Dynamic Regret of Strongly Adaptive Methods

Lijun Zhang [1] **Tianbao Yang** [2] **Rong Jin** [3] **Zhi-Hua Zhou** [1]

Abstract

To cope with changing environments, recent developments in online learning have introduced the concepts of *adaptive* regret and *dynamic* regret independently. In this paper, we illustrate an intrinsic connection between these two concepts by showing that the dynamic regret can be expressed in terms of the adaptive regret and the functional variation. This observation implies that strongly adaptive algorithms can be directly leveraged to minimize the dynamic regret. As a result, we present a series of strongly adaptive algorithms that have small dynamic regrets for convex functions, exponentially concave functions, and strongly convex functions, respectively. To the best of our knowledge, this is the *first* time that exponential concavity is utilized to upper bound the dynamic regret. Moreover, all of those adaptive algorithms do not need any prior knowledge of the functional variation, which is a significant advantage over previous specialized methods for minimizing dynamic regret.

1. Introduction

Online convex optimization is a powerful paradigm for sequential decision making (Zinkevich, 2003). It can be viewed as a game between a learner and an adversary: In the t-th round, the learner selects a decision $\mathbf{w}_t \in \Omega$, simultaneously the adversary chooses a function $f_t(\cdot) : \Omega \mapsto \mathbb{R}$, and then the learner suffers an instantaneous loss $f_t(\mathbf{w}_t)$. This study focuses on the full-information setting, where the learner can query the value and gradient of f_t (Cesa-Bianchi & Lugosi, 2006). The goal of the learner is to minimize the cumulative loss over T periods . The standard performance measure is *regret*, which is the difference between the loss incurred by the learner and that of the best fixed decision in hindsight, i.e.,

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\Omega} \sum_{t=1}^{T} f_t(\mathbf{w}).$$

The above regret is typically referred to as *static* regret in the sense that the comparator is time-invariant. The rationale behind this evaluation metric is that one of the decision in Ω is reasonably good over the T rounds. However, when the underlying distribution of loss functions changes, the static regret may be too optimistic and fails to capture the hardness of the problem.

To address this limitation, new forms of performance measure, including *adaptive* regret (Hazan & Seshadhri, 2007; 2009) and *dynamic* regret (Zinkevich, 2003; Hall & Willett, 2013), were proposed and received significant interest recently. Following the terminology of Daniely et al. (2015), we define the strongly adaptive regret as the maximum static regret over intervals of length τ, i.e.,

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

Minimizing the adaptive regret enforces the learner to have a small static regret over any interval of length τ. Since the best decision for different intervals could be different, the learner is essentially competing with a changing comparator.

A parallel line of research introduces the concept of dynamic regret, where the cumulative loss of the learner is compared against a comparator sequence $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \Omega$, i.e.,

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

It is well-known that in the worst case, a sublinear dynamic regret is impossible unless we impose some regularities on the comparator sequence or the function sequence (Jadbabaie et al., 2015). A representative example is the functional variation defined below

$$V_T = \sum_{t=2}^{T} \max_{\mathbf{w}\in\Omega} |f_t(\mathbf{w}) - f_{t-1}(\mathbf{w})|. \tag{3}$$

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China [2]Department of Computer Science, The University of Iowa, Iowa City, USA [3]Alibaba Group, Seattle, USA. Correspondence to: Lijun Zhang <zhanglj@lamda.nju.edu.cn>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

Besbes et al. (2015) have proved that as long as V_T is sublinear in T, there exists an algorithm that achieves a sublinear dynamic regret. Furthermore, a general restarting procedure is developed, and it enjoys $O(T^{2/3}V_T^{1/3})$ and $O(\log T\sqrt{TV_T})$ rates for convex functions and strongly convex functions, respectively. However, the restarting procedure can only be applied when an upper bound of V_T is known beforehand, thus limiting its application in practice.

While both the adaptive and dynamic regrets aim at coping with changing environments, little is known about their relationship. This paper makes a step towards understanding their connections. Specifically, we show that the strongly adaptive regret in (1), together with the functional variation, can be used to upper bound the dynamic regret in (2). Thus, an algorithm with a small strongly adaptive regret is automatically equipped with a tight dynamic regret. As a result, we obtain a series of algorithms for minimizing the dynamic regret that do not need any prior knowledge of the functional variation. The main contributions of this work are summarized below.

- We provide a general theorem that upper bounds the dynamic regret in terms of the strongly adaptive regret and the functional variation.
- For convex functions, we show that the strongly adaptive algorithm of Jun et al. (2017) has a dynamic regret of $O(T^{2/3}V_T^{1/3}\log^{1/3} T)$, which matches the minimax rate (Besbes et al., 2015), up to a polylogarithmic factor.
- For exponentially concave functions, we propose a strongly adaptive algorithm that allows us to control the tradeoff between the adaptive regret and the computational cost explicitly. Then, we demonstrate that its dynamic regret is $O(d\sqrt{TV_T\log T})$, where d is the dimensionality. To the best of our knowledge, this is the *first* time that exponential concavity is utilized in the analysis of dynamic regret.
- For strongly convex functions, our proposed algorithm can also be applied and yields a dynamic regret of $O(\sqrt{TV_T\log T})$, which is also minimax optimal up to a polylogarithmic factor.

2. Related Work

We give a brief introduction to previous work on static, adaptive, and dynamic regrets in the context of online convex optimization.

2.1. Static Regret

The majority of studies in online learning are focused on static regret (Shalev-Shwartz & Singer, 2007; Langford et al., 2009; Shalev-Shwartz, 2011; Zhang et al., 2013). For general convex functions, the classical online gradient descent achieves $O(\sqrt{T})$ and $O(\log T)$ regret bounds for convex and strongly convex functions, respectively (Zinkevich, 2003; Hazan et al., 2007; Shalev-Shwartz et al., 2007). Both the $O(\sqrt{T})$ and $O(\log T)$ rates are known to be minimax optimal (Abernethy et al., 2009). When functions are exponentially concave, a different algorithm, named online Newton step, is developed and enjoys an $O(d\log T)$ regret bound, where d is the dimensionality (Hazan et al., 2007).

2.2. Adaptive Regret

The concept of adaptive regret is introduced by Hazan & Seshadhri (2007), and later strengthened by Daniely et al. (2015). Specifically, Hazan & Seshadhri (2007) introduce the weakly adaptive regret

$$\begin{aligned}&\text{WA-Regret}(T)\\&=\max_{[s,q]\subseteq[T]}\left(\sum_{t=s}^{q} f_t(\mathbf{w}_t)-\min_{\mathbf{w}\in\Omega}\sum_{t=s}^{q} f_t(\mathbf{w})\right).\end{aligned}$$

To minimize the adaptive regret, Hazan & Seshadhri (2007) have developed two meta-algorithms: an efficient algorithm with $O(\log T)$ computational complexity per iteration and an inefficient one with $O(T)$ computational complexity per iteration. These meta-algorithms use an existing online method (that was possibly designed to have small static regret) as a subroutine.[1] For convex functions, the efficient and inefficient meta-algorithms have $O(\sqrt{T\log^3 T})$ and $O(\sqrt{T\log T})$ regret bounds, respectively. For exponentially concave functions, those rates are improved to $O(d\log^2 T)$ and $O(d\log T)$, respectively. We can see that the price paid for the adaptivity is very small: The rates of weakly adaptive regret differ from those of static regret only by logarithmic factors.

A major limitation of weakly adaptive regret is that it does not respect short intervals well. Taking convex functions as an example, the $O(\sqrt{T\log^3 T})$ and $O(\sqrt{T\log T})$ bounds are meaningless for intervals of length $O(\sqrt{T})$. To overcome this limitation, Daniely et al. (2015) proposed the strongly adaptive regret $\text{SA-Regret}(T,\tau)$ which takes the length of the interval τ as a parameter, as indicated in (1). From the definitions, we have $\text{SA-Regret}(T,\tau)\leq \text{WA-Regret}(T)$, but it does not mean the notation of weakly adaptive regret is stronger, because an upper bound for $\text{WA-Regret}(T)$ could be very loose for $\text{SA-Regret}(T,\tau)$ when τ is small.

If the strongly adaptive regret is small for all $\tau < T$, we can guarantee the learner has a small regret over any interval of

[1] For brevity, we ignored the factor of subroutine in the statements of computational complexities. The $O(\cdot)$ computational complexity should be interpreted as $O(\cdot)\times s$ space complexity and $O(\cdot)\times t$ time complexity, where s and t are space and time complexities of the subroutine per iteration, respectively.

any length. In particular, Daniely et al. (2015) introduced the following definition.

Definition 1 *Let $R(\tau)$ be the minimax static regret bound of the learning problem over τ periods. An algorithm is* strongly adaptive*, if*

$$\text{SA-Regret}(T, \tau) = O(\text{poly}(\log T) \cdot R(\tau)), \ \forall \tau.$$

It is easy to verify that the meta-algorithms of Hazan & Seshadhri (2007) are strongly adaptive for exponentially concave functions,[2] but not for convex functions. Thus, Daniely et al. (2015) developed a new meta-algorithm that satisfies $\text{SA-Regret}(T, \tau) = O(\sqrt{\tau} \log T)$ for convex functions, and thus is strongly adaptive. The algorithm is also efficient and the computational complexity per iteration is $O(\log T)$. Later, the strongly adaptive regret of convex functions was improved to $O(\sqrt{\tau \log T})$ by Jun et al. (2017), and the computational complexity remains $O(\log T)$ per iteration. All the previously mentioned algorithms for minimizing adaptive regret need to query the gradient of the loss function at least $O(\log t)$ times in the t-th iteration. In a recent study, Wang et al. (2018) demonstrate that the number of gradient evaluations per iteration can be reduced to 1 by introducing the surrogate loss.

2.3. Dynamic Regret

In a seminal work, Zinkevich (2003) proposed to use the *path-length* defined as

$$\mathcal{P}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_2$$

to upper bound the dynamic regret, where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \Omega$ is a comparator sequence. Specifically, Zinkevich (2003) proved that for any sequence of convex functions, the dynamic regret of online gradient descent can be upper bounded by $O(\sqrt{T}\mathcal{P}(\mathbf{u}_1, \ldots, \mathbf{u}_T))$. Another regularity of the comparator sequence, which is similar to the path-length, is defined as

$$\mathcal{P}'(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=2}^{T} \|\mathbf{u}_t - \Phi_t(\mathbf{u}_{t-1})\|_2$$

where $\Phi_t(\cdot)$ is a dynamic model that predicts a reference point for the t-th round. Hall & Willett (2013) developed a novel algorithm named dynamic mirror descent and proved that its dynamic regret is on the order of $\sqrt{T}\mathcal{P}'(\mathbf{u}_1, \ldots, \mathbf{u}_T)$. The advantage of $\mathcal{P}'(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ is that when the comparator sequence follows the dynamical model closely, it can be much smaller than the path-length $\mathcal{P}(\mathbf{u}_1, \ldots, \mathbf{u}_T)$.

Let $\mathbf{w}_t^* \in \text{argmin}_{\mathbf{w} \in \Omega} f_t(\mathbf{w})$ be a minimizer of $f_t(\cdot)$. For any sequence of $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \Omega$, we have

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

Thus, $\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ can be treated as the worst case of the dynamic regret, and there are many works that were devoted to minimizing $\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ (Jadbabaie et al., 2015; Mokhtari et al., 2016; Yang et al., 2016; Zhang et al., 2017).

When a prior knowledge of $\mathcal{P}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ is available, $\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ can be upper bounded by $O(\sqrt{T\mathcal{P}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)})$ (Yang et al., 2016). If all the functions are strongly convex and smooth, the upper bound can be improved to $O(\mathcal{P}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*))$ (Mokhtari et al., 2016). The $O(\mathcal{P}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*))$ rate is also achievable when all the functions are convex and smooth, and all the minimizers $\mathbf{w}_t^*$'s lie in the interior of Ω (Yang et al., 2016). In a recent study, Zhang et al. (2017) introduced a new regularity—*squared* path-length

$$\mathcal{S}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) = \sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2^2$$

which could be much smaller than the path-length $\mathcal{P}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ when the difference between successive minimizers is small. Zhang et al. (2017) developed a novel algorithm named online multiple gradient descent, and proved that $\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*)$ is on the order of $\min(\mathcal{P}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*), \mathcal{S}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*))$ for (semi-) strongly convex and smooth functions.

Discussions Although closely related, adaptive regret and dynamic regret are studied independently and there are few discussions of their relationships. In the literature, dynamic regret is also referred to as tracking regret or shifting regret (Littlestone & Warmuth, 1994; Herbster & Warmuth, 1998; 2001). In the setting of "prediction with expert advice", Adamskiy et al. (2012) have shown that the tracking regret can be derived from the adaptive regret. In the setting of "online linear optimization in the simplex", Cesa-bianchi et al. (2012) introduced a generalized notion of shifting regret which unifies adaptive regret and shifting regret. Different from previous work, this paper considers the setting of online convex optimization, and illustrates that the dynamic regret can be upper bounded by the adaptive regret and the functional variation.

[2] That is because (i) $\text{SA-Regret}(T, \tau) \leq \text{WA-Regret}(T)$, and (ii) there is a $\text{poly}(\log T)$ factor in the definition of strong adaptivity.

3. A Unified Adaptive Algorithm

In this section, we introduce a unified approach for minimizing the adaptive regret of exponentially concave functions, as well as strongly convex functions.

3.1. Motivation

We first provide the definition of exponentially concave (abbr. exp-concave) functions (Cesa-Bianchi & Lugosi, 2006).

Definition 2 *A function $f(\cdot) : \Omega \mapsto \mathbb{R}$ is α-exp-concave if $\exp(-\alpha f(\cdot))$ is concave over Ω.*

For exp-concave functions, Hazan & Seshadhri (2007) have developed two meta-algorithms that take the online Newton step as its subroutine, and proved the following properties.

- The inefficient one has $O(T)$ computational complexity per iteration, and its adaptive regret is $O(d \log T)$.
- The efficient one has $O(\log T)$ computational complexity per iteration, and its adaptive regret is $O(d \log^2 T)$.

As can be seen, there is a tradeoff between the computational complexity and the adaptive regret: A lighter computation incurs a looser bound and a tighter bound requires a higher computation. Our goal is to develop a unified approach, that allows us to trade effectiveness for efficiency explicitly.

3.2. Improved Following the Leading History (IFLH)

Let E be an online learning algorithm that is designed to minimize the static regret of exp-concave functions or strongly convex functions, e.g., online Newton step (Hazan et al., 2007) or online gradient descent (Zinkevich, 2003). Similar to the approach of following the leading history (FLH) (Hazan & Seshadhri, 2007), at any time t, we will instantiate an expert by applying the online learning algorithm E to the sequence of loss functions $f_t, f_{t+1}, \ldots$, and utilize the strategy of learning from expert advice to combine solutions of different experts (Herbster & Warmuth, 1998). Our method is named as improved following the leading history (IFLH), and is summarized in Algorithm 1.

Algorithm 1 Improved Following the Leading History (IFLH)

1: **Input:** An integer K
2: Initialize $\mathcal{S}_0 = \emptyset$.
3: **for** $t = 1, \ldots, T$ **do**
4: Set $Z_t = 0$
{Remove some existing experts}
5: **for** $E^j \in \mathcal{S}_{t-1}$ **do**
6: **if** $e^j \leq t$ **then**
7: Update $\mathcal{S}_{t-1} \leftarrow \mathcal{S}_{t-1} \setminus \{E^j\}$
8: **else**
9: Set $Z_t = Z_t + \widehat{p}_t^j$
10: **end if**
11: **end for**
{Normalize the probability}
12: **for** $E^j \in \mathcal{S}_{t-1}$ **do**
13: Set $p_t^j = \frac{\widehat{p}_t^j}{Z_t}\left(1 - \frac{1}{t}\right)$
14: **end for**
{Add a new expert E^t}
15: Set $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{E^t\}$
16: Compute the ending time $e^t = \mathrm{E}_K(t)$ according to Definition 3 and set $p_t^t = \frac{1}{t}$
{Compute the final predicted model}
17: Submit the solution

$$\mathbf{w}_t = \sum_{E^j \in \mathcal{S}_t} p_t^j \mathbf{w}_t^j$$

and suffer loss $f_t(\mathbf{w}_t)$
{Update weights and expert}
18: Set $Z_{t+1} = 0$
19: **for** $E^j \in \mathcal{S}_t$ **do**
20: Compute $p_{t+1}^j = p_t^j \exp(-\alpha f_t(\mathbf{w}_t^j))$ and $Z_{t+1} = Z_{t+1} + p_{t+1}^j$
21: Pass the function $f_t(\cdot)$ to E^j
22: **end for**
23: **for** $E^j \in \mathcal{S}_t$ **do**
24: Set $\widehat{p}_{t+1}^j = \frac{p_{t+1}^j}{Z_{t+1}}$
25: **end for**
26: **end for**

Let E^t be the expert that starts to work at time t. To control the computational complexity, we will associate an ending time e^t for each E^t. The expert E^t is alive during the period $[t, e^t - 1]$. In each round t, we maintain a working set of experts $\mathcal{S}_t$, which contains all the alive experts, and assign a probability p_t^j for each $E^j \in S_t$. In Steps 6 and 7, we remove all the experts whose ending times are no larger than t. Since the number of alive experts has changed, we need to update the probability assigned to them, which is performed in Steps 12 to 14. In Steps 15 and 16, we add a new expert E^t to $\mathcal{S}_t$, calculate its ending time according to Definition 3 introduced below, and set $p_t^t = \frac{1}{t}$. It is easy to verify $\sum_{E^j \in \mathcal{S}_t} p_t^j = 1$. Let $\mathbf{w}_t^j$ be the output of E^j at the t-th round, where $t \geq j$. In Step 17, we submit the weighted average of $\mathbf{w}_t^j$ with coefficient p_t^j as the output $\mathbf{w}_t$, and suffer the loss $f_t(\mathbf{w}_t)$. From Steps 18 to 25, we use the exponential weighting scheme to update the weight for each expert E^j based on its loss $f_t(\mathbf{w}_t^j)$. In Step 21, we pass the loss function to all the alive experts such that they can update their predictions for the next round.

The difference between our IFLH and the original FLH is how to decide the ending time e^t of expert E^t. In this paper, we propose the following base-K ending time.

Definition 3 (Base-K Ending Time) *Let K be an integer, and the representation of t in the base-K number system as*

$$t = \sum_{\tau \geq 0} \beta_\tau K^\tau$$

where $0 \leq \beta_\tau < K$, for all $\tau \geq 0$. Let k be the smallest integer such that $\beta_k > 0$, i.e., $k = \min\{\tau : \beta_\tau > 0\}$. Then, the base-$K$ ending time of t is defined as

$$\mathrm{E}_K(t) = \sum_{\tau \geq k+1} \beta_\tau K^\tau + K^{k+1}.$$

In other words, the ending time is the number represented by the new sequence obtained by setting the first nonzero element in the sequence $\beta_0, \beta_1, \ldots$ to be 0 and adding 1 to the element after it.

Let's take the decimal system as an example (i.e., $K = 10$). Then,

$$\begin{aligned} E_{10}(1) &= E_{10}(2) = \cdots = E_{10}(9) = 10, \\ E_{10}(11) &= E_{10}(12) = \cdots = E_{10}(19) = 20, \\ E_{10}(10) &= E_{10}(20) = \cdots = E_{10}(90) = 100. \end{aligned}$$

3.3. Theoretical Guarantees

When the base-K ending time is used in Algorithm 1, we have the following properties.

Lemma 1 *Suppose we use the base-K ending time in Algorithm 1.*

1. *For any $t \geq 1$, we have*

$$|\mathcal{S}_t| \leq (\lfloor \log_K t \rfloor + 1)(K-1) = O\left(\frac{K \log t}{\log K}\right).$$

2. *For any interval $I = [r, s] \subseteq [T]$, we can always find m segments $I_j = [t_j, e^{t_j} - 1]$, $j \in [m]$ with $m \leq \lceil \log_K (s - r + 1) \rceil + 1$, such that $t_1 = r$, $e^{t_j} = t_{j+1}$, $j \in [m-1]$, and $e^{t_m} > s$.*

The first part of Lemma 1 implies that the size of $\mathcal{S}_t$ is $O(K \log t / \log K)$. An example of $\mathcal{S}_t$ in the decimal system is given below.

$$\mathcal{S}_{486} = \begin{Bmatrix} 481, & 482, & \ldots, & 486, \\ 410, & 420, & \ldots, & 480, \\ 100, & 200, & \ldots, & 400 \end{Bmatrix}.$$

The second part of Lemma 1 implies that for any interval $I = [r, s]$, we can find $O(\log s / \log K)$ experts such that their survival periods cover I. Again, we present an example in the decimal system: The interval $[111, 832]$ can be covered by

$$[111, 119], \ [120, 199], \ \text{and} \ [200, 999]$$

which are the survival periods of experts E^{111}, E^{120}, and E^{200}, respectively. Recall that $E_{10}(111) = 120$, $E_{10}(120) = 200$, and $E_{10}(200) = 1000$.

We note that a similar strategy for deciding the ending time was proposed by György et al. (2012) in the study of "prediction with expert advice". The main difference is that their strategy is built upon base-2 number system and introduces an additional parameter g to compromise between the computational complexity and the regret, in contrast our method relies on base-K number system and uses K to control the tradeoff. Lemma 2 of György et al. (2012) indicates an $O(g \log t)$ bound on the number of alive experts, which is worse than our $O(K \log t / \log K)$ bound by a logarithmic factor.

To present adaptive regret bounds, we introduce the following common assumption.

Assumption 1 *Both the gradient and the domain are bounded.*

- *The gradients of all the online functions are bounded by G, i.e., $\max_{\mathbf{w} \in \Omega} \|\nabla f_t(\mathbf{w})\| \leq G$ for all f_t.*
- *The diameter of the domain Ω is bounded by B, i.e., $\max_{\mathbf{w}, \mathbf{w}' \in \Omega} \|\mathbf{w} - \mathbf{w}'\| \leq B$.*

Based on Lemma 1, we have the following theorem regarding the adaptive regret of exp-concave functions.

Theorem 1 *Suppose Assumption 1 holds, $\Omega \subset \mathbb{R}^d$, and all the functions are α-exp-concave. If online Newton step is used as the subroutine in Algorithm 1, we have*

$$\begin{aligned} &\sum_{t=r}^{s} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \Omega} \sum_{t=r}^{s} f_t(\mathbf{w}) \\ \leq &\left(\frac{(5d+1)m + 2}{\alpha} + 5dmGB\right) \log T \end{aligned}$$

where $[r, s] \subseteq [T]$ and $m \leq \lceil \log_K (s - r + 1) \rceil + 1$. Thus,

$$\begin{aligned} &\text{SA-Regret}(T, \tau) \\ \leq &\left(\frac{(5d+1)\bar{m} + 2}{\alpha} + 5d\bar{m}GB\right) \log T = O\left(\frac{d \log^2 T}{\log K}\right) \end{aligned}$$

where $\bar{m} = \lceil \log_K \tau \rceil + 1$.

From Lemma 1 and Theorem 1, we observe that the adaptive regret is a decreasing function of K, while the computational cost is an increasing function of K. Thus, we can control the tradeoff by tuning the value of K. Specifically, Lemma 1 indicates the proposed algorithm has

$$(\lfloor \log_K T \rfloor + 1)(K - 1) = O\left(\frac{K \log T}{\log K}\right)$$

computational complexity per iteration. On the other hand, Theorem 1 implies that for α-exp-concave functions that

Table 1. Efficiency and Effectiveness Tradeoff

K	Complexity	Adaptive Regret
2	$O(\log T)$	$O(d\log^2 T)$
$\lceil T^{1/\gamma}\rceil$	$O(\gamma T^{1/\gamma})$	$O(\gamma d \log T)$
T	$O(T)$	$O(d\log T)$

satisfy Assumption 1, the strongly adaptive regret of Algorithm 1 is

$$\left(\frac{(5d+1)\bar{m}+2}{\alpha}+5d\bar{m}GB\right)\log T = O\left(\frac{d\log^2 T}{\log K}\right)$$

where d is the dimensionality and $\bar{m} = \lceil \log_K(\tau)\rceil + 1$.

We list several choices of K and the resulting theoretical guarantees in Table 1, and have the following observations.

- When $K = 2$, we recover the guarantee of the efficient algorithm of Hazan & Seshadhri (2007), and when $K = T$, we obtain the inefficient one.
- By setting $K = \lceil T^{1/\gamma}\rceil$ where $\gamma > 1$ is a small constant, such as 10, the strongly adaptive regret can be viewed as $O(d\log T)$, and at the same time, the computational complexity is also very low for a large range of T.

Next, we consider strongly convex functions.

Definition 4 *A function $f(\cdot) : \Omega \mapsto \mathbb{R}$ is λ-strongly convex if*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\lambda}{2}\|\mathbf{y}-\mathbf{x}\|_2^2,\ \forall \mathbf{x},\mathbf{y}\in\Omega.$$

It is easy to verify that strongly convex functions with bounded gradients are also exp-concave (Hazan et al., 2007).

Lemma 2 *Suppose $f(\cdot) : \Omega \mapsto \mathbb{R}$ is λ-strongly convex and $\|\nabla f(\mathbf{w})\| \leq G$ for all $\mathbf{w}\in\Omega$. Then, $f(\cdot)$ is $\frac{\lambda}{G^2}$-exp-concave.*

According to the above lemma, we still use Algorithm 1 as the meta-algorithm, but choose online gradient descent as the subroutine. In this way, the adaptive regret does not depend on the dimensionality d.

Theorem 2 *Suppose Assumption 1 holds, and all the functions are λ-strongly convex. If online gradient descent is used as the subroutine in Algorithm 1, we have*

$$\sum_{t=r}^{s} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\Omega}\sum_{t=r}^{s} f_t(\mathbf{w}) \leq \frac{G^2}{2\lambda}\big(m + (3m+4)\log T\big)$$

where $[r,s]\subseteq[T]$ and $m \leq \lceil \log_K(s-r+1)\rceil + 1$. Thus

$$\begin{aligned}&\text{SA-Regret}(T,\tau)\\ \leq&\frac{G^2}{2\lambda}\big(\bar{m}+(3\bar{m}+4)\log T\big) = O\left(\frac{\log^2 T}{\log K}\right)\end{aligned}$$

where $\bar{m} = \lceil \log_K \tau\rceil + 1$.

4. From Adaptive to Dynamic

In this section, we first introduce a general theorem that bounds the dynamic regret by the adaptive regret, and then derive specific regret bounds for convex functions, exponentially concave functions, and strongly convex functions.

4.1. Adaptive-to-Dynamic Conversion

Let $\mathcal{I}_1 = [s_1,q_1], \mathcal{I}_2 = [s_2,q_2], \ldots, \mathcal{I}_k = [s_k,q_k]$ be a partition of $[1,T]$. That is, they are successive intervals such that

$$s_1 = 1,\ q_i + 1 = s_{i+1},\ i\in[k-1], \text{ and } q_k = T. \quad (4)$$

Define the local functional variation of the i-th interval as

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

and it is obvious that $\sum_{i=1}^k V_T(i) \leq V_T$.[3] Then, we have the following theorem for bounding the dynamic regret in terms of the strongly adaptive regret and the functional variation.

Theorem 3 *Let $\mathbf{w}_t^* \in \operatorname{argmin}_{\mathbf{w}\in\Omega} f_t(\mathbf{w})$. For all integer $k\in[T]$, we have*

$$\begin{aligned}&\text{D-Regret}(\mathbf{w}_1^*,\ldots,\mathbf{w}_T^*)\\ \leq&\min_{\mathcal{I}_1,\ldots,\mathcal{I}_k}\sum_{i=1}^{k}\big(\text{SA-Regret}(T,|\mathcal{I}_i|) + 2|\mathcal{I}_i|\cdot V_T(i)\big)\end{aligned}$$

where the minimization is taken over any sequence of intervals that satisfy (4).

The above theorem is analogous to Proposition 2 of Besbes et al. (2015), which provides an upper bound for a special choice of the interval sequence. The main difference is that there is a minimization operation in our bound, which allows us to get rid of the issue of parameter selection. For a specific type of problems, we can plug in the corresponding

[3] Note that in certain cases, the sum of local functional variation $\sum_{i=1}^k V_T(i)$ can be much smaller than the total functional variation V_T. For example, when the sequence of functions only changes k times, we can construct the intervals based on the changing rounds such that $\sum_{i=1}^k V_T(i) = 0$.

upper bound of strongly adaptive regret, and then choose any sequence of intervals to obtain a concrete upper bound. In particular, the choice of the intervals may depend on the (possibly unknown) functional variation.

4.2. Convex Functions

For convex functions, we choose the meta-algorithm of Jun et al. (2017) and take the online gradient descent as its subroutine. The following theorem regarding the adaptive regret can be obtained from that paper.

Theorem 4 *Under Assumption 1, the meta-algorithm of Jun et al. (2017) is strongly adaptive with*

$$\begin{aligned}&\text{SA-Regret}(T,\tau)\\ \leq&\left(\frac{12BG}{\sqrt{2}-1}+8\sqrt{7\log T+5}\right)\sqrt{\tau}=O(\sqrt{\tau\log T}).\end{aligned}$$

From Theorems 3 and 4, we derive the following bound for the dynamic regret.

Corollary 5 *Under Assumption 1, the meta-algorithm of Jun et al. (2017) satisfies*

$$\begin{aligned}&\text{D-Regret}(\mathbf{w}_1^*,\ldots,\mathbf{w}_T^*)\\ \leq&\max\begin{cases}(c+9\sqrt{7\log T+5})\sqrt{T}\\ \dfrac{(c+8\sqrt{5})T^{2/3}V_T^{1/3}}{\log^{1/6}T}+24T^{2/3}V_T^{1/3}\log^{1/3}T\end{cases}\\ =&O\left(\max\left\{\sqrt{T\log T},T^{2/3}V_T^{1/3}\log^{1/3}T\right\}\right)\end{aligned}$$

where $c=12BG/(\sqrt{2}-1)$.

According to Theorem 2 of Besbes et al. (2015), we know that the minimax dynamic regret of convex functions is $O(T^{2/3}V_T^{1/3})$. Thus, our upper bound is minimax optimal up to a polylogarithmic factor. Although the restarted online gradient descent of Besbes et al. (2015) achieves a dynamic regret of $O(T^{2/3}V_T^{1/3})$, it requires to know an upper bound of the functional variation V_T. In contrast, the meta-algorithm of Jun et al. (2017) does not need any prior knowledge of V_T. We note that the meta-algorithm of Daniely et al. (2015) can also be used here, and its dynamic regret is on the order of $\max\left\{\sqrt{T}\log T,T^{2/3}V_T^{1/3}\log^{2/3}T\right\}$.

4.3. Exponentially Concave Functions

We proceed to consider exp-concave functions, defined in Definition 2. Exponential concavity is stronger than convexity but weaker than strong convexity. It can be used to model many popular losses used in machine learning, such as the square loss in regression, logistic loss in classification and negative logarithm loss in portfolio management (Koren, 2013).

For exp-concave functions, we choose Algorithm 1 in this paper, and take the online Newton step as its subroutine. Based on Theorems 1 and 3, we derive the dynamic regret of the proposed algorithm.

Corollary 6 *Let $K=\lceil T^{1/\gamma}\rceil$, where $\gamma>1$ is a small constant. Suppose Assumption 1 holds, $\Omega\subset\mathbb{R}^d$, and all the functions are α-exp-concave. Algorithm 1, with online Newton step as its subroutine, is strongly adaptive with*

$$\begin{aligned}&\text{SA-Regret}(T,\tau)\\ \leq&\left(\frac{(5d+1)(\gamma+1)+2}{\alpha}+5d(\gamma+1)GB\right)\log T\\ =&O\left(\gamma d\log T\right)=O\left(d\log T\right)\end{aligned}$$

and its dynamic regret satisfies

$$\begin{aligned}&\text{D-Regret}(\mathbf{w}_1^*,\ldots,\mathbf{w}_T^*)\\ \leq&\left(\frac{(5d+1)(\gamma+1)+2}{\alpha}+5d(\gamma+1)GB+2\right)\\ &\cdot\max\left\{\log T,\sqrt{TV_T\log T}\right\}\\ =&O\left(d\cdot\max\left\{\log T,\sqrt{TV_T\log T}\right\}\right).\end{aligned}$$

To the best of our knowledge, this is the *first* dynamic regret that exploits exponential concavity. Furthermore, according to the minimax dynamic regret of strongly convex functions (Besbes et al., 2015), our upper bound is minimax optimal, up to a polylogarithmic factor.

4.4. Strongly Convex Functions

Finally, we study strongly convex functions. According to Lemma 2, we know that strongly convex functions with bounded gradients are also exp-concave. Thus, Corollary 6 can be directly applied to strongly convex functions, and yields a dynamic regret of $O(d\sqrt{TV_T\log T})$. However, the upper bound depends on the dimensionality d. To address this limitation, we use online gradient descent as the subroutine in Algorithm 1.

From Theorems 2 and 3, we have the following theorem, in which both the adaptive and dynamic regrets are independent from d.

Corollary 7 *Let $K=\lceil T^{1/\gamma}\rceil$, where $\gamma>1$ is a small constant. Suppose Assumption 1 holds, and all the functions are λ-strongly convex. Algorithm 1, with online gradient descent as its subroutine, is strongly adaptive with*

$$\begin{aligned}&\text{SA-Regret}(T,\tau)\\ \leq&\frac{G^2}{2\lambda}\left(\gamma+1+(3\gamma+7)\log T\right)\\ =&O\left(\gamma\log T\right)=O\left(\log T\right)\end{aligned}$$

and its dynamic regret satisfies

$$\begin{aligned}
&\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) \\
\leq &\max \begin{cases} \frac{\gamma G^2}{\lambda} + \left(\frac{5\gamma G^2}{\lambda} + 2 \right) \log T \\ \frac{\gamma G^2}{\lambda} \sqrt{\frac{TV_T}{\log T}} + \left(\frac{5\gamma G^2}{\lambda} + 2 \right) \sqrt{TV_T \log T} \end{cases} \\
= &O \left(\max \left\{ \log T, \sqrt{TV_T \log T} \right\} \right).
\end{aligned}$$

According to Theorem 4 of Besbes et al. (2015), the minimax dynamic regret of strongly convex functions is $O(\sqrt{TV_T})$, which implies our upper bound is almost minimax optimal. By comparison, the restarted online gradient descent of Besbes et al. (2015) has a dynamic regret of $O(\log T \sqrt{TV_T})$, but it requires to know an upper bound of V_T.

5. Analysis

We here present the proof of Theorem 3. The omitted proofs are provided in the supplementary.

5.1. Proof of Theorem 3

First, we upper bound the dynamic regret in the following way

$$\begin{aligned}
&\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) \\
= &\sum_{i=1}^{k} \left(\sum_{t=s_i}^{q_i} f_t(\mathbf{w}_t) - \sum_{t=s_i}^{q_i} \min_{\mathbf{w} \in \Omega} f_t(\mathbf{w}) \right) \\
= &\sum_{i=1}^{k} \Bigg(\underbrace{\sum_{t=s_i}^{q_i} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \Omega} \sum_{t=s_i}^{q_i} f_t(\mathbf{w})}_{:=a_i} \\
&+ \underbrace{\min_{\mathbf{w} \in \Omega} \sum_{t=s_i}^{q_i} f_t(\mathbf{w}) - \sum_{t=s_i}^{q_i} \min_{\mathbf{w} \in \Omega} f_t(\mathbf{w})}_{:=b_i} \Bigg).
\end{aligned} \tag{5}$$

From the definition of strongly adaptive regret, we can upper bound a_i by

$$\sum_{t=s_i}^{q_i} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \Omega} \sum_{t=s_i}^{q_i} f_t(\mathbf{w}) \leq \text{SA-Regret}(T, |\mathcal{I}_i|).$$

To upper bound b_i, we follow the analysis of Proposition 2 of Besbes et al. (2015):

$$\begin{aligned}
&\min_{\mathbf{w} \in \Omega} \sum_{t=s_i}^{q_i} f_t(\mathbf{w}) - \sum_{t=s_i}^{q_i} \min_{\mathbf{w} \in \Omega} f_t(\mathbf{w}) \\
= &\min_{\mathbf{w} \in \Omega} \sum_{t=s_i}^{q_i} f_t(\mathbf{w}) - \sum_{t=s_i}^{q_i} f_t(\mathbf{w}_t^*) \\
\leq &\sum_{t=s_i}^{q_i} f_t(\mathbf{w}_{s_i}^*) - \sum_{t=s_i}^{q_i} f_t(\mathbf{w}_t^*) \\
\leq &|\mathcal{I}_i| \cdot \max_{t \in [s_i, q_i]} \left(f_t(\mathbf{w}_{s_i}^*) - f_t(\mathbf{w}_t^*) \right).
\end{aligned} \tag{6}$$

Furthermore, for any $t \in [s_i, q_i]$, we have

$$\begin{aligned}
&f_t(\mathbf{w}_{s_i}^*) - f_t(\mathbf{w}_t^*) \\
= &f_t(\mathbf{w}_{s_i}^*) - f_{s_i}(\mathbf{w}_{s_i}^*) + f_{s_i}(\mathbf{w}_{s_i}^*) - f_t(\mathbf{w}_t^*) \\
\leq &f_t(\mathbf{w}_{s_i}^*) - f_{s_i}(\mathbf{w}_{s_i}^*) + f_{s_i}(\mathbf{w}_t^*) - f_t(\mathbf{w}_t^*) \\
\leq &2V_T(i).
\end{aligned} \tag{7}$$

Combining (6) with (7), we have

$$\min_{\mathbf{w} \in \Omega} \sum_{t=s_i}^{q_i} f_t(\mathbf{w}) - \sum_{t=s_i}^{q_i} \min_{\mathbf{w} \in \Omega} f_t(\mathbf{w}) \leq 2|\mathcal{I}_i| \cdot V_T(i).$$

Substituting the upper bounds of a_i and b_i into (5), we arrive at

$$\begin{aligned}
&\text{D-Regret}(\mathbf{w}_1^*, \ldots, \mathbf{w}_T^*) \\
\leq &\sum_{i=1}^{k} \left(\text{SA-Regret}(T, |\mathcal{I}_i|) + 2|\mathcal{I}_i| \cdot V_T(i) \right).
\end{aligned}$$

Since the above inequality holds for any partition of $[1, T]$, we can take minimization to get a tight bound.

6. Conclusions and Future Work

In this paper, we demonstrate that the dynamic regret can be upper bounded by the adaptive regret and the functional variation, which implies strongly adaptive algorithms are automatically equipped with tight dynamic regret bounds. As a result, we are able to derive dynamic regret bounds for convex functions, exp-concave functions, and strongly convex functions. Moreover, we provide a unified approach for minimizing the adaptive regret of exp-concave functions, as well as strongly convex functions.

The adaptive-to-dynamic conversion leads to a series of dynamic regret bounds in terms of the functional variation. As we mentioned before, dynamic regret can also be upper bounded by other regularities such as the path-length. It is interesting to investigate whether those kinds of upper bounds can also be established for strongly adaptive algorithms.

Acknowledgements

This work was partially supported by the National Key R&D Program of China (2018YFB1004300), NSFC (61603177, 61333014), JiangsuSF (BK20160658), YESS (2017QNRC001), NSF (IIS-1545995), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Abernethy, J., Agarwal, A., Bartlett, P. L., and Rakhlin, A. A stochastic view of optimal regret through minimax duality. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

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

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

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

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

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.

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

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

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

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

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

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

Herbster, M. and Warmuth, M. K. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1: 281–309, 2001.

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

Jun, K.-S., Orabona, F., Wright, S., and Willett, R. Improved Strongly Adaptive Online Learning using Coin Betting. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 943–951, 2017.

Koren, T. Open problem: Fast stochastic exp-concave optimization. In *Proceedings of the 26th Annual Conference on Learning Theory*, pp. 1073–1075, 2013.

Langford, J., Li, L., and Zhang, T. Sparse online learning via truncated gradient. In *Advances in Neural Information Processing Systems 21*, pp. 905–912, 2009.

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

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

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

Shalev-Shwartz, S. and Singer, Y. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2): 115–142, 2007.

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.

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

Yang, T., Zhang, L., Jin, R., and Yi, J. 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*, pp. 449–457, 2016.

Zhang, L., Yi, J., Jin, R., Lin, M., and He, X. Online kernel learning with a near optimal sparsity bound. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

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

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