A Simple Online Algorithm for Competing with Dynamic Comparators

Yu-Jie Zhang, Peng Zhao, Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{zhangyj, zhaop, zhouzh}@lamda.nju.edu.cn

Abstract

Online learning in dynamic environments has recently drawn considerable attention, where dynamic regret is usually employed to compare decisions of online algorithms to dynamic comparators. In previous works, dynamic regret bounds are typically established in terms of regularity of comparators C_T or that of online functions V_T. Recently, Jadbabaie et al. [2015] propose an algorithm that can take advantage of both regularities and enjoy an $\widetilde{O}(\sqrt{1+D_T}+\min\{\sqrt{(1+D_T)C_T},(1+D_T)^{1/3}V_T^{1/3}T^{1/3}\})$ dynamic regret, where D_T is an additional quantity to measure the niceness of environments. The regret bound adapts to the smaller regularity of problem environments and is tighter than all existing dynamic regret guarantees. Nevertheless, their algorithm involves non-convex programming at each iteration, and thus requires burdensome computations. In this paper, we design a simple algorithm based on the online ensemble, which provably enjoys the same (even slightly stronger) guarantee as the state-of-the-art rate, yet is much more efficient because our algorithm does not involve any non-convex problem solving. Empirical studies also verify the efficacy and efficiency.

1 INTRODUCTION

Online Convex Optimization (OCO), in which an online learner iteratively makes the decision against the environments, has demonstrated powerful modeling capability in many real-life applications [Shalev-Shwartz, 2012, Hazan, 2016]. The basic protocol of OCO is as follows: at each iteration $t \in \{1,\dots,T\}$, the learner first chooses a decision x_t from some convex feasible set $\mathcal{X} \subseteq \mathbb{R}^d$. Afterwards, a convex function $f_t : \mathcal{X} \mapsto \mathbb{R}$ is revealed by the environments and the learner suffers an instantaneous loss $f_t(x_t)$. The classical measure for an online algorithm is the *regret* [Zinkevich, 2003],

$$\mathbf{Reg}_T^s = \sum_{t=1}^T f_t(x_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^T f_t(x),$$

which compares learner's decisions against a single best decision in hindsight. The measure is often referred to as the *static* regret since the comparator is time-invariant. For convex functions, Zinkevich [2003] shows that online gradient descent enjoys an $O(\sqrt{T})$ static regret, which is further proved to be the minimal regret that can be achieved in the worst case [Abernethy et al., 2008].

Notwithstanding the minimax optimality, the rate can be further improved by exploiting the *niceness* of the environments, which aims to design algorithms that can automatically adapt to easier problem environments, yet safeguard the worst-case rate. There are many ways to exploit the niceness of environments. Rakhlin and Sridharan [2013] develop a general framework to unify previous works by introducing the notion of *predictable sequences* $\{M_t\}_{t=1}^T$, which can be regarded as external knowledge on the gradients of the online functions. They show that an $O(\sqrt{D_T})$ static regret is attainable, where D_T measures the quality of the predictable sequences defined as

$$D_T = \sum_{t=1}^T \|\nabla f_t(x_t) - M_t\|_*^2. \quad (1)$$

The term is at most $O(T)$ under certain standard assumptions, yet it could be quite small when M_t approximates the gradient well. Therefore, the attained regret bound is tighter than the minimax optimal rate of $O(\sqrt{T})$.

Another way to strengthen the theoretical guarantees is to compare the learner's decision with time-varying comparators. This is often desired, particularly for learning in the non-stationary environments [Sugiyama and Kawanabe, 2012, Gama et al., 2014, Zhao et al., 2019], where

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

there is no single fixed decision that performs well over all the iterations. Under such a circumstance, *dynamic regret* is introduced to measure the performance of an online algorithm [Zinkevich, 2003, Besbes et al., 2015, Yang et al., 2016, Zhang et al., 2017, Baby and Wang, 2019, Zhao and Zhang, 2020], defined as

$$\mathbf{Reg}_T^d = \sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x_t^*),$$

where $x_t^* \in \arg\min_{x\in\mathcal{X}} f_t(x)$ is the minimizer of the online function at each iteration. It is known that a sublinear dynamic regret bound is not possible unless imposing certain restrictions on the environments. In the literature, two types of regularities are introduced. The first is the *path-length* [Zinkevich, 2003, Yang et al., 2016, Mokhtari et al., 2016, Zhang et al., 2018],

$$C_T = \sum_{t=2}^T \|x_t^* - x_{t-1}^*\|, \tag{2}$$

measuring the fluctuation of comparators. Another regularity is the *temporal variability* [Besbes et al., 2015, Chen et al., 2019, Baby and Wang, 2019],

$$V_T = \sum_{t=2}^T \sup_{x\in\mathcal{X}} |f_t(x) - f_{t-1}(x)|, \tag{3}$$

which captures the variation of the function values.

Thus, it is natural to ask whether the niceness of environments D_T can also be exploited in the dynamic regret analysis; and meanwhile, as C_T and V_T capture different properties of the environments, is it possible to achieve the best of the two regularities? The problem has been studied by the remarkable work of Jadbabaie et al. [2015]. They affirmatively answer the question by designing an algorithm with the fully adaptive dynamic regret,

$$\mathbf{Reg}_T^d \le \widetilde{O}\big(\sqrt{1+D_T} + \min\{R_C, R_V\}\big), \tag{4}$$

where $\widetilde{O}(\cdot)$ hides logarithmic terms in T, $R_C = \sqrt{(1+D_T)(1+C_T)}$ and $R_V = (1+D_T)^{1/3}V_T^{1/3}T^{1/3}$ are path-length and temporal variability bounds, respectively. The regret bound in (4) is *adaptive* in the sense that it makes use of both regularities C_T, V_T, and adaptivity D_T. Thus, it can adapt to the smaller regularity of problem environments and is tighter than all existing dynamic regret guarantees.

The key challenge of designing an online algorithm with adaptive bound is how to tune the learning parameters interplaying among the complexity measures (D_T, C_T, V_T), *when their exact quantities are unknown in advance*. Jadbabaie et al. [2015] successfully address the issue by carefully designing a novel *doubling trick* scheme. Nevertheless, their method requires the empirical version of temporal variability V_T for checking the doubling condition. Therefore, it will introduce a non-convex inner optimization problem of $\sup_{x\in\mathcal{X}} |f_t(x) - f_{t-1}(x)|$ at each iteration. By exploiting the special propriety of this problem, difference of convex functions programming can be used for solving it, however, is very time-consuming.

In this paper, we discover that a simple and efficient approach provably achieves the same (even slightly stronger) dynamic regret bound (4) as Jadbabaie et al. [2015]. Our contribution is that the proposed algorithm avoids the time-consuming inner optimization problem, and thus is more computationally efficient. The crucial advantage roots in a novel step size tuning approach via the *online ensemble* rather than the doubling trick, where a master algorithm is employed to search over base algorithms with multiple learning parameters. Meanwhile, for exploiting the niceness of environments, we build a new master algorithm based on optimistic Hedge [Rakhlin and Sridharan, 2013] with properly designed optimism, which makes compatible regret bounds for both master and base algorithms, and finally leads to the desired result. Moreover, we provide novel analysis of the temporal variability dynamic regret bound for running online algorithms over the surrogate linearized loss, which could be of interest.

2 RELATED WORK

We review related work about online convex optimization in the following two aspects.

2.1 STATIC REGRET

Significant efforts have been devoted to designing algorithms for static regret minimization. For convex functions, Zinkevich [2003] proves that online gradient descent (OGD) ensures an $O(\sqrt{T})$ static regret, and the rate can be improved to $O(\log T)$ for strongly convex functions [Hazan et al., 2007]. Besides, Hazan et al. [2007] prove that a logarithmic regret $O(d\log T)$ is also achievable for exp-concave functions by the online Newton step algorithm, where d is the dimensionality. All these results are minimax optimal [Abernethy et al., 2008].

One direction to improve the guarantee is to exploit the niceness of environments. Hazan and Kale [2008] establish a bound relative to the variation of the gradient sequences of order $O(\sqrt{\sum_{t=1}^T \|\nabla f_t(x_t) - \bar{\nabla}_T\|_*^2}$ for convex functions, where $\bar{\nabla}_T = \frac{1}{T}\sum_{t=1}^T \nabla f_t(x_t)$. When the online functions are convex and smooth, Chiang et al. [2012] show that the regret bound scales with the variation of consecutive gradients in the rate

Table 1: Comparisons of our results and existing dynamic regret bounds for convex functions.

complexity measure	regret bound	reference
path-length bound	$O(\sqrt{T(1+C_T)})$	[Zinkevich, 2003, Zhang et al., 2018]
temporal variability bound	$O(T^{2/3}V_T^{1/3})$	[Besbes et al., 2015]
adaptive bound	$O(\min\{\sqrt{T(1+C_T)}, T^{2/3}V_T^{1/3}\})$	[Jadbabaie et al., 2015] & this paper
fully adaptive bound	$O(\sqrt{D_T} + \min\{\sqrt{(1+D_T)(1+C_T)}, (1+D_T)^{1/3}T^{1/3}V_T^{1/3}\})$	[Jadbabaie et al., 2015] & this paper

of $O(\sqrt{\sum_{t=1}^T \sup_{x\in\mathcal{X}} \|\nabla f_t(x) - \nabla f_{t-1}(x)\|_*^2})$. Afterwards, Rakhlin and Sridharan [2013] propose a general framework to unify and generalize these results by introducing the notion of predictable sequences $\{M_t\}_{t=1}^T$. They propose the optimistic mirror descent which provably enjoys an $O(\sqrt{D_T})$ static regret, where $D_T = \sum_{t=1}^T \|\nabla f_t(x_t) - M_t\|_*^2$. The framework is very powerful as it can recover previous regret bounds by different configurations of predictable sequences.

2.2 DYNAMIC REGRET

Roughly, there are two types of dynamic regret bounds in terms of different regularities, the path-length C_T, and temporal variability V_T.

Path-length Bound. The path-length is introduced by Zinkevich [2003], measuring the fluctuation of the online minimizers, as defined in (2). When the path-length C_T is known in advance, Zinkevich [2003] shows OGD enjoys an $O(\sqrt{T(1+C_T)})$ dynamic regret for convex functions. This rate can be enhanced to $O(C_T)$ for strongly convex and smooth functions [Mokhtari et al., 2016], and the $O(C_T)$ rate is also attainable for convex and smooth functions if all the minimizers lie in the interior of $\mathcal{X}$ [Yang et al., 2016]. Another similar regularity called squared path-length $S_T = \sum_{t=1}^T \|x_{t-1}^* - x_t^*\|^2$ is introduced by Zhang et al. [2017], and authors propose an algorithm for strongly convex and smooth functions, with $O(\min\{C_T, S_T\})$ dynamic regret, and the regret rate is recently improved to $O(\min\{C_T, S_T, V_T\})$ by Zhao and Zhang [2020]. Since this paper focuses on convex functions, we do not take the squared path-length into account.

Temporal Variability Bound. The temporal variability is introduced by Besbes et al. [2015], capturing the variation of function values, as defined in (3). When V_T is known in advance, Besbes et al. [2015] propose the restarted OGD and derive minimax optimal regret bounds of $O(T^{2/3}V_T^{1/3})$ and $\widetilde{O}(T^{1/2}V_T^{1/2})$ for convex and strongly convex functions respectively. The algorithm and analysis are further generalized by later studies [Chen et al., 2019, Baby and Wang, 2019].

Adaptive Bound. The two regularities C_T and V_T are in general incomparable, as they capture different aspects of the environmental changes. Therefore, it is desired to attain an adaptive guarantee, which is able to track the minimum of path-length and temporal variability bounds. Actually, Jadbabaie et al. [2015] propose a variant of optimistic mirror descent called adaptive optimistic mirror descent (AOMD), which achieves a more general result of order $O(\sqrt{D_T} + \min\{\sqrt{(1+D_T)(1+C_T)}, (1+D_T)^{1/3}T^{1/3}V_T^{1/3}\})$. The term D_T is the cumulative deviation of gradients to the predictable sequences, as defined in (1). Note that the regret bound implies the desired $O(\min\{\sqrt{T(1+C_T)}, T^{2/3}V_T^{1/3}\})$ bound since the term D_T is at most $O(T)$, yet could be tighter when the predictable sequences are in a high quality.

Finally, we would like to point out that Jadbabaie et al. [2015] achieve the fully adaptive bound by carefully designing a doubling trick mechanism, which demands the instantaneous quantities C_t, V_t and D_t at every iteration. Although D_t can be computed directly, the acquisition of C_t, and V_t requires solving optimization problems. Particularly, in order to calculate the temporal variability V_t, a non-convex problem of $\sup_{x\in\mathcal{X}} |f_t(x) - f_{t-1}(x)|$ will be involved at each iteration, which is time-consuming, even employing the concave-convex procedure [Yuille and Rangarajan, 2003] to solve this difference of convex functions programming. Our paper proposes a simple online algorithm that is able to achieve the *same* regret guarantee with Jadbabaie et al. [2015], yet is much *more efficient* to implement. We list the comparisons of existing dynamic regret bounds for convex functions in Table 1.

3 THE PROPOSED ALGORITHM

In this section, we propose our algorithm, which enjoys a fully adaptive bound without involving burdensome computations for non-convex problem solving. To achieve this goal, we face the following two challenges:

- how to incorporate the niceness of environments (D_T) in the dynamic regret analysis?
- how to make the algorithm adapt to various regularities (V_T, C_T) and track the best of them?

To address the first problem, similar to [Jadbabaie et al., 2015], we start with the optimistic mirror descent (OMD) algorithm [Rakhlin and Sridharan, 2013], which performs an extra gradient step to update the model with the predictable sequences $\{M_t\}_{t=1}^T$,

$$\begin{aligned} \widehat{x}_{t+1} &= \operatorname*{arg\,min}_{x\in\mathcal{X}} \eta_t\langle \nabla f_t(x_t), x\rangle + \mathcal{D}_{\mathcal{R}}(x, \widehat{x}_t), \\ x_{t+1} &= \operatorname*{arg\,min}_{x\in\mathcal{X}} \eta_{t+1}\langle M_{t+1}, x\rangle + \mathcal{D}_{\mathcal{R}}(x, \widehat{x}_{t+1}), \end{aligned} \tag{5}$$

where $\widehat{x}_{t+1}$ is an intermediate output and x_{t+1} is the final decision at iteration $t+1$. The notation $\mathcal{D}_{\mathcal{R}}(\cdot,\cdot)$ refers to the Bregman divergence with respect to the convex function $\mathcal{R}: \mathcal{X} \mapsto \mathbb{R}$. By setting $\mathcal{R}(\cdot)$ as a 1-strongly convex function and setting $\eta_t = R_{\max}\cdot\min\{1, \big(\sqrt{D_{t-2}} + \sqrt{D_{t-1}}\big)^{-1}\}$, Rakhlin and Sridharan [2013] show that OMD enjoys the following static regret,

$$\mathbf{Reg}_T^s \le 4R_{\max}(\sqrt{D_T}+1),$$

where $R_{\max}^2 = \sup_{x,y\in\mathcal{X}} \mathcal{D}_{\mathcal{R}}(x,y)$ and D_t is the empirical version of D_T as $D_t = \sum_{s=1}^t \|\nabla f_s(x_s) - M_s\|_*^2$

OMD algorithm can be seen as a variant of online mirror descent with an additional update guessing about the gradient of the next iteration. This framework provides a way to incorporate the niceness of the environments. For example, the learner can set $M_t = \nabla f_{t-1}(x_{t-1})$ when the online functions are known to vary gradually, which recovers the update scheme of Chiang et al. [2012].

In the following, we present the dynamic regret bounds for variants of OMD in terms of different regularities.

3.1 DYNAMIC REGRET WITH NICENESS

Path-length Bound. We first show that by running OMD with a fixed step size, we can obtain a dynamic regret bound relative to C_T and D_T.

Lemma 1. *When $\mathcal{R}(\cdot)$ is a 1-strongly convex function, running OMD with any fixed step size $\eta > 0$ satisfies*

$$\mathbf{Reg}_T^d \le \frac{\eta D_T}{2} + \frac{R_{\max}^2 + \gamma C_T}{2\eta},$$

provided $\mathcal{D}_{\mathcal{R}}(x,z) - \mathcal{D}_{\mathcal{R}}(y,z) \le \gamma\|x-y\|, \forall x,y,z \in \mathcal{X}$.

This lemma implies that the dynamic regret of OMD is at most $O(\sqrt{(1+C_T)(1+D_T)})$ by setting the step size as $\eta_{\mathtt{path}}^* = \sqrt{(R_{\max}^2 + \gamma C_T)/(1+D_T)}$. However, the optimal tuning is infeasible as quantities of C_T and D_T are unknown in advance. The standard doubling trick technique is not applicable due to the non-monotone behavior of the step size. Different from the special doubling mechanism used in AOMD [Jadbabaie et al., 2015], we grid search the optimal step size, which will be described in the next subsection.

Temporal Variability Bound. The OMD algorithm with a fixed step size also enjoys the temporal variability bound, which is shown as the following lemma.

Lemma 2. *Under the same condition in Lemma 1, running OMD with the fixed step size $\eta = \sqrt{(C_1 + C_2\lceil T/\Delta\rceil)/(1+D_T)}$ satisfies*

$$\mathbf{Reg}_T^d \le \sqrt{(1+D_T)(C_1 + C_2\lceil T/\Delta\rceil)} + 2\Delta V_T,$$

which holds for any parameter $\Delta \in [T]$. The constants are $C_1 = R_{\max}^2$ and $C_2 = \gamma\sqrt{2}R_{\max}$.

Note that the parameter Δ can be interpreted as the restarting period, similar to the restarted OMD proposed by Besbes et al. [2015]. Actually, OMD with a fixed step size can be regarded as a kind of restarted OMD. The difference is that we here have only one parameter to specify (η depends on Δ), while the restarted OMD of previous studies needs to set two parameters. In the dynamic regret analysis, we will exploit this key fact.

According to Lemma 2, by setting $\Delta_* = (C_2)^{2/3}(1+D_T)^{1/3}(1+T)^{1/3}V_T{}^{-2/3})$ and running OMD with $\eta_{\mathtt{var}}^* = \sqrt{(C_1 + C_2\lceil T/\Delta_*\rceil)/(1+D_T)}$, we can obtain an $O((1+D_T)^{1/3}T^{1/3}V_T^{1/3})$ dynamic regret bound[1]. However, the obstacle is that the optimal Δ_* requires the knowledge of V_T, which is also unknown in advance.

3.2 ADAPTIVE BOUND

Until now, we have separately incorporated the niceness of environments (D_T) with various regularities (C_T and V_T) in the dynamic regret analysis. There are still two steps to obtain an adaptive bound: (i) tuning the learning parameters $\eta_{\mathtt{path}}^*$ and $\eta_{\mathtt{var}}^*$; (ii) designing a mechanism to track the minimum of the regularities C_T and V_T.

Inspired by the recent works [van Erven and Koolen, 2016, Zhang et al., 2018, Zhao et al., 2020], we adopt the *online ensemble* to tune the unknown step size and track the best regularity simultaneously. Our approach is essentially an ensemble method [Zhou, 2012]. Concretely, we initiate multiple base algorithms, each running an OMD with a specific learning parameter, and then employ a master algorithm to aggregate their predictions. We show that the aforementioned goals can be handled in this framework with a carefully designed master algorithm to search over the OMDs with multiple step sizes.

Tuning Parameters. We first deal with the step size issue on the path-length bound. The optimal step size parameter is $\eta_{\mathtt{path}}^* = \sqrt{(R_{\max}^2 + \gamma C_T)/(1+D_T)}$, since the quantities of terms C_T and D_T are upper bounded, we can specify a pool $\mathcal{P}_{\mathtt{path}}$ containing the optimal one and search over the pool to identify it.

[1] The parameter Δ is constrained in $[T]$. We show in the proof of Theorem 2 that by setting $\widetilde{\Delta} = \max\{1, \min\{T, \Delta_*\}\}$, dynamic regret in the similar order is achievable.

Concretely, as a consequence of the boundedness of $\mathcal{X}$ in terms of Bergman divergence, we have $\|x - y\| \leq \sqrt{2}R_{\max}$, for all $x, y \in \mathcal{X}$, which leads to the fact $C_T \leq \sqrt{2}R_{\max}T$. Meanwhile, assuming the gradient and predictable sequence are bounded by G, then the term D_T is bounded by $4G^2T$. Thus, we have[2]

$$\eta_L = \frac{R_{\max}}{\sqrt{1+4G^2T}} \leq \eta^*_{\texttt{path}} \leq R_{\max}\sqrt{(1+\gamma T)} = \eta_H.$$

Then, we can specify a parameter pool in the range of $[\eta_L, \eta_H]$, and run several OMDs with the parameters in the pool. Specifically, the parameter pool is constructed with a logarithmic grid as

$$\mathcal{P}_{\texttt{path}} = \Big\{\eta_i = \frac{2^{i-1}R_{\max}}{\sqrt{1+4G^2T}} \mid i \in [N_{\texttt{path}}]\Big\},$$

where $N_{\texttt{path}} = \lceil \frac{1}{2}\log((1+\gamma T)(1+4G^2T))\rceil + 1$. By the construction of $\mathcal{P}_{\texttt{path}}$, we can see that there exists a parameter $\eta' \in \mathcal{P}_{\texttt{path}}$ satisfying $\eta^*_{\texttt{path}}/2 \leq \eta' \leq \eta^*_{\texttt{path}}$. Hence, once the output of the algorithm with the parameter η' is selected as the final decision, the regret bound of $O(\sqrt{(1+D_T)(1+C_T)})$ is achievable.

In the previous study [Zhang et al., 2018], Hedge algorithm [Littlestone and Warmuth, 1994] is a common choice for the master algorithm. Denoting x^i_t by prediction of the base algorithm running with step size η_i, the Hedge algorithm combines the bases by their weights w^i_t and output the final decision $x_{t+1} = \sum_{i\in[N_{\texttt{path}}]} w^i_{t+1}x^i_{t+1}$. The weight w^i_{t+1} is updated according to the previous performance of base algorithms

$$w^i_{t+1} = \frac{\exp(-\epsilon F^i_t)}{\sum_{i\in[N_{\texttt{path}}]}\exp(-\epsilon F^i_t)},$$

where $F^i_t = \sum_{s=1}^t f_t(x^i_t)$ is the cumulative loss of the i-th base algorithm at iteration t and $\epsilon > 0$ is the step size of the Hedge algorithm.

The Hedge algorithm ensures to track any base algorithm with regret at most $O(\sqrt{T\ln N})$. Thus, by taking Hedge as the master and OMDs with parameter pool $\mathcal{P}_{\texttt{path}}$ as base algorithms, we can obtain the path-length bound,

$$\mathbf{Reg}^d_T \leq O\big(\sqrt{T} + \sqrt{(1+C_T)(1+D_T)}\big).$$

Following a similar argument, we can derive the dynamic regret bound in terms of temporal variability without knowing the quantities of V_T. Since the parameter Δ is constrained in $[T]$, we have

$$\frac{C_1+C_2}{\sqrt{1+4G^2T}} \leq \eta^*_{\texttt{var}} \leq \sqrt{(C_1+C_2T)}.$$

[2]Without loss of generality, we assume $R_{\max} > 1$.

Thus, we can specify the parameter pool

$$\mathcal{P}_{\texttt{var}} = \Big\{\eta_i = \frac{2^{i-1}(C_1+C_2)}{\sqrt{1+4G^2T}} \mid i \in [N_{\texttt{var}}]\Big\},$$

to search the optimal parameter $\eta^*_{\texttt{var}}$, where $N_{\texttt{var}} = \lceil \frac{1}{2}\log\Big(\frac{(C_1+C_2T)(1+4G^2T)}{C1+C_2}\Big)\rceil + 1$. We should notice that since Lemma 2 holds only for $\Delta \in [T]$, what we search is actually $\widetilde{\Delta} = \max\{1, \min\{T, \Delta_*\}\}$, which can also yield the following dynamic regret bound

$$\mathbf{Reg}^d_T \leq O\big(\sqrt{T} + \max\{(1+D_T)^{1/3}T^{1/3}V_T^{1/3}, V_T\}\big).$$

Although an additional term V_T is introduced due to the constraint of the feasible interval length pool, it can be eliminated when aggregating the temporal variability bound with the path-length bound.

Aggregating Base Algorithms. We can run $N = N_{\texttt{path}} + N_{\texttt{var}}$ OMDs with the pool $\mathcal{P} = \mathcal{P}_{\texttt{path}} \bigcup \mathcal{P}_{\texttt{var}}$ and use Hedge to combine them. The scheme ensures the following dynamic regret guarantee.

Theorem 1. *When the predictable sequences satisfy $\|M_t\|_* \leq G$ for all $t \in [T]$, and $\mathcal{R}(\cdot)$ is 1-strongly convex, running Hedge as the master with N OMDs as base algorithms with the pool $\mathcal{P}$ satisfies*

$$\mathbf{Reg}^d_T \leq O(\sqrt{T} + \min\{R_C, R_V\}), \tag{6}$$

where $R_C = (1+D_T)(1+C_T)$ and $R_V = (1+D_T)^{1/3}T^{1/3}V_T^{1/3}$, provided $\mathcal{D}_{\mathcal{R}}(x,z) - \mathcal{D}_{\mathcal{R}}(y,z) \leq \gamma\|x-y\|, \forall x,y,z \in \mathcal{X}$.

Remark 1. Since D_T is at most $O(T)$, the dynamic regret is bounded by $O(\min\{\sqrt{T(1+C_T)}, T^{2/3}V_T^{1/3}\})$, which successfully tracks the minimum of various regularities including path-length and temporal variability.

3.3 FULLY ADAPTIVE BOUND

Note that Theorem 1 is not fully adaptive in the sense that it suffers an extra term of $O(\sqrt{T})$ aside from $O(\min\{R_C, R_V\})$. The former term will override the D_T term in the latter one, which makes the bound cannot fully exploit the niceness of environments, for instances, when the predictable sequence is in a high quality. This inadequacy comes from the master algorithm. To see this, we can decompose the dynamic regret $\mathbf{Reg}^d_T$ as,

$$\underbrace{\sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(x^i_t)}_{O(\sqrt{T})} + \underbrace{\sum_{t=1}^T f_t(x^i_t) - \sum_{t=1}^T f_t(x^*_t)}_{O(\min\{R_C,R_V\})},$$

which holds for any base algorithm indexed by $i \in [N]$. The first term is the regret of the master algorithm, while

Algorithm 1 `Master Algorithm`

Input: step size ϵ, parameter pools $\mathcal{P}$

1: Initiate N base algorithms $\mathcal{S} = \{S_i \mid i \in [N]\}$ by running Algorithm 2 with each step size $\eta_i \in \mathcal{P}$
2: Initialize: $L_0^i = 0$ for all $i \in [N]$, and receive M_1
3: **for** $t = 1$ **to** T **do**
4: Receive x_t^i from base and update weights by (9)
5: Play $x_t = \sum_{i\in[N]} w_t^i x_t^i$
6: Observe the function $f_t(\cdot)$, query the gradient $\nabla f_t(x_t)$ and receive M_{t+1}
7: Update $L_t^i = L_{t-1}^i + \langle \nabla f_t(x_t), x_t^i - x_t \rangle$
8: Send $\nabla f_t(x_t)$ and M_{t+1} to base algorithms
9: **end for**

Algorithm 2 OMD `(Base Algorithm)`

Input: step size $\eta_i \in \mathcal{P}$

1: Let x_1^i be any point in $\mathcal{X}$
2: **for** $t = 1$ **to** T **do**
3: Submit x_t^i to the master, then receive the gradient $\nabla f_t(x_t)$ and current predictable sequence M_{t+1}
4: Prepare the prediction for the next iteration as,

$$\widehat{x}_{t+1}^i = \operatorname*{arg\,min}_{x\in\mathcal{X}} \eta_i \langle \nabla f_t(x_t), x\rangle + \mathcal{D}_{\mathcal{R}}(x, \widehat{x}_t^i),$$
$$x_{t+1}^i = \operatorname*{arg\,min}_{x\in\mathcal{X}} \eta_i \langle M_{t+1}, x\rangle + \mathcal{D}_{\mathcal{R}}(x, \widehat{x}_{t+1}^i)$$

5: **end for**

the second is the dynamic regret of the base algorithm. The $O(\sqrt{T})$ term comes only from the master algorithm regret. Thus, to derive a fully adaptive bound, we require to substitute Hedge by one that can exploit the niceness.

To incorporate D_T in the master regret, our initial idea is to use optimistic Hedge [Rakhlin and Sridharan, 2013] to aggregate predictions from the N base algorithms $x_{t+1} = \sum_{i=1}^N w_{t+1}^i x_{t+1}^i$ with the weight update procedure

$$w_{t+1}^i \propto \exp\big(-\epsilon(F_t^i + m_{t+1}^i)\big), \tag{7}$$

where the optimism m_{t+1}^i can be seen as a guess about the loss suffered by the i-th base algorithm at iteration $t+1$, and enjoys the following regret guarantee [Chiang et al., 2012, Rakhlin and Sridharan, 2013].

Lemma 3. *By updating the weights as* (7) *and setting* $\epsilon = \sqrt{(\ln N + 2)/D_T^\infty}$, *for any base algorithm indexed by* $i \in [N]$, *the optimistic Hedge enjoys*

$$\sum_{t=1}^T \langle w_t, f_t\rangle - \sum_{t=1}^T f_t(x_t^i) \le 2\sqrt{(2 + \ln N)D_T^\infty},$$

where $D_T^\infty = \sum_{t=1}^T \|f_t - m_t\|_\infty^2$. *With a slight abuse of notation, we denote by* $f_t \in \mathbb{R}^N$ *the loss vector suffered by the* N *base algorithms, which takes* $f_t(x_t^i)$ *as its* i*-th entry. Similarly,* $w_t \in \mathbb{R}^N$ *is used for the weight vector and* $m_t \in \mathbb{R}^N$ *is for the optimism vector.*

The step size ϵ can be tunned by doubling trick since the value of D_T^∞ is monotone in the learning process. Combining Lemma 3 with aforementioned analysis and the fact that $f_t(x_t) \le \langle w_t, f_t\rangle$ by Jensen's inequality, we can derive a dynamic regret bound in the rate of $O(\sqrt{D_T^\infty} + \sqrt{(1 + C_T)(1 + D_T)})$. However, the bound is incompatible, as D_T^∞ is defined in terms of optimism m_t rather than the accessible predictable sequences M_t.

To associate $D_T^\infty = \sum_{t=1}^T \|f_t - m_t\|_\infty^2$ with $D_T = \sum_{t=1}^T \|\nabla f_t(x_t) - M_t\|_*^2$, we specify m_t with M_t as

$$m_t^i = \langle M_t, x_t^i - x_t\rangle. \tag{8}$$

The rationale behind is that m_t^i is the guess about $f_t(x_t^i)$, which can be substituted by the linearized surrogate loss $\ell_t(x_t^i) = \langle \nabla f_t(x_t), x_t^i - x_t\rangle$.[3] Thus, following similar spirit, it is natural to approximate m_t as (8), since M_t is a guess about $\nabla f_t(x_t)$. By reducing the common term $\langle M_{t+1}, x_{t+1}\rangle$, the weight update procedure with the linearized surrogate loss $\ell_t(x)$ and m_t becomes,

$$w_{t+1}^i \propto \exp\left(-\epsilon(L_t^i + \langle M_{t+1}, x_{t+1}^i\rangle)\right), \tag{9}$$

where $L_t^i = \sum_{s=1}^t \ell_t(x_t^i)$ is the cumulative surrogate loss for i-th base algorithm. Note that the predictions for base algorithm x_{t+1}^i is achievable at the beginning of iteration $t+1$, since its update only requires the information over past t iterations and M_{t+1}.

Let ℓ_t be the loss vector taking $\ell_t(x_t^i)$ as its i-th entry. By introducing the surrogate loss, we have

$$\begin{aligned}\|\ell_t - m_t\|_\infty &= \max_{i\in[N]}\{\langle \nabla f_t(x_t) - M_t, x_t^i - x_t\rangle\}\\ &\le \sqrt{2}R_{\max}\|\nabla f_t(x_t) - M_t\|_* \end{aligned} \tag{10}$$

where the last inequality holds due to the Cauchy-Schwartz inequality and the boundedness of $\mathcal{X}$. Thus, the term D_T^∞ is bounded by $2R_{\max}^2 D_T$.

By taking the optimistic Hedge with surrogate loss function as the master algorithm, the fully adaptive bound is obtained immediately. However, a direct combination would be time-consuming, particularly when the gradient acquisition requires heavy computations since we run $N = O(\log T)$ base algorithms and have to query the gradients for each one. This can be relieved by additionally incorporating the surrogate loss function to the base algorithm, where models are updated with $\nabla \ell_t(x_t^i)$ instead of $\nabla f_t(x_t^i)$. Since $\nabla \ell_t(x_t^i)$ = $\nabla f_t(x_t^i)$ for all $i \in [N]$, the times of querying gradients are reduced from N to 1 at

[3]The regret w.r.t $f_t(x)$ is upper bounded by that w.r.t. $\ell_t(x)$ Thus, the online algorithm can learn with $\ell_t(x)$ instead of $f_t(x)$.

every iteration. We summarize the master algorithm in Algorithm 1 and the base algorithms, OMD in Algorithm 2.

Theorem 2. *Under the same condition of Theorem 1, running the master algorithm (Algorithm 1) with N base OMDs (Algorithm 2) satisfies*

$$\mathbf{Reg}_T^d \leq O(\sqrt{D_T} + \min\{R_C, R_V\}),$$

where $R_C = \sqrt{(1+D_T)(1+C_T)}$ *and* $R_V = (1+D_T)^{1/3}T^{1/3}V_T^{1/3}$.

Theorem 2 is the main result of this paper, whose proof is provided in Appendix A. We leave proofs of the rest lemmas and theorems in the supplementary material.

Remark 2. The result in Theorem 2 is the same as the fully adaptive bound in Jadbabaie et al. [2015]. Comparing with the bound in Theorem 1, the fully adaptive one improves the term $O(\sqrt{T})$ introduced by the master algorithm to $O(\sqrt{D_T})$. The improvement enables the algorithm to exploit niceness of environments, particularly when high-quality predictable sequences are provided.

3.4 COMPARISON WITH PRECEDING WORK

Both our algorithm and AOMD [Jadbabaie et al., 2015] are built upon the optimistic mirror descent algorithm. The main difference is that AOMD tunes the optimal step size by the doubling trick, which involves a non-convex program at every iteration and requires burdensome calculations; while our approach avoids the heavy computations by searching over multiple OMDs. Due to the construction of the parameter pool, we only introduce $O(\log T)$ additional OMDs. Besides, by introducing surrogate loss to base algorithms, the acquisition of gradients is reduced from $O(\log T)$ to 1 at each iteration, which further accelerates the algorithm as the gradient evaluation is arguably the most time-consuming step in OMD.

Meanwhile, since our algorithm does not require the exact value of regularities C_T, it actually ensures a general dynamic regret bound, which supports the comparison against any sequence. By specifying the comparators for the worst case ones $\{x_t^*\}_{t=1}^T$ as the best fixed decision in hindsight, our result can adjust automatically from the fully adaptive dynamic bound to the static regret bound of $O(\sqrt{D_T})$, which facilitates another type of adaptivity.

4 EXPERIMENTS

In this section, we examine the efficacy and efficiency of our proposal on the matrix regression problem with the nuclear norm [Bach, 2008]. Empirical studies on classification tasks are also investigated, and we present results in the supplementary material due to page limits.

For the matrix regression problem, at iteration t, a sample (Z_t, y_t) arrives with predictable sequence M_t, where $Z_t \in \mathbb{R}^{p\times q}$ is a feature matrix and $y_t \in \mathbb{R}$ is the label. Then the learner makes the prediction $X_t \in \mathbb{R}^{p\times q}$ and suffers

$$f_t(X_t) = \frac{1}{a}\left[\frac{1}{2}(y_t - \mathrm{Tr}(\langle Z_t, X_t\rangle))^2 \ + \ b\|X_t\|_{\mathrm{nu}}\right],$$

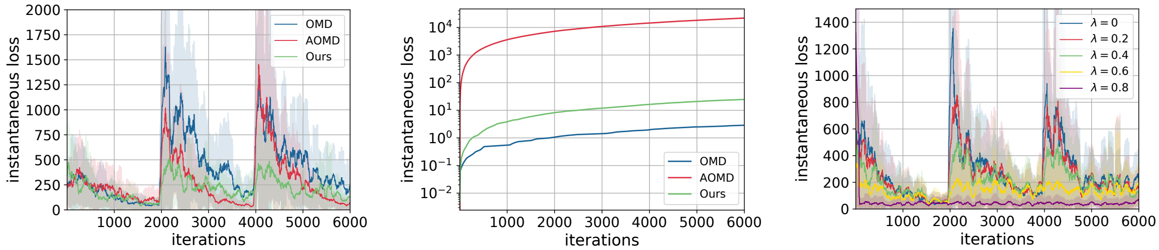
where $a, b > 0$ are constants and $\|\cdot\|_{\mathrm{nu}}$ is the nuclear norm, defined as the sum of singular values of the matrix.

Contenders. To verify the efficiency of our algorithm, we compare our method with AOMD and the vanilla OMD with the static regret guarantee. We set $\mathcal{R}(x) = \frac{1}{2}\|x\|_F^2$ for both algorithms, where $\|\cdot\|_F$ is the Frobenius norm. For AOMD, we use the concave-convex procedure to handle the non-convex program, which is solved by a sequence of quadratically constrained quadratic program.

Setting. All contenders are compared in dynamic environments, where the potential optimal decision changes abruptly. Specifically, we consider a 3-stage scenario containing 2000 iterations in each iteration. Denoting by X_*^k the potentially best decision for k-th stage, the label is assigned as $y_t = \mathrm{Tr}(\langle Z_t, X_*^k\rangle) + \nu$, where the data point Z_t is randomly sampled from $[-1,1]^{20\times 20}$ and $\nu \sim \mathcal{N}(0, 10)$ is the random Gaussian noise.

Meanwhile, to see the effectiveness of our algorithm in exploiting the niceness of environments. We simulate predictable sequences as $M_t = \lambda \cdot \nabla f_t(x_t)$, where $\lambda \in [0,1]$ is a parameter adjusting the quality of the predictable sequence between the ideal case where $M_t = \nabla f_t(x_t)$ (i.e., $D_T = 0$) and the basic case where $M_t = 0$. A larger λ implies a higher quality of the predictable sequences.

Results. We focus on the efficacy first. Figure 1(a) shows the mean and standard deviation of instantaneous loss over a moving time window of 100 iterations when $\lambda = 0.4$. We can see that both AOMD and our approach perform better than the vanilla OMD, which validates the advantage of adaptive algorithms. It is also interesting to notice that AOMD converges better in the stationary period than ours. The reason might be that AOMD runs with the fixed step size until the doubling condition is satisfied, while our method maintains multiple step sizes, where the price has to pay for hedging the non-stationarity. Besides, Figure 1(b) presents comparisons on the running time in a logarithmic scale, where our method (24 seconds) is slightly slower than vanilla OMD but significantly faster than AOMD (around 6 hours), which validates the efficiency of our method. Last, Figure 1(c) reports that the performance of our method improves with better predictable sequences, which shows that it can efficiently and adaptively exploit the niceness of environments.



(a) Comparison on instantaneous loss (b) Comparison on running time (c) Different qualities of M_t

Figure 1: The dynamic regret divided by iteration of three methods with the quality of hint varying from poor to ideal

5 CONCLUSION

In this paper, we design a simple algorithm enjoying the fully adaptive bound in Jadbabaie et al. [2015], which achieves the minimizer of previous results in terms of different regularities C_T, V_T and can benefit from the niceness of environments measured by D_T. By employing the online ensemble and designing a new master algorithm, our algorithm avoids solving the non-convex program and is more computationally efficient. Experiments validate the effectiveness and efficacy of our proposed algorithm.

Acknowledgements

This research was supported by the National Key R&D Program of China (2018YFB1004300) and NSFC (61921006). We are grateful for the anonymous reviewers for their helpful comments.

A PROOF OF THEOREM 2

Proof. In this section, we first present the proof of the path-length bound, followed by that of the temporal variability bound. The combination of these two types of results implies the fully adaptive bound in Theorem 2.

Path-length Bound. First, we can see that the dynamic regret w.r.t the original loss $f_t(x)$ can be bounded by that of the surrogate loss $\ell_t(x) = \langle \nabla f_t(x_t), x - x_t \rangle$ as

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x_t^*) \leq \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x_t^*),$$

where the inequality holds due to the Jensen's inequality $f_t(x_t) - f_t(x_t^*) \leq \langle \nabla f_t(x_t), x_t - x_t^* \rangle$ and the fact that $\ell_t(x_t) = 0$. Then, for any base algorithm indexed by $i \in [N]$, we can decompose the dynamic regret as

$$\underbrace{\sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x_t^i)}_{\texttt{Reg-Mas(i)}} + \underbrace{\sum_{t=1}^{T} \ell_t(x_t^i) - \sum_{t=1}^{T} \ell_t(x_t^*)}_{\texttt{Reg-Base(i)}}.$$

where the first term is regret of the master algorithm while the second dynamic regret of the base algorithm.

As for the master regret, our algorithm runs a new variant of optimistic Hedge, where the loss suffered by each base algorithm is $\ell_t(x_t^i) = \langle \nabla f_t(x_t), x_t^i - x_t \rangle$ and predictable sequences $m_t^i = \langle M_t, x_t^i - x_t \rangle$ are offered. Thus, due to Lemma 3 and the argument in (10), for any base algorithm indexed by i and each iteration t, we have

$$\ell_t(x_t) - \ell_t(x_t^i) = \sum_{i \in [N]} w_t^i \ell_t(x_t^i) - \ell_t(x_t^i)$$

$$\leq 2R_{\max} \sqrt{(4 + 2 \ln N) \sum_{t=1}^{T} \|\nabla f_t(x_t) - M_t\|_*^2}$$

Summing over T iterations, we obtain

$$\texttt{Reg-Mas(i)} \leq 2R_{\max} \sqrt{(4 + 2 \ln N) D_T}.$$

Then, we proceed to analyze the base regret for path-length bound. The optimal step size for the path-length bound is $\eta^*_{\texttt{path}} = \sqrt{(R_{\max}^2 + \gamma C_T)/(1 + D_T)}$. Due to the construction of $\mathcal{P}_\eta$, there must exist a base algorithm indexed by $p \in [N]$, whose step size satisfies $\eta^*_{\texttt{path}}/2 \leq \eta_p \leq \eta^*_{\texttt{path}}$. Therefore, for the p-th base algorithm, by applying Lemma 5 in the supplementary material, a counterpart of Lemma 1 incorporating the surrogate loss function, we know that the base regret is bounded by

$$\begin{aligned} \texttt{Reg-Base(p)} &\leq \frac{\eta_p D_T}{2} + \frac{R_{\max}^2 + \gamma C_T}{2\eta_p} \\ &\leq \frac{\eta^*_{\texttt{path}} D_T}{2} + \frac{R_{\max}^2 + \gamma C_T}{\eta^*_{\texttt{path}}} \\ &= \underbrace{\frac{3}{2}\sqrt{(1 + D_T)(\gamma C_T + R_{\max}^2)}}_{\texttt{Path-Base}}. \end{aligned}$$

Choosing the p-th base algorithm as the intermediate term in the decomposition, we obtain the path-length bound as,

$$\mathbf{Reg}_T^d \leq 2R_{\max} \sqrt{(4 + 2 \ln N) D_T} + \texttt{Path-Base}$$

Temporal Variability Bound. We first note that the argument in the proof of path-length bound does not directly apply for proving the temporal variability bound, due to the introduction of the surrogate loss. Specifically, previous arguments can only yield a temporal variability bound in terms of $V_T^\ell = \sum_{t=2}^T \sup_{x\in\mathcal{X}} |\ell_t(x) - \ell_{t-1}(x)|$, which is hard to be converted to the desired V_T w.r.t. the original function f_t. To address the difficulty, we decompose the dynamic regret as follows.

$$\begin{aligned}\mathbf{Reg}_T^d = &\underbrace{\sum_{i=1}^{\lceil T/\Delta\rceil}\sum_{t\in\mathcal{I}_i} f_t(x_t) - \sum_{i=1}^{\lceil T/\Delta\rceil}\sum_{t\in\mathcal{I}_i} f_t(x^*_{\mathcal{I}_i})}_{\texttt{term A}} \\ &+ \underbrace{\sum_{i=1}^{\lceil T/\Delta\rceil}\sum_{t\in\mathcal{I}_i} f_t(x^*_{\mathcal{I}_i}) - \sum_{i=1}^{\lceil T/\Delta\rceil}\sum_{t\in\mathcal{I}_i} f_t(x^*_t)}_{\texttt{term B}},\end{aligned} \tag{11}$$

where $\mathcal{I}_i = [s_i, e_i]$ is the i-th interval with the length Δ, and $s_i = (i-1)\cdot\Delta + 1$, $e_i = i\cdot\Delta$. Notation $x^*_{\mathcal{I}_i} = \arg\min_{x\in\mathcal{X}} \sum_{t\in\mathcal{I}_i} f_t(x)$ refers to the best fixed decision in interval $\mathcal{I}_i$.

The term A is the regret of learner's decisions comparing to the best decisions over consecutive intervals w.r.t. the original loss, which can be bounded by that of the surrogate loss by the Jensen's inequality as

$$\texttt{term A} \le \sum_{i=1}^{\lceil T/\Delta\rceil}\sum_{t\in\mathcal{I}_i} \ell_t(x_t) - \sum_{i=1}^{\lceil T/\Delta\rceil}\sum_{t\in\mathcal{I}_i} \ell_t(x^*_{\mathcal{I}_i}).$$

As shown by Lemma 6 in the supplementary material, the above inequality can be further bounded by

$$\begin{aligned}\texttt{term A} \le\ & 2R_{\max}\sqrt{(4+2\ln N)D_T} \\ &+ 2\sqrt{(1+D_T)(C_1 + C_2\lceil T/\Delta\rceil)},\end{aligned} \tag{12}$$

for any parameter $\Delta \in [T]$. As for term B, following the same argument in Besbes et al. [2015], we ensure that

$$\texttt{term B} \le 2\Delta V_T. \tag{13}$$

Note that here we can achieve a temporal variability term V_T w.r.t. the *original function* f_t. Combining (12), (13) and setting $\Delta = \widetilde{\Delta}$ where $\widetilde{\Delta} = \max\{1, \min\{T, (C_2)^{2/3}(1+D_T)^{1/3}(1+T)^{1/3}V_T{}^{-2/3})\}$, we can obtain that

$$\begin{aligned}\mathbf{Reg}_T^d \le\ & 2R_{\max}\sqrt{(4+2\ln N)D_T} \\ &+ \underbrace{2\sqrt{(1+D_T)(C_1 + C_2\lceil T/\widetilde{\Delta}\rceil)} + 2\widetilde{\Delta}V_T}_{\texttt{Var-Base}}.\end{aligned}$$

Since what we are searching is actually $\widetilde{\Delta}$ rather than Δ_*, we discuss the result in the following three cases.

- Case 1 ($\Delta_* \in [1, T]$): by setting $\widetilde{\Delta} = \Delta_*$, we have
  $$\begin{aligned}\texttt{Var-Base} \le\ & \sqrt{C_1(1+D_T)} \\ &+ 4C_2^{2/3}(1+D_T)^{1/3}(1+T)^{1/3}V_T^{1/3}.\end{aligned}$$
- Case 2 ($\Delta_* < 1$): by setting $\widetilde{\Delta} = 1$ and the fact $C_2\sqrt{(1+D_T)(1+T)} < V_T$, we have
  $$\texttt{Var-Base} \le \sqrt{C_1(1+D_T)} + 2(1 + C_2/\sqrt{C_2})V_T.$$
- Case 3 ($\Delta_* \ge T+1$): by setting $\widetilde{\Delta} = T$ and the fact that : $C_2\sqrt{1+D_T} \ge (1+T)V_T$, we have
  $$\texttt{Var-Base} \le (\sqrt{C_1} + 2C_2 + 2\sqrt{2C_2})\sqrt{1+D_T}.$$

Although the temporal variability bound behaves differently in various cases, we can aggregate it with the path-length bound to yield the desired regret bound.

Combining Base Algorithms. Our algorithm enjoys both path-length and temporal variability bound, namely,

$$\begin{aligned}\mathbf{Reg}_T^d \le\ & 2R_{\max}\sqrt{(4+2\ln N)D_T} \\ &+ \min\{\texttt{Path-Base}, \texttt{Var-Base}\}.\end{aligned} \tag{14}$$

We show that (14) actually implies the following claim

$$\begin{aligned}\mathbf{Reg}_T^d \le O\big(&\sqrt{D_T\ln N} + \min\{\sqrt{(1+D_T)(1+C_T)}, \\ &(1+D_T)^{\frac{1}{3}}(1+T)^{\frac{1}{3}}V_T^{\frac{1}{3}}\}\big),\end{aligned} \tag{15}$$

which is proved by considering the following three cases.

- Case 1 ($\Delta_* \in [1, T]$): $\texttt{Var-Base} = O((1+D_T)^{\frac{1}{3}}(1+T)^{\frac{1}{3}}V_T^{\frac{1}{3}})$, (14) and (15) are in same order.
- Case 2 ($\Delta_* < 1$): $\texttt{Var-Base} = O(V_T)$, since $V_T > C_2\sqrt{(1+D_T)(1+T)}$. Combining with the fact $C_T \le \sqrt{2}R_{\max}T$, we have $\texttt{Path-Base} \le \widetilde{C}V_T$ where $\widetilde{C}$ is a constant. Therefore, (14) is at most
  $$\begin{aligned}&O\big(\sqrt{D_T\ln N} + \min\{\sqrt{(1+D_T)(1+C_T)}, V_T\}\big) \\ &= O\big(\sqrt{D_T\ln N} + \sqrt{(1+D_T)(1+C_T)}\big),\end{aligned}$$
  which implies (15) as $V_T \gtrsim \sqrt{(1+D_T)(1+T)}$.
- Case 3: ($\Delta_* \ge T+1$): $C_2\sqrt{1+D_T} \ge (1+T)V_T$. In this case, (15) becomes
  $$\begin{aligned}\mathbf{Reg}_T^d &\le O\big(\sqrt{D_T\ln N} + (1+D_T)^{\frac{1}{3}}(1+T)^{\frac{1}{3}}V_T^{\frac{1}{3}}\big) \\ &= O(\sqrt{D_T\ln N}).\end{aligned}$$
  Since $\texttt{Var-Base} = O(\sqrt{1+D_T})$, (14) becomes $\mathbf{Reg}_T^d \le O(\sqrt{D_T\ln N})$, which implies (15).

Notice that we treat the double logarithmic factors in T as constants, following previous studies [Luo and Schapire, 2015, Adamskiy et al., 2016].

We complete the proof of the fully adaptive bound in Theorem 2 by combining all above three cases. □

References

J. D. 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 415–424, 2008.

D. Adamskiy, W. M. Koolen, A. V. Chernov, and V. Vovk. A closer look at adaptive regret. *Journal of Machine Learning Research*, 17:23:1–23:21, 2016.

D. Baby and Y.-X. Wang. Online forecasting of total-variation-bounded sequences. In *Advances in Neural Information Processing Systems 32*, pages 11071–11081, 2019.

F. R. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9:1019–1048, 2008.

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

X. Chen, Y. Wang, and Y.-X. Wang. Non-stationary stochastic optimization under $L_{p,q}$ -variation measures. *Operations Research*, 67(6):1503–1782, 2019.

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

J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

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

E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 57–68, 2008.

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

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.

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 Annual Conference Computational 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.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Proceedings of the 26th Conference On Learning Theory*, pages 993–1019, 2013.

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

M. Sugiyama and M. Kawanabe. *Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation*. MIT Press, 2012.

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

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.

A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, pages 915–936, 2003.

L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou. Improved dynamic regret for non-degeneracy 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, 2018.

P. Zhao and L. Zhang. Improved analysis for dynamic regret of strongly convex and smooth functions. *ArXiv preprint*, arXiv:2006.05876, 2020.

P. Zhao, X. Wang, S. Xie, L. Guo, and Z.-H. Zhou. Distribution-free one-pass learning. *IEEE Transaction on Knowledge and Data Engineering*, 2019.

P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou. Dynamic regret of convex and smooth functions. *ArXiv preprint*, 2020.

Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.

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