, \ldots, \mathbf{u}_T$ *is at most* τ *and (ii) the dynamic regret with switching cost w.r.t.* $\mathbf{u}_1, \ldots, \mathbf{u}_T$ *is at least* $\Omega(\sqrt{T(D^2 + D\tau)})$.

Remark: The above theorem indicates an $\Omega(\sqrt{T(1+P_T)})$ lower bound, which implies that the lookahead SAder is optimal up to constant factors. Thus, even in the lookahead setting, it is impossible to improve the $O(\sqrt{T(1+P_T)})$ upper bound.

5 Conclusion and future work

We investigate the problem of smoothed online learning (SOL), and derive constant competitive ratio or sublinear dynamic regret with switching cost. For competitive ratio, we demonstrate that the naive approach, which only minimizes the hitting cost, is $\max(1, \frac{2}{\alpha})$-competitive for α-polyhedral functions and $1 + \frac{4}{\lambda}$-competitive for λ-quadratic growth functions. Furthermore, we show that the greedy algorithm, which minimizes the weighted sum of the hitting cost and the switching cost, is $1 + \frac{2}{\sqrt{\lambda}}$-competitive for convex and λ-quadratic growth functions. For dynamic regret with switching cost, we propose smoothed Ader (SAder), which attains the optimal $O(\sqrt{T(1+P_T)})$ bound. We also develop a lookahead version of SAder to make use of the prior knowledge of the hitting cost, and establish an $\Omega(\sqrt{T(1+P_T)})$ lower bound.

The research on SOL is still on its early stage, and there are many open problems.

1. Although we can upper bound the sum of the hitting cost and the switching cost, we do not have a direct control over the switching cost. However, in many real problems, there may exist a hard constraint on the switching cost, motivating the study of switch-constrained online learning, in which the times of switches are limited [Altschuler and Talwar, 2018, Chen et al., 2020]. It would be interesting to investigate how to impose a budget on the switching cost [Wang et al., 2021].
2. This work investigates competitive ratio and dynamic regret with switching cost separately. To bound the two metrics simultaneously, one possible way is to create two experts which are designed for competitive ratio and dynamic regret with switching cost respectively, and then aggregate their predictions by the meta-algorithm of Daniely and Mansour [2019, Algorithm 2]. But we need to assume the hitting cost is bounded, because that meta-algorithm cannot make use of the lookahead ability.
3. As aforementioned, for polyhedral functions and quadratic growth functions, the best competitive ratio is obtained by the naive approach which ignores the switching cost, and this fact is counterintuitive. To better understand the challenge, it is important to reveal the lower bound of polyhedral functions and quadratic growth functions.

Acknowledgments and Disclosure of Funding

The authors would like to thank Yuxuan Xiang for discussions about Theorem 4. Funding in direct support of this work: NSFC grant 62122037 and 61921006, JiangsuSF grant BK20200064.

References

R. Agrawal, M. Hegde, and D. Teneketzis. Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic Reports*, 29(4):437–459, 1990.

J. Altschuler and K. Talwar. Online learning over a finite action set with limited switching. In *Proceedings of the 31st Annual Conference on Learning Theory*, pages 1569–1573, 2018.

L. Chen, Q. Yu, H. Lawrence, and A. Karbasi. Minimax regret of switching-constrained online convex optimization: No phase transition. In *Advances in Neural Information Processing Systems 33*, pages 3477–3486, 2020.

N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.

N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman. Using predictions in online optimization: Looking forward with an eye on the past. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pages 193–206, 2016.

N. Chen, G. Goel, and A. Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *Proceedings of the 31st Conference on Learning Theory*, pages 1574–1594, 2018.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006.

A. Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2250–2259, 2020.

A. Daniely and Y. Mansour. Competitive ratio vs regret minimization: achieving the best of both worlds. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pages 333–368, 2019.

O. Dekel, J. Ding, T. Koren, and Y. Peres. Bandits with switching costs: $t^{2/3}$ regret. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 459–467, 2014.

D. Drusvyatskiy and A. S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3): 592–606, 2012.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

J. Friedman and N. Linial. On convex body chasing. *Discrete & Computational Geometry*, 9:293–321, 1993.

G. Goel and A. Wierman. An online algorithm for smoothed regression and lqr control. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pages 2504–2513, 2019.

G. Goel, N. Chen, and A. Wierman. Thinking fast and slow: Optimization decomposition across timescales. In *Proceedings of the 56th IEEE Conference on Decision and Control*, pages 1291–1298, 2017.

G. Goel, Y. Lin, H. Sun, and A. Wierman. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. In *Advances in Neural Information Processing Systems 32*, pages 1875–1885, 2019.

S. Guha and K. Munagala. Multi-armed bandits with metric switching costs. In *Proceedings of the 36th International Colloquium on Automata, Languages, and Programming*, pages 496–507, 2009.

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

E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 421–436, 2011.

L. Huang and M. J. Neely. Delay reduction via lagrange multipliers in stochastic network optimization. *IEEE Transactions on Automatic Control*, 56(4):842–857, 2011.

A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan. Online optimization: Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 398–406, 2015.

V. Joseph and G. de Veciana. Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications*, pages 567–575, 2012.

S.-J. Kim and G. B. Giannakis. Real-time electricity pricing for demand response using online convex optimization. In *Proceedings of the 2012 IEEE PES Conference on Innovative Smart Grid Technologies*, pages 1–5, 2014.

T. Kim, Y. Yue, S. Taylor, and I. Matthews. A decision tree framework for spatiotemporal sequence prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 577–586, 2015.

T. Koren, R. Livni, and Y. Mansour. Multi-armed bandits with metric movement costs. In *Advances in Neural Information Processing Systems 30*, pages 4119–4128. 2017a.

T. Koren, R. Livni, and Y. Mansour. Bandits with movement costs and adaptive pricing. In *Proceedings of the 30th Annual Conference on Learning Theory*, pages 1242–1268, 2017b.

W. H. Kwon and S. Han. *Receding Horizon Control*. Springer, 2005.

Y. Li and N. Li. Leveraging predictions in smoothed online convex optimization via gradient-based algorithms. In *Advances in Neural Information Processing Systems 33*, pages 14520–14531, 2020.

Y. Li, G. Qu, and N. Li. Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit. In *the 2018 Annual American Control Conference*, pages 3008–3013, 2018.

M. Lin, A. Wierman, L. L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proceedings of the 30th IEEE International Conference on Computer Communications*, pages 1098–1106, 2011.

M. Lin, Z. Liu, A. Wierman, and L. L. Andrew. Online algorithms for geographical load balancing. In *Proceedings of the 2012 International Green Computing Conference*, pages 1–10, 2012.

Y. Lin, G. Goel, and A. Wierman. Online optimization with predictions and non-convex losses. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):18:1–18:32, 2020.

A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 7195–7201, 2016.

I. Necoara, Y. Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1):69–107, 2019.

M. Sellke. Chasing convex bodies optimally. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1509–1518, 2020.

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

Y. Wan, B. Xue, and L. Zhang. Projection-free online learning in dynamic environments. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 10067–10075, 2021.

G. Wang, Y. Wan, T. Yang, and L. Zhang. Online convex optimization with continuous switching constraint. In *Advances in Neural Information Processing Systems 34*, 2021.

T. Yang, L. Zhang, R. Jin, and J. Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 449–457, 2016.

F. Zanini, D. Atienza, G. De Micheli, and S. P. Boyd. Online convex optimization-based algorithm for thermal management of mpsocs. In *Proceedings of the 20th Symposium on Great Lakes Symposium on VLSI*, pages 203–208, 2010.

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

L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems 30*, pages 732–741, 2017.

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

L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5882–5891, 2018b.

L. Zhang, S. Lu, and T. Yang. Minimizing dynamic regret and adaptive regret simultaneously. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 309–319, 2020.

P. Zhao and L. Zhang. Improved analysis for dynamic regret of strongly convex and smooth functions. In *Proceedings of the 3rd Annual Learning for Dynamics and Control Conference*, pages 48–59, 2021.

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

P. Zhao, Y.-X. Wang, and Z.-H. Zhou. Non-stationary online learning with memory and non-stochastic control. *ArXiv e-prints*, arXiv:2102.03758, 2021.

Y. Zhao, Q. Zhao, X. Zhang, E. Zhu, X. Liu, and J. Yin. Understand dynamic regret with switching cost for online decision making. *ACM Transactions on Intelligent Systems and Technology*, 11(3), 2020b.

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