Minimizing Dynamic Regret and Adaptive Regret Simultaneously

Lijun Zhang* **Shiyin Lu*** **Tianbao Yang†**

*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

†Department of Computer Science, The University of Iowa, Iowa City, IA 52242, USA

{zhanglj, lusy}@lamda.nju.edu.cn, tianbao-yang@uiowa.edu

Abstract

Regret minimization is treated as the golden rule in the traditional study of online learning. However, regret minimization algorithms tend to converge to the static optimum, thus being suboptimal for changing environments. To address this limitation, new performance measures, including dynamic regret and adaptive regret have been proposed to guide the design of online algorithms. The former one aims to minimize the global regret with respect to a sequence of changing comparators, and the latter one attempts to minimize every local regret with respect to a fixed comparator. Existing algorithms for dynamic regret and adaptive regret are developed independently, and only target one performance measure. In this paper, we bridge this gap by proposing novel online algorithms that are able to minimize the dynamic regret and adaptive regret *simultaneously*. In fact, our theoretical guarantee is even stronger in the sense that one algorithm is able to minimize the dynamic regret over *any* interval.

1 Introduction

Online convex optimization (OCO) is a powerful framework for sequential decision making and has found a variety of applications (Hazan, 2016). The protocol of OCO can be viewed as a repeated game between a learner and an adversary: In each round $t = 1, 2, \ldots, T$, the learner selects an action $\mathbf{w}_t$ from a convex feasible set Ω, and at the same time the adversary chooses a convex loss function $f_t(\cdot) : \Omega \mapsto \mathbb{R}$. Then, the function is revealed to the learner who incurs an instantaneous loss $f_t(\mathbf{w}_t)$. The goal of the learner is to minimize the regret:

$$\mathrm{R}(T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\Omega} \sum_{t=1}^{T} f_t(\mathbf{w}) \tag{1}$$

which compares the cumulative loss of the learner to that of the best fixed action in hindsight, and is typically referred to as *static* regret since the comparator is time-invariant.

Over the past decades, static regret has been extensively studied and algorithms with minimax optimal regret bounds have been developed (Zinkevich, 2003; Hazan et al., 2007; Shalev-Shwartz et al., 2007; Bartlett et al., 2008; Srebro et al., 2010; Shalev-Shwartz, 2011). However, the metric of static regret is only meaningful for stationary environments, and low static regret does not necessarily imply a good performance in changing environments since the time-invariant comparator in (1) may behave badly. To address this limitation, recent studies have introduced more stringent performance metrics, including dynamic regret and strongly adaptive regret, to measure the learner's performance.

The dynamic regret is defined as the difference between the cumulative loss of the learner and that of a sequence of comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \Omega$ (Zinkevich, 2003):

$$\text{D-R}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t). \tag{2}$$

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. A remarkable example is given by Zinkevich (2003), who introduces the notion of path-length defined in (4) to measure the temporal variability of the comparator sequence, and derives an $O(\sqrt{T}(1 + P_T))$ dynamic regret bound, where P_T is the path-length. Very recently, Zhang et al. (2018a) improve this result to be optimal by establishing an

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

$O(\sqrt{T(1+P_T)})$ upper bound as well as a matching lower bound.

The strongly adaptive regret evaluates the learner's performance on each time interval of length τ, and is defined as the maximum static regret over these intervals (Daniely et al., 2015):

$$\begin{aligned} &\text{SA-R}(T,\tau) \\ =& \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\Omega} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right). \end{aligned} \tag{3}$$

In the above definition, since the benchmark action $\mathbf{w}$ that minimizes the cumulative loss over interval $[s, s+\tau-1]$ typically varies with s, the learner is essentially competing with changing comparators. The state-of-the-art strongly adaptive regret bound is $O(\sqrt{\tau \log T})$ (Jun et al., 2017a) which matches the minimax static regret over a fixed interval up to a logarithmic factor (Abernethy et al., 2008).

Dynamic regret handles changing environments from a global prospective, as it measures the performance over the whole interval but allows the comparator changes over time. By contrast, adaptive regret takes a local perspective, since it focuses on short intervals with a fixed comparator but allows the interval changes over time. Although Zhang et al. (2018a) demonstrate that it is possible to derive dynamic regret from adaptive regret, their dynamic regret bound only takes a special form, and thus it does not mean adaptive regret is more fundamental than dynamic regret. Since dynamic regret and adaptive regret reflect different perspectives and are complementary to each other, it is appealing to ask whether we can minimize these two metrics simultaneously. Unfortunately, existing algorithms for minimizing dynamic regret and adaptive regret are developed independently and differ significantly.

In this paper, we propose novel algorithms that minimize the dynamic regret and adaptive regret simultaneously. Our methods follow the framework of "prediction with expert advice" (Cesa-Bianchi and Lugosi, 2006) and share a similar hierarchical structure: a series of expert algorithms configured with different lifetimes running in parallel, and an expert-tracking algorithm that combines the actions of all active experts. Specifically, the first method uses the simple online gradient descent (OGD) as the expert algorithm but manages the lifetime of experts through specifically designed intervals. On the contrary, the second method utilizes standard techniques to activate and deactivate experts, at the cost of a more complicated expert algorithm. Theoretical analysis shows that both methods attain the state-of-the-art $O(\sqrt{\tau \log T})$ adaptive regret, and they achieve $O(\sqrt{T(1+P_T)\log T})$ and $O(\sqrt{T(\log T+P_T)})$ dynamic regrets, respectively. Furthermore, the second method enjoys an even stronger theoretical guarantee: it can minimize the dynamic regret over any interval.

2 Related Work

In this section, we briefly review related work in dynamic regret and adaptive regret for OCO.

2.1 Dynamic Regret

Dynamic regret is first introduced by Zinkevich (2003), who proposes to use the *path-length*

$$P_T = \sum_{t=1}^{T} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|_2 \tag{4}$$

to measure the performance. Specifically, Zinkevich (2003) demonstrates that OGD with a constant step size attains a dynamic regret of $O(\sqrt{T}(1+P_T))$ for *any* sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T$. This upper bound is adaptive in the sense that it automatically becomes tighter when the comparators change slowly. Another regularity of the comparator sequence is defined as

$$P'_T = \sum_{t=1}^{T} \|\mathbf{u}_{t+1} - \Phi_t(\mathbf{u}_t)\|_2$$

where $\Phi_t(\cdot)$ is a dynamic model that predicts a reference point for the t-th round. Hall and Willett (2013) develope a novel algorithm named dynamic mirror descent and prove a dynamic regret of $O(\sqrt{T}(1+P'_T))$. An $\Omega(\sqrt{T(1+P_T)})$ lower bound of dynamic regret is established by Zhang et al. (2018a), which indicates the results of Zinkevich (2003) and Hall and Willett (2013) are far away from the optimum. To address this limitation, Zhang et al. (2018a) develop an optimal algorithm, namely adaptive learning for dynamic environment (Ader), which attains an $O(\sqrt{T(1+P_T)})$ bound in the general case, and an $O(\sqrt{T(1+P'_T)})$ bound when a sequence of dynamical models is available.

Deviating from the definition in (2), most studies on dynamic regret only consider a restricted form, defined with respect to a sequence of minimizers of the loss functions due to its greater mathematical tractability (Jadbabaie et al., 2015; Besbes et al., 2015; Yang et al., 2016; Mokhtari et al., 2016; Zhang et al., 2017):

$$\begin{aligned} \text{D-R}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) &= \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_t^*) \\ &= \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} \min_{\mathbf{w}\in\Omega} f_t(\mathbf{w}) \end{aligned} \tag{5}$$

where $\mathbf{w}_t^* \in \operatorname{argmin}_{\mathbf{w}\in\Omega} f_t(\mathbf{w})$ is a minimizer of $f_t(\cdot)$ over domain Ω. Although one can show that $\text{D-R}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) \geq \text{D-R}(\mathbf{u}_1, \ldots, \mathbf{u}_T)$, it does not imply the former one is stronger since an upper bound for $\text{D-R}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ could be very loose for $\text{D-R}(\mathbf{u}_1, \ldots, \mathbf{u}_T)$. In fact, the definition in (2) is more general since it holds for any sequence of comparators, and thus includes the static regret in (1) and the restricted dynamic regret in (5) as special cases.

Let P_T^* be the path-length of the minimizer sequence $\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*$. When the loss functions are strongly convex and smooth, Mokhtari et al. (2016) show that the restricted dynamic regret of OGD is $O(P_T^*)$. This rate is also attainable for convex and smooth functions under the condition that the minimizers lie in the interior of Ω (Yang et al., 2016). Zhang et al. (2017) introduce the squared path-length:

$$S_T^* = \sum_{t=1}^{T} \|\mathbf{w}_{t+1}^* - \mathbf{w}_t^*\|_2^2$$

which could be much smaller than P_T^* in the case that minimizers move slowly. They demonstrate that the restricted dynamic regret bound for strongly convex functions could be improved to $O(\min(P_T^*, S_T^*))$.

Instead of measuring the complexity of the comparator sequence, Besbes et al. (2015) propose to evaluate the movement of the loss functions as follows:

$$F_T = \sum_{t=1}^{T} \sup_{\mathbf{w}\in\Omega} |f_{t+1}(\mathbf{w}) - f_t(\mathbf{w})|. \tag{6}$$

Besbes et al. (2015) show that a restarted OGD algorithm equipped with a prior knowledge of an upper bound $V_T \geq F_T$ achieves $O(V_T^{1/3}T^{2/3})$ and $O(\sqrt{V_T T})$ dynamic regret for convex functions and strongly convex functions, respectively. However, these bounds depend on the predetermined V_T rather than the actual F_T, and thus are not adaptive.

2.2 Adaptive Regret

In their seminal work, Hazan and Seshadhri (2007) define adaptive regret as

$$\text{A-R}(T) = \max_{[s,q]\subseteq[T]} \left(\sum_{t=s}^{q} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\Omega} \sum_{t=s}^{q} f_t(\mathbf{w}) \right) \tag{7}$$

which is the maximum regret over any contiguous interval. They develop a novel algorithm named as follow the leading history (FLH), which runs an instance of low-regret algorithm in each round as an expert, and then combines them with an expert-tracking method. To improve the efficiency, Hazan and Seshadhri (2007) deploy a data-streaming technique to prune the set of experts, and as a result only $O(\log t)$ experts are stored at round t. The efficient version of FLH attains $O(d\log^2 T)$ and $O(\sqrt{T\log^3 T})$ adaptive regrets for exponentially concave functions and convex functions, respectively (Hazan and Seshadhri, 2009).

However, the adaptive regret in (7) does not respect short intervals well. For example, the $O(\sqrt{T\log^3 T})$ adaptive regret of convex functions is vacuous for intervals of size $O(\sqrt{T})$. To avoid this limitation, Daniely et al. (2015) propose the strongly adaptive regret in (3), which emphasizes the dependence on the interval length. The strongly adaptive algorithm of Daniely et al. (2015) shares a similar structure to that of FLH (Hazan and Seshadhri, 2007), but with the following differences.

(i) Daniely et al. (2015) construct a set of geometric covering (GC) intervals, and run an instance of low-regret algorithm for each interval as an expert.

(ii) A new meta-algorithm named as strongly adaptive online learner (SOAL) is used to combine experts.

The GC intervals are defined as

$$\mathcal{I} = \bigcup_{k\in\mathbb{N}\cup\{0\}} \mathcal{I}_k \tag{8}$$

where for all $k \in \mathbb{N}\cup\{0\}$, $\mathcal{I}_k = \{[i\cdot 2^k, (i+1)\cdot 2^k - 1] : i \in \mathbb{N}\}$. For convex functions, Daniely et al. (2015) establish an $O(\sqrt{\tau}\log T)$ strongly adaptive regret. In a subsequent work, Jun et al. (2017a) design a new meta-algorithm named as sleeping coin betting (CB), and improve the strongly adaptive regret to $O(\sqrt{\tau\log T})$. The adaptive regret of convex and smooth functions are studied by Jun et al. (2017b) and Zhang et al. (2019).

2.3 The Relationship between Dynamic Regret and Adaptive Regret

In the setting of prediction with expert advice (PEA), dynamic regret is usually referred to as tracking regret or shifting regret (Littlestone and Warmuth, 1994; Herbster and Warmuth, 1998; György et al., 2012). In this case, it has been proved that the tracking regret can be derived from the adaptive regret (Adamskiy et al., 2012; Cesa-bianchi et al., 2012; Daniely et al., 2015). In particular, Theorem 4 of Luo and Schapire (2015) indicates that it is possible to bound the dynamic regret by the adaptive regret and the following variation:

$$V_T = \sum_{t=1}^{T}\sum_{i=1}^{N}[u_{t+1,i} - u_{t,i}]_+$$

where N is the number of experts, $u_{t,i}$ is the i-th component of $\mathbf{u}_t$, and $[x]_+ = \max(0, x)$. Thus, for PEA, it is commonly believed that adaptive regret is more fundamental.

In the setting of OCO, to the best of our knowledge, there is only one work (Zhang et al., 2018b) that has investigated the relationship between dynamic regret and adaptive regret. Let $I_1 = [s_1, q_1], \ldots, I_k = [s_k, q_k]$ be a partition of $[1, T]$ and for each interval I_i, define the local variation of functions as

$$F_T(i) = \sum_{t=s_i}^{q_i-1} \sup_{\mathbf{w}\in\Omega} |f_{t+1}(\mathbf{w}) - f_t(\mathbf{w})|.$$

Zhang et al. (2018b) prove that the restricted dynamic regret can be upper bounded in terms of the strongly adaptive regret and $F_T(i)$ as follows:

$$\begin{aligned}&\text{D-R}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)\\ \leq &\min_{I_1,\ldots,I_k} \sum_{i=1}^{k} \big(\text{SA-R}(T, |I_i|) + 2|I_i| \cdot F_T(i)\big).\end{aligned}$$

As can be seen, this result is only applicable to the restricted dynamic regret instead of the general dynamic regret considered in this paper.

Following the analysis of Zhang et al. (2018b), we have tried to upper bound the dynamic regret by the strongly adaptive regret and the path-length.

Theorem 1 *Assuming all the online functions are G-Lipschitz continuous, we have*

$$\begin{aligned}&\text{D-R}(\mathbf{u}_1, \ldots, \mathbf{u}_T)\\ \leq &\min_{I_1,\ldots,I_k} \sum_{i=1}^{k} \big(\text{SA-R}(T, |I_i|) + G|I_i| \cdot P_T(i)\big)\end{aligned} \tag{9}$$

where $P_T(i) = \sum_{t=s_i}^{q_i-1} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|_2$. Combining with the adaptive regret of convex functions (Jun et al., 2017a), we obtain the following dynamic regret for convex functions

$$\begin{aligned}&\text{D-R}(\mathbf{u}_1, \ldots, \mathbf{u}_T)\\ =&O\left(\max\left\{\sqrt{T\log T}, T^{2/3}P_T^{1/3}\log^{1/3} T\right\}\right).\end{aligned} \tag{10}$$

The above theorem shows that although the strongly adaptive regret can be used to control the dynamic regret, it may not be able to give the optimal result, since the regret bound in (10) is much worse than the $O(\sqrt{T(1+P_T)})$ bound of Zhang et al. (2018a). Thus, in the setting of OCO, which one of dynamic regret and adaptive regret is more fundamental remains an open problem. Note that our algorithms are able to minimize the dynamic regret and adaptive regret simultaneously. So, no matter which performance measure is stronger, they are always meaningful.

Algorithm 1 Online Gradient Descent (OGD)

1: **Input:** Initial point $\mathbf{w}_1$, and step size η
2: **for** $t = 1$ **to** T **do**
3: Submit $\mathbf{w}_t$, and then receive $f_t(\cdot)$
4: Suffer a loss $f_t(\mathbf{w}_t)$ and update as

$$\mathbf{w}_{t+1} = \Pi_\Omega\big[\mathbf{w}_t - \eta\nabla f_t(\mathbf{w}_t)\big]$$

5: **end for**

3 Our Methods

In this section, we present our online algorithms that are able to minimize the dynamic regret and adaptive regret simultaneously. The first method uses a two-layer structure, but with specifically designed components. In contrast, the second method has a three-layer structure, but with standard techniques that are easy to comprehend.

Assumption 1 *The gradients of all functions are bounded by G, i.e.,*

$$\max_{\mathbf{w}\in\Omega} \|\nabla f_t(\mathbf{w})\|_2 \leq G, \ \forall t \in [T]. \tag{11}$$

Assumption 2 *The domain Ω contains the origin $\mathbf{0}$, and its diameter is bounded by D, i.e.,*

$$\max_{\mathbf{w},\mathbf{w}'\in\Omega} \|\mathbf{w} - \mathbf{w}'\|_2 \leq D. \tag{12}$$

Assumption 3 *The value of each function belongs to $[0, 1]$, i.e.,*

$$0 \leq f_t(\mathbf{w}) \leq 1, \ \forall \mathbf{w} \in \Omega, t \in [T].$$

As long as the loss functions are bounded, they can always be scaled and restricted to $[0, 1]$.

3.1 The First Method

Our first method follows the framework of adaptive algorithms for convex functions (Daniely et al., 2015; Jun et al., 2017a). On one hand, the proposed method inherits their ability to minimize the adaptive regret. On the other hand, our method contains new features so that the dynamic regret can also be minimized.

We take the classical online gradient descent (OGD) as the expert algorithm, and present the procedure in Algorithm 1. After receiving the loss function $f_t(\cdot)$, OGD performs gradient descent to update $\mathbf{w}_t$:

$$\mathbf{w}_{t+1} = \Pi_\Omega\big[\mathbf{w}_t - \eta\nabla f_t(\mathbf{w}_t)\big]$$

where $\Pi_\Omega[\cdot]$ denotes the projection onto the nearest point in Ω, and $\eta > 0$ is the step size. The following static regret bound of OGD is well-known (Zinkevich, 2003).

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	$\cdots$
$\mathcal{D}_0$	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	$\cdots$
$\mathcal{D}_1$	[	]	[	]	[	]	[	]	[	]	[	]	[	]	[	]	[	]	[	$\cdots$
$\mathcal{D}_2$	[			]	[			]	[			]	[			]	[			$\cdots$
$\mathcal{D}_3$	[							]	[							]	[			$\cdots$

Figure 1: Dense geometric covering (DGC) intervals.

Theorem 2 *Under Assumptions 1 and 2, we have*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\Omega}\sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{D^2}{2\eta} + \frac{\eta T G^2}{2} = DG\sqrt{T}$$

where the step size is set as $\eta = D/(G\sqrt{T})$.

Similar to existing adaptive algorithms, we will run multiple instances of OGD over a set of intervals. Instead of using the GC intervals of Daniely et al. (2015), we construct a dense version of GC intervals:

$$\mathcal{D} = \bigcup_{k\in\mathbb{N}\cup\{0\}\& 2^k\leq T} \mathcal{D}_k$$

where for all $k \in \mathbb{N}\cup\{0\}$, $\mathcal{D}_k = \left\{I_k^i = [(i-1)\cdot 2^k+1, i\cdot 2^k] : i\in\mathbb{N}\right\}$. We present a graphical illustration of our dense geometric covering (DGC) intervals in Fig. 1. Compared with the original GC intervals, the main difference is that $\mathcal{I}_k$ in GC intervals is a partition of $\mathbb{N}\setminus\{1,\cdots,2^k-1\}$ to consecutive intervals of length 2^k, while $\mathcal{D}_k$ in our DGC intervals is a partition of $\mathbb{N}$ to consecutive intervals of length 2^k. Furthermore, we assume the total number of iterations T is given beforehand, so we only construct intervals whose lengths are not larger than T.

For each interval $I_k^i \in \mathcal{D}$, we will run an instance of OGD. According to Theorem 2, we set the step size as $\eta = D/(G\sqrt{2^k})$, which is able to minimize the static regret over I_k^i. For the initial solution, we choose one of the following two ways:

- If $i = 1$, we set the initial solution as an arbitrary point in Ω.
- If $i > 1$, we set the initial solution as the last output of the OGD associated with I_k^{i-1}.

In other words, the expert associated with each interval in $\mathcal{D}_k$, except the first one, is warm started by initiating OGD with the solution of the previous expert.

To combine the actions of all experts, we choose the AdaNormalHedge (Luo and Schapire, 2015) as our meta-algorithm. AdaNormalHedge is a parameter-free expert-tracking algorithm, which shares a similar regret bound as the sleeping CB (Jun et al., 2017a), but with simpler updating rules. It makes use of a potential function:

$$\Phi(R, C) = \exp\left(\frac{[R]_+^2}{3C}\right)$$

where $[x]_+ = \max(0, x)$ and $\Phi(0,0)$ is defined to be 1, and a weight function with respect to this potential:

$$w(R, C) = \frac{1}{2}\big(\Phi(R+1, C+1) - \Phi(R-1, C+1)\big).$$

The complete procedure is named as Adaptive Online learning with Dynamic regret (AOD), and summarized in Algorithm 2. We explain the main steps below. In each round t, we will maintain a set of active experts, denoted by $\mathcal{A}_t$. To simplify the notation, let's define the set of intervals in $\mathcal{D}$ that start from t as $\mathcal{C}_t$, i.e.,

$$\mathcal{C}_t = \{I | I\in\mathcal{D}, t\in I, (t-1)\notin I\}. \tag{13}$$

Algorithm 2 Adaptive Online learning with Dynamic regret (AOD)

for $t = 1$ **to** T **do**
  for $I \in \mathcal{C}_t$ **do**
    Create an expert E_I which runs OGD, and set $R_{t-1,I} = C_{t-1,I} = 0$
    Set the step size of E_I as $\eta = D/(G\sqrt{|I|})$
    if $t = 1$ **then**
      Set the initial solution of E_I arbitrarily
    else
      Identify the expert $E_J \in \mathcal{A}_t$ such that $|J| = |I|$
      Set the initial solution of E_I to be the output of expert E_J, denoted by $\mathbf{w}_{t,J}$
      Remove E_J from $\mathcal{A}_t$
    end if
    Add expert E_I to the set of active experts $\mathcal{A}_t$
  end for
  Receive the action $\mathbf{w}_{t,I}$ of each expert $E_I \in \mathcal{A}_t$, and calculate its weight $p_{t,I}$ according to (14)
  Submit $\mathbf{w}_t$ defined in (15) and then receive $f_t(\cdot)$
  For each $E_I \in \mathcal{A}_t$, update

$$\begin{aligned} R_{t,I} =& R_{t-1,I} + f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t,I}),\\ C_{t,I} =& C_{t-1,I} + |f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t,I})| \end{aligned}$$

  Pass $f_t(\cdot)$ to each expert $E_I \in \mathcal{A}_t$
end for

For each interval I in $\mathcal{C}_t$, we create an expert E_I which runs an instance of OGD, and initialize two variables $R_{t-1,I}$ and $C_{t-1,I}$ that are used to calculate the weight of E_I (Step 3). The step size of E_I is set in Step 4, and the initial solution is set in Step 6 or 9. Note that when $t > 1$, for each interval $I \in \mathcal{C}_t$, there must be an expert $E_J \in \mathcal{A}_t$ such that $|J| = |I|$ (Step 8). Then, we use the output of E_J to initialize E_I (Step 9), and remove E_J from $\mathcal{A}_t$ (Step 10). The new expert E_I is added to $\mathcal{A}_t$ (Step 12).

In Step 14, we receive the action $\mathbf{w}_{t,I}$ of each active expert $E_I \in \mathcal{A}_t$, and assign the following weight to E_I

$$p_{t,I} = \frac{w(R_{t-1,I}, C_{t-1,I})}{\sum_{E_I \in \mathcal{A}_t} w(R_{t-1,I}, C_{t-1,I})} \tag{14}$$

where

$$R_{t-1,I} = \sum_{u=\min I}^{t-1} f_u(\mathbf{w}_u) - f_u(\mathbf{w}_{u,I}),$$
$$C_{t-1,I} = \sum_{u=\min I}^{t-1} \left| f_u(\mathbf{w}_u) - f_u(\mathbf{w}_{u,I}) \right|,$$

and $\min I$ denotes the starting round of interval I. In Step 15, we submit the weighted average

$$\mathbf{w}_t = \sum_{E_I \in \mathcal{A}_t} p_{t,I} \mathbf{w}_{t,I} \tag{15}$$

as the output, and receive the loss function $f_t(\cdot)$. In Step 16, we update variables that are used to calculate probabilities in (14). Finally, we reveal the function $f_t(\cdot)$ to all active experts so that they can make predictions for the next round.

We first present the strongly adaptive regret of AOD.

Theorem 3 *Under Assumptions 1, 2, and 3, the strongly adaptive regret of AOD in Algorithm 2 satisfies*

$$\text{SA-R}(T, \tau) \leq 8 \left(\sqrt{3c(T)} + DG \right) \sqrt{\tau} = O\left(\sqrt{\tau \log T} \right)$$

where

$$\begin{aligned} & c(T) \\ \leq & 1 + \ln T + \ln(1 + \log_2 T) + \ln \frac{5 + 3\ln(1+T)}{2}. \end{aligned} \tag{16}$$

Note that our strongly adaptive regret matches the state-of-the-art result of Jun et al. (2017a) exactly. The main advantage is that AOD is also equipped with a dynamic regret bound, which is nearly optimal.

Theorem 4 *Under Assumptions 1, 2, and 3, for* any *comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \Omega$, AOD in Algorithm 2 satisfies*

$$\begin{aligned} & \text{D-R}(\mathbf{u}_1, \ldots, \mathbf{u}_T) \\ \leq & \left(\frac{3DG}{2} + \frac{5G}{2}\sqrt{DP_T} + \sqrt{6c(T)\left(1 + \frac{2P_T}{D}\right)} \right) \sqrt{T} \\ = & O\left(\sqrt{T(1+P_T)\log T} \right) \end{aligned}$$

where $c(T)$ is given in (16).

Remark: The dynamic regret of AOD matches the $\Omega(\sqrt{T(1+P_T)})$ lower bound up to a logarithmic factor, and is slightly worse than the $O(\sqrt{T(1+P_T)})$ bound of Ader (Zhang et al., 2018a). However, Ader is not equipped with any adaptive regret, while our AOD achieves the state-of-the-art adaptive regret as shown in Theorem 3.

Complexity: The computational complexity of AOD in each round is $O(\log T)$, since it needs to maintain $O(\log T)$ experts (i.e., instances of OGD), and the complexity of each expert is $O(1)$.

3.2 The Second Method

One limitation of AOD is that the total number of iterations T needs to be known and fixed. In this section, we address this limitation by developing a three-layer algorithm, in which an additional layer is inserted to decouple the adaptive regret and the dynamic regret.

The basic idea is very simple. Instead of running OGD as the expert in our previous AOD algorithm, we use Ader (Zhang et al., 2018a), which is designed to minimize the dynamic regret, as the expert algorithm. The new algorithm is named as Adaptive Online learning based on Ader (AOA), and summarized in Algorithm 3. Because Ader itself is a two-layer algorithm, AOA is essentially a three-layer algorithm. The top layer takes responsibility for the adaptive regret, and the middle-layer is responsible for the dynamic regret. Because of this design, AOA is able to minimize the dynamic regret over any interval. In contrast, the top layer in AOD takes care of both the adaptive regret and the dynamic regret.

We create experts based on the original GC intervals in (8) (Daniely et al., 2015), because they can be constructed dynamically and do not need to know the total number of iterations T. Similar to the AOD algorithm, we use $\mathcal{A}_t$ to denote the set of active experts in round t, and $\widetilde{\mathcal{C}}_t$ to denote the set of intervals in $\mathcal{I}$ that start from t, i.e.,

$$\widetilde{\mathcal{C}}_t = \{I | I \in \mathcal{I}, t \in I, (t-1) \notin I\}.$$

For each interval I in $\widetilde{\mathcal{C}}_t$, we will create an expert E_I which runs an instance of Ader (Step 3), pass the interval length $|I|$ to E_I (Step 4), and add it to the set

Algorithm 3 Adaptive Online learning based on Ader (AOA)

1: **for** $t = 1$ **to** T **do**
2:   **for** $I \in \widetilde{\mathcal{C}}_t$ **do**
3:     Create an expert E_I, which runs Ader, and set $R_{t-1,I} = C_{t-1,I} = 0$
4:     Pass the interval length I to expert E_I
5:     Add expert E_I to the set of active experts $\mathcal{A}_t$
6:   **end for**
7:   Receive the action $\mathbf{w}_{t,I}$ of each expert $E_I \in \mathcal{A}_t$, and calculate its weight $p_{t,I}$ according to (14)
8:   Submit $\mathbf{w}_t$ defined in (15) and receive $f_t(\cdot)$
9:   Remove experts whose ending times are t

$$\mathcal{A}_t = \mathcal{A}_t \setminus \{E_I | t \in I, (t+1) \notin I\}$$

10:   For each $E_I \in \mathcal{A}_t$, update

$$\begin{aligned} R_{t,I} =& R_{t-1,I} + f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t,I}), \\ C_{t,I} =& C_{t-1,I} + |f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t,I})| \end{aligned}$$

11:   Pass $f_t(\cdot)$ to each expert $E_I \in \mathcal{A}_t$
12: **end for**

of active experts $\mathcal{A}_t$ (Step 5). As before, we combine the actions of all active experts by AdaNormalHedge (Steps 7 and 8). After submitting $\mathbf{w}_t$, AOA removes all the experts whose ending times are t (Step 9). All the remaining steps of AOA are identical to those of AOD.

For the sake of completeness, we present the procedure of Ader in Algorithm 4, and give a brief introduction. Ader takes the total number of iterations T as the input, and constructs a set of step sizes

$$\mathcal{H} = \left\{ \eta_i = \frac{2^{i-1} D}{G} \sqrt{\frac{7}{2T}} \middle| i = 1, \ldots, N \right\} \tag{17}$$

where $N = \lceil \frac{1}{2} \log_2(1 + 4T/7) \rceil + 1$ (Step 2). For each $\eta \in \mathcal{H}$, Ader creates an expert E_η by running an instance of OGD with step size η (Step 3).[1] The actions of experts are combined by the standard Hedge algorithm (Steps 7 and 8) with nonuniform initial weights (Step 4) (Freund and Schapire, 1997; Cesa-Bianchi and Lugosi, 2006).

We present the theoretical guarantee of AOA.

Theorem 5 *Under Assumptions 1, 2, and 3, for* any *interval* $I = [r, s] \subseteq \mathbb{N}$ *and* any *comparator sequence*

[1]The initial solution of OGD can be set arbitrarily.

Algorithm 4 Adaptive learning for dynamic environment (Ader)

1: **Input:** The total number of iterations T
2: Construct the set $\mathcal{H}$ according to (17)
3: Create a set of experts $\{E_\eta | \eta \in \mathcal{H}\}$ by running OGD with each step size $\eta \in \mathcal{H}$
4: Sort step sizes in ascending order $\eta_1 \leq \eta_2 \leq \cdots \leq \eta_N$, and set $p_{1,\eta_i} = \frac{C}{i(i+1)}$ where $C = 1 + \frac{1}{|\mathcal{H}|}$
5: **for** $t = 1, \ldots, T$ **do**
6:   Receive the action $\mathbf{w}_{t,\eta}$ from each expert E_η
7:   Submit

$$\mathbf{w}_t = \sum_{\eta \in \mathcal{H}} p_{t,\eta} \mathbf{w}_{t,\eta}$$

  and then receive $f_t(\cdot)$
8:   Update the weight of each expert by

$$p_{t+1,\eta} = \frac{p_{t,\eta} e^{-\alpha f_t(\mathbf{w}_{t,\eta})}}{\sum_{\mu \in \mathcal{H}} p_{t,\mu} e^{-\alpha f_t(\mathbf{w}_{t,\mu})}}$$

  where $\alpha = \sqrt{8/T}$
9:   Pass $f_t(\cdot)$ to each expert E_η
10: **end for**

$\mathbf{u}_r, \ldots, \mathbf{u}_s \in \Omega$, *AOA in Algorithm 3 satisfies*

$$\begin{aligned} & \sum_{t=r}^{s} f_t(\mathbf{w}_t) - \sum_{t=r}^{s} f_t(\mathbf{u}_t) \\ \leq & \left(14\sqrt{c'(s)} + 3\left[1 + 2\ln(k_I + 1)\right] + 23DG \right) \sqrt{|I|} \\ & + 5G\sqrt{DP_I}\sqrt{|I|} \\ =& O\left(\sqrt{|I|(\log s + P_I)} \right) \end{aligned}$$

where

$$\begin{aligned} c'(s) \leq & 1 + \ln s + \ln(1 + \log_2 s) + \ln \frac{5 + 3\ln(1+s)}{2}, \\ P_I =& \sum_{t=r}^{s} \|\mathbf{u}_{t+1} - \mathbf{u}_t\|_2, \\ k_I =& \left\lfloor \frac{1}{2} \log_2 \left(1 + \frac{4P_I}{7D} \right) \right\rfloor + 1. \end{aligned}$$

Remark: The above theorem indicates that our AOA algorithm can minimize the dynamic regret over any interval, which is a strong theoretical guarantee that allows us to derive either dynamic regret or adaptive regret. To derive dynamic regret over the whole interval, we set $I = [1, T]$, and then obtain an $O(\sqrt{T(\log T + P_T)})$ dynamic regret bound, which nearly matches the $\Omega(\sqrt{T(1 + P_T)})$ lower bound (Zhang et al., 2018a), and becomes optimal when $P_T = \Omega(\log T)$. To derive adaptive regret, we set $\mathbf{u}_r = \cdots = \mathbf{u}_s$ such that $P_I = 0$ and consider $s \leq T$, and then can prove SA-R$(T,\tau) = O\left(\sqrt{\tau \log T}\right)$ which matches the state-of-the-art result (Jun et al., 2017a) exactly.

Complexity: Note that AOA maintains $O(\log t)$ experts in the t-th round, and the complexity of each expert, which is an instance of Ader, is $O(\log t)$. So, the computational complexity of AOA in the t-round is $O(\log^2 t)$.

4 Analysis

Due to the limitation of space, we only prove Theorem 3, and the omitted proofs can be found in the full version (Zhang et al., 2020).

4.1 Proof of Theorem 3

We first present the meta-regret of AOD. Let $m(t)$ be the total number of experts created up to round t. It is easy to verify that

$$m(t) \leq t(1 + \log_2 T).$$

Then, according to Theorems 1&3 of Luo and Schapire (2015) and Jensen's inequality (Boyd and Vandenberghe, 2004), we have the following lemma.

Lemma 1 *Under Assumption 3, for any interval $J = [i,j] \in \mathcal{D}$, AOD satisfies*

$$\sum_{u=i}^{t} f_u(\mathbf{w}_u) - \sum_{u=i}^{t} f_u(\mathbf{w}_{u,J}) \leq \sqrt{3(t-i+1)c(t)}, \ \forall t \in J$$

where

$$\begin{split} c(t) \leq & 1 + \ln m(t) + \ln \frac{5 + 3\ln(1+t)}{2} \\ \leq & 1 + \ln t + \ln(1 + \log_2 T) + \ln \frac{5 + 3\ln(1+t)}{2}. \end{split}$$

We proceed to bound the adaptive regret of AOD. To this end, we first bound the regret of AOD over any interval $J = [i,j] \in \mathcal{D}$. By combining the meta-regret in Lemma 1 and the expert-regret in Theorem 2, we immediately have the following bound.

Lemma 2 *Under Assumptions 1, 2, and 3, for any interval $J = [i,j] \in \mathcal{D}$, AOD satisfies*

$$\sum_{t \in J} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \Omega} \sum_{t \in J} f_t(\mathbf{w}) \leq \left(\sqrt{3c(j)} + DG\right) \sqrt{|J|}.$$

We then extend the above regret bound to any interval $I = [r,s] \subseteq [T]$. To this end, we need the following lemma about the DGC intervals, which has a similar property as the original GC intervals (Daniely et al., 2015).

Lemma 3 *For any interval $[r,s] \subseteq [T]$, it can be partitioned into two sequences of disjoint and consecutive intervals, denoted by $I_{-p}, \ldots, I_0 \in \mathcal{D}$ and $I_1, \ldots, I_q \in \mathcal{D}$, such that*

$$|I_{-i}|/|I_{-i+1}| \leq 1/2, \ \forall i \geq 1$$

and

$$|I_i|/|I_{i-1}| \leq 1/2, \ \forall i \geq 2.$$

Then, based on Lemmas 2 and 3, we bound the regret with respect to any $\mathbf{w} \in \Omega$ over $I = [r,s]$ in the following way

$$\begin{split} & \sum_{t=r}^{s} f_t(\mathbf{w}_t) - \sum_{t=r}^{s} f_t(\mathbf{w}) \\ = & \sum_{i=-p}^{q} \left(\sum_{t \in I_i} f_t(\mathbf{w}_t) - \sum_{t \in I_i} f_t(\mathbf{w}) \right) \\ \leq & \sum_{i=-p}^{q} \left(\sqrt{3c(s)} + DG\right) \sqrt{|I_i|} \\ \leq & 2\left(\sqrt{3c(s)} + DG\right) \sum_{i=0}^{\infty} (2^{-i}|I|)^{1/2} \\ \leq & 8\left(\sqrt{3c(s)} + DG\right) \sqrt{I}. \end{split}$$

Thus, the strongly adaptive regret

$$\begin{split} & \text{SA-R}(T,\tau) \\ = & \max_{[s,s+\tau-1] \subseteq [T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \Omega} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{w}) \right) \\ \leq & 8\left(\sqrt{3c(T)} + DG\right) \sqrt{\tau}. \end{split}$$

5 Conclusion

Inspired by recent developments of dynamic regret and adaptive regret, this paper asks whether it is possible to bound them simultaneously. We provide affirmative answers by proposing novel algorithms that achieve this goal. The first method, namely AOD, runs multiple instances of OGD over specifically designed intervals, uses warm start to connect successive OGD's, and then combines multiple decisions by an expert-tracking algorithm. Theoretical analysis shows that AOD enjoys a tight adaptive regret and a nearly optimal dynamic regret. The second method, namely AOA, maintains multiple instances of Ader, and combines them in the same way as AOD. We demonstrate that AOA is equipped with a strong theoretical guarantee in the sense that it can minimize the dynamic regret over any interval.

Acknowledgements

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

References

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

D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk. A closer look at adaptive regret. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, pages 290–304, 2012.

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

O. Besbes, Y. Gur, and A. Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5): 1227–1244, 2015.

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

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

N. Cesa-bianchi, P. Gaillard, G. Lugosi, and G. Stoltz. Mirror descent meets fixed share (and feels no regret). In *Advances in Neural Information Processing Systems 25*, pages 980–988, 2012.

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

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.

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

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

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

E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 393–400, 2009.

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

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

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.

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

K.-S. Jun, F. Orabona, S. Wright, and R. Willett. Online learning for changing environments using coin betting. *Electronic Journal of Statistics*, 11(2):5282–5310, 2017b.

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

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

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.

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

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

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

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.

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 1330–1340, 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*, 2018b.

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

L. Zhang, S. Lu, and T. Yang. Minimizing dynamic regret and adaptive regret simultaneously. *ArXiv e-prints*, arXiv:2002.02085, 2020.

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