Understanding and Exploration for "Adaptive Online Learning in Dynamic Environments"

Zhitong Gao*, **Haotian Tian***, **Tianyue Zhou***
ShanghaiTech University
{gaozht,tianht2022,zhouty1}@shanghaitech.edu.cn

Abstract

In this report, we do the literature review of the paper "Adaptive Online Learning in Dynamic Environments". We first demonstrate our understanding of the paper by giving a brief description of the method and then we try to improve it both theoretically and empirically. Theoretically, we analyze how the performance improves when adding an additional prediction during the Expert updating, by designing a specific algorithm and derive its Regret. Then we design some empirical experiments to understand the performance of the proposed Ader algorithm under different environments with various dynamic levels. We find that this algorithm might have some limitations when some prior information of the environment is not available ahead. We therefore explore several novel algorithms to try to decide these parameters dynamically according to the feedback of the environment in an online manner.

1 Introduction

Online learning has become a critical area of research due to its ability to adapt to changing environments and make predictions based on real-time data. This approach is particularly useful in dynamic environments where the underlying data distribution can change over time. One of the key challenges in online learning is to design algorithms that can adapt to these changes and maintain good performance. In this context, the paper "Adaptive Online Learning in Dynamic Environments" [1] presents a novel method, the Adaptive Dynamic Expert Regret (Ader) algorithm, which aims to address this challenge. The Ader algorithm leverages multiple experts with different step sizes with their weights representing an understanding of the environment, thereby achieving robust performance in dynamic environments.

In this report, we conduct a comprehensive literature review of the aforementioned paper. Firstly, We give a summarization, as well as our own understanding of the problem, proposed methods, and theories, which is shown in Sec. 2. Then, we go deeper and try to do some explorations both theoretically and empirically. Our exploration begins with a detailed analysis of the algorithm's performance enhancement when an additional prediction is incorporated during the model updating process. We design a specific algorithm for this purpose and derive its upper bound, providing a theoretical foundation for performance improvement. We show this content in Sec. 3 Subsequently, we turn our attention to the empirical performance of the Ader algorithm. Through a series of carefully designed experiments, we investigate the algorithm's behavior under different environments characterized by varying dynamic levels. Our findings reveal that while the Ader algorithm exhibits robust performance in many scenarios, it may encounter limitations when certain prior information about the environment is not readily available. To address these limitations, we propose several novel algorithms that dynamically determine parameters based on environmental feedback. These

*denotes equal contribution.

Preprint. Under review.

	Static Regret	Dynamic Regret
Comparator	Best fixed decision	Sequence (optimal/local min.)
Environment	Static	Dynamic
Regret Measure	$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in X} \sum_{t=1}^{T} f_t(x)$	$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(u_t)$

Table 1: Comparison of Static and Dynamic Regret

algorithms operate in an online manner, continuously adjusting their strategies in response to changes in the environment. The relevant content can be found in Sec. 4.

All in all, this report provides a comprehensive review and analysis of the Ader algorithm and contributes to the ongoing research in online learning in dynamic environments. We indicate our individual contribution in Appendix A

2 Paper Review

In this section, we will give a brief review of the problem, motivation, method and theories proposed in the original paper [1]. [2]

2.1 Problem and Motivation

Problem The paper studies the problem of Online Convex Optimization (OCO) under dynamic environments. Specifically, it considers the situation when the system is changing over time. Therefore, it differs from the static regret that we learned in the lecture. It tackles a concept known as *general dynamic regret*, which can be represented as follows:

$$R(u_1, \ldots, u_T) = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(u_t) \tag{1}$$

where $u_1, \cdots, u_T$ are comparators, and their relative difference, known as *path length*, reveals how dynamic the environment is, thus is a very important characteristic of the environment system. We note that, although in this paper [1], the comparators are defined as a random sequence, in the original paper of [2], it is defined as the optimal sequence given a path-length (cf. Definition 7 in [2] for details). We actually find the latter one easier to comprehend.

Comparison and Motivation The understanding of the general dynamic regret and the corresponding environmental situation is very important to properly understand the main idea of the method and its realistic usage. A comparison of static regret and dynamic regret is summarized in Table 1. As we can see, static regret compares the cumulative loss of the learner against the best constant point chosen in hindsight, which makes this measure becomes inadequate when the environment is changing. On the other hand, most existing studies on dynamic regret focus on a constrained form, where the sequence of comparators is composed of local minima of online functions, which is unsuitable for the stationary problem as it potentially results in overfitting.To better understand how the performance changes of different algorithms under different environmental situations, we also do an empirical study in Sec. 4. For these reasons, the authors study the problem of general dynamic regret and aim to design a model that is suitable for a variety of environments.

2.2 Method

The paper presents a novel method named Adaptive Learning for Dynamic Environment (Ader) for addressing the challenge of online convex optimization (OCO) in dynamic environments. Following, we first introduce the assumptions made in the paper, followed by the main idea and specific algorithm of the paper. Later, we explain how it is improved by using a surrogate loss and extend to dynamic models.

[2] Since there is no empirical part in this paper, we will put the empirical analysis, together with our improvements in Sec. 4.

Assumptions The Ader algorithm is built upon a set of assumptions about the online problem, including the boundedness of function values and gradients, and the diameter of the domain. These assumptions are common in online learning studies, therefore, we do not repeat details here and you can find this part in Sec.3.1 in the original papaer [1].

Main idea and our understanding The main idea of Ader is to run multiple instances of the Online Gradient Descent (OGD) algorithm in parallel, each with a different step size, and choose the best one using an expert-tracking algorithm. As we understand, this can be interpreted as trying to figure out a latent variable that encodes the dynamic level of the environment, which is correlated to the step size in the gradient descent algorithm. Intuitively, a more dynamic environment may correspond to a larger step size, and vice versa.

The specific algorithm Specifically, each expert is an instance of OGD, the step size is defined as

$$\eta_i = 2^{i-1} DG \sqrt{\frac{7}{2T}}, \tag{2}$$

where D is the diameter of the domain, G is the bound on the gradients, and T is the total number of iterations. In each round, a meta-algorithm receives a set of predictions from all experts and outputs the weighted average, which is calculated as

$$x_t = \sum_{\eta \in H} w_{\eta t} x_{\eta t}, \tag{3}$$

where $w_{\eta t}$ is the weight assigned to expert E_η. After observing the loss function, the weights of experts are updated according to the exponential weighting scheme:

$$w_{\eta t+1} = \frac{w_{\eta t} e^{-\alpha f_t(x_{\eta t})}}{\sum_{\mu \in H} w_{\mu t} e^{-\alpha f_t(x_{\mu t})}}. \tag{4}$$

The overall pipeline of the framework can be found in the Algorithm 1 and 2 of the original paper [1]. And a modified version of the algorithm is shown in 1.

Improve efficiency with surrogate loss The basic approach of Ader needs to query the value and gradient of the loss function multiple times in each round, thus being inefficient when the expert number is large. To address this, the authors introduce an improved approach by incorporating surrogate loss. The loss is formally defined as

$$\ell_t(x) = \langle \nabla f_t(x_t), x - x_t \rangle, \tag{5}$$

which is a linear approximation of the original loss function. Importantly, this loss is designed to be an upper bound of the original loss, ensuring that minimizing the surrogate loss will also minimize the original loss. By using the surrogate loss in the original algorithm, the authors make it to reduce the number of gradient evaluations to one per iteration.

Extend to dynamic models Finally, the authors extend Ader to the case where a sequence of dynamical models is given. These models can be used to characterize the comparators of interest. The authors modify the expert-algorithm to utilize the dynamical model, and provide a dynamic regret bound for the new algorithm. This extension further demonstrates the flexibility and adaptability of Ader in handling OCO in dynamic environments. We refer to details in this part in Section 3.5 of the original paper [1].

2.3 Proofs

The authors provide proofs for several theorems that establish the performance and properties of Ader. In the following, we will briefly go through the main contents and ideas for the proofs.

Proof of Theorem 1 The authors establish the regret bound of the Expert algorithm in Ader method. They start by defining an auxiliary sequence and then use the standard analysis technique to derive an inequality that bounds the difference between the loss at the current point and the loss at an arbitrary point. This inequality is then summed over all iterations to obtain the cumulative regret. The authors then use several inequalities and assumptions to simplify the regret bound, which results in a bound that depends on the domain's diameter, the step size, the gradient norm, and the path-length of the comparator sequence.

Proof of Theorem 2 This proof introduces the concept of minimax regret and uses it to analyze the performance of the Ader method in a game-theoretic setting. The authors first recall the definition of minimax static regret and then define the minimax dynamic regret with respect to a set of comparator sequences. They then consider two cases based on the path-length and derive lower bounds for the minimax dynamic regret in both cases. The proof concludes by combining these bounds to obtain a lower bound for the minimax dynamic regret that depends on the gradient norm, the diameter of the feasible set, the number of iterations, and the path-length.

Proof of Theorem 3 The authors divide the analysis into three parts. First, they show that the cumulative loss of the meta-algorithm is comparable to all experts. Then, they demonstrate that for any sequence of comparators, there is an expert whose dynamic regret is almost optimal. Finally, they combine the regret bounds of the meta-algorithm and experts to obtain the dynamic regret of the Ader method. The proof involves several steps, including bounding the regret of the meta-algorithm with respect to all experts, identifying an expert with near-optimal dynamic regret, and combining these results to derive the dynamic regret bound of the Ader method.

Proofs of Theorems 4, 5 and 6 These proofs follow similar lines of reasoning, but they incorporate additional assumptions and techniques. For example, the proof of Theorem 4 replaces the original function with a surrogate loss and derives a dynamic regret bound in terms of the surrogate losses. The proof of Theorem 5 incorporates dynamical models into the analysis and updates the theoretical guarantees of experts accordingly. The proof of Theorem 6 extends the analysis to a setting where the comparator sequence is generated by a dynamical model, and it derives a regret bound that depends on the distance between the comparator sequence and the sequence generated by the dynamical model.

Overall, the theoretical analysis in the paper is comprehensive and rigorous. It provides strong theoretical guarantees for the performance of the Ader method and sheds light on the key factors that influence its performance.

3 Theoretical Exploration

Based on the paper *Adaptive Online Learning in Dynamic Environments*, we added a prediction $\hat{f}_t(\cdot)$, which has the same property as its loss function $f_t(\cdot)$.Finally We prove that given a prediction before update the Experts and combine them to submit, the worst case also has the regret $\mathcal{O}(\sqrt{T(1+P_T)})$, or get a constant regret if prediction is perfect.

3.1 Statement

In the original paper *Adaptive Online Learning in Dynamic Environments*, the **Algorithm 2** Ader: Exprt-algorithm is that:

$$\mathbf{x}_{t+1}^{\eta} = \Pi_{\mathcal{X}}(\mathbf{x}_t^{\eta} - \eta \nabla f_t(\mathbf{x}_t^{\eta})) \tag{6}$$

which is the standard online gradient descent (OGD). It's one step update and without any prediction, here we wish to add a prediction, denote as $\hat{f}_t(\cdot)$ for t round, i.e.

$$\mathbf{y}_{t+1}^{\eta} = \Pi_{\mathcal{X}}(\mathbf{x}_t^{\eta} - \eta \nabla f_t(\mathbf{x}_t^{\eta})) \tag{7}$$
$$\mathbf{x}_{t+1}^{\eta} = \Pi_{\mathcal{X}}(\mathbf{y}_{t+1}^{\eta} - \eta \nabla \hat{f}_t(\mathbf{y}_{t+1}^{\eta})) \tag{8}$$

in the next subsection, we will prove its Regret. The **Ader: Expert-algorithm with prediction** is in **Algorithm 1**.

Algorithm 1 Ader: Expert-algorithm with prediction

Require: The step size η

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

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

Submit $\mathbf{x}_1^\eta$ to the meta-algorithm

Receive gradient $\nabla f_t(\mathbf{x}_1^\eta)$ from the meta-algorithm

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

Using $\mathbf{y}_{t+1}^\eta$ to get $\hat{f}_t(\mathbf{y}_{t+1}^\eta)$ as a prediction of $f_{t+1}(\cdot)$

Update Expert with prediction

$$\mathbf{x}_{t+1}^\eta = \Pi_{\mathcal{X}}(\mathbf{y}_{t+1}^\eta - \eta \nabla \hat{f}_t(\mathbf{y}_{t+1}^\eta))$$

end for=0

3.2 Main result

Theorem 3.1 (Ader with precition). *Consider the online gradient descent (OGD) with prediction $\hat{f}_t(\cdot)$ following*

$$\mathbf{y}_{t+1}^\eta = \Pi_{\mathcal{X}}(\mathbf{x}_t^\eta - \eta \nabla f_t(\mathbf{x}_t^\eta))$$
$$\mathbf{x}_{t+1}^\eta = \Pi_{\mathcal{X}}(\mathbf{y}_{t+1}^\eta - \eta \nabla \hat{f}_t(\mathbf{y}_{t+1}^\eta))$$

where $\Pi_{\mathcal{X}}$ denotes the projection onto the nearest point in $\mathcal{X}$. Assume that $\hat{f}_t(\cdot)$ has the same property as $f_t(\cdot)$, such as convexity, smoothness and bounded. Under Assumptions 2 and 3 in original paper, OGD with prediction satisfy

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \leq \frac{D^2}{\eta} + \frac{D}{\eta}\sum_{t=2}^{T}\|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2$$
$$\leq \frac{D^2}{\eta} + \frac{D}{\eta}\sum_{t=2}^{T}\|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2 + 2\eta T G^2$$

for any comparator sequence $\mathbf{u}_1, ..., \mathbf{u}_T \in \mathcal{X}$.

Proof. We first define

$$F(\mathbf{x}) = f_t(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_t\|_2^2 \tag{9}$$

using pushback lemma,

$$F(\mathbf{x}_{t+1}) \leq F(\mathbf{x}) - \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2 \tag{10}$$

let $\mathbf{x} = \mathbf{y}_{t+1}$ and $\hat{f}_t(\cdot)$ has the same property as its loss function $f_t(\cdot)$, we have

$$\hat{f}_t(\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \leq \hat{f}_t(\mathbf{y}_{t+1}) + \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{y}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \tag{11}$$

for observed loss $f_t(\cdot)$ and let $\mathbf{x} = \mathbf{u}_t$, we also have

$$f_t(\mathbf{y}_{t+1}) + \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{y}_t\|_2^2 \leq f_t(\mathbf{u}_t) + \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 \tag{12}$$

$$\frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{y}_t\|_2^2 \leq -f_t(\mathbf{y}_{t+1}) + f_t(\mathbf{u}_t) + \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 \tag{13}$$

substitute $\frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{y}_t\|_2^2$ in (11) with (13), we have

$$\begin{aligned}\hat{f}_t(\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \leq f_t(\mathbf{u}_t) &+ \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 \\ &+ \left[\hat{f}_t(\mathbf{y}_{t+1}) - f_t(\mathbf{y}_{t+1})\right] - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2\end{aligned} \tag{14}$$

$$\left[\hat{f}_t(\mathbf{x}_t) - f_t(\mathbf{x}_t)\right] + f_t(\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \leq f_t(\mathbf{u}_t) + \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 + \left[\hat{f}_t(\mathbf{y}_{t+1}) - f_t(\mathbf{y}_{t+1})\right] - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \tag{15}$$

$$f_t(\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{y}_t\|_2^2 \leq f_t(\mathbf{u}_t) + \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 + \left[\left(\hat{f}_t(\mathbf{y}_{t+1}) - f_t(\mathbf{y}_{t+1})\right) - \left(\hat{f}_t(\mathbf{x}_t) - f_t(\mathbf{x}_t)\right)\right] - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \tag{16}$$

delete the second term $\frac{1}{2\eta}\|\mathbf{x}_t - \mathbf{y}_t\|_2^2$ in the left hand, we have

$$\begin{aligned} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \leq& \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 \\ &+ \left[\left(\hat{f}_t(\mathbf{y}_{t+1}) - \hat{f}_t(\mathbf{x}_t)\right) - \left(f_t(\mathbf{y}_{t+1}) - f_t(\mathbf{x}_t)\right)\right] - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \end{aligned} \tag{17}$$

using convexity, i.e.

$$\hat{f}_t(\mathbf{y}_{t+1}) - \hat{f}_t(\mathbf{x}_t) \leq \langle \nabla \hat{f}_t(\mathbf{y}_{t+1}), \mathbf{y}_{t+1} - \mathbf{x}_t \rangle,\ -\left(f_t(\mathbf{y}_{t+1}) - f_t(\mathbf{x}_t)\right) \leq \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{y}_{t+1} \rangle \tag{18}$$

then

$$\begin{aligned} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \leq& \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 \\ &+ \langle \nabla \hat{f}_t(\mathbf{y}_{t+1}), \mathbf{y}_{t+1} - \mathbf{x}_t \rangle + \langle \nabla f_t(\mathbf{x}_t), \mathbf{x}_t - \mathbf{y}_{t+1} \rangle - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \\ =& \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 \\ &+ \langle \nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t), \mathbf{y}_{t+1} - \mathbf{x}_t \rangle - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \end{aligned} \tag{19}$$

notice that

$$\begin{aligned} &\langle \nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t), \mathbf{y}_{t+1} - \mathbf{x}_t \rangle - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \\ &= \langle \nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t), \mathbf{y}_{t+1} - \mathbf{x}_t \rangle - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{x}_t\|_2^2 \\ &\quad - \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \\ &= -\|\frac{1}{\sqrt{2\eta}}\left(\mathbf{y}_{t+1} - \mathbf{x}_t\right) - \sqrt{\frac{\eta}{2}}\left(\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\right)\|_2^2 + \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \\ &\leq \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \end{aligned} \tag{20}$$

substitute in (19), we get

$$f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \leq \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 + \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \tag{21}$$

the rest step could follow the equation (16) in original paper,

$$\begin{aligned} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \ \leq& \frac{1}{2\eta}\|\mathbf{y}_t - \mathbf{u}_t\|_2^2 - \frac{1}{2\eta}\|\mathbf{y}_{t+1} - \mathbf{u}_t\|_2^2 + \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \\ =& \frac{1}{2\eta}\left(\|\mathbf{y}_t\|_2^2 - \|\mathbf{y}_{t+1}\|_2^2\right) + \frac{1}{\eta}\langle \mathbf{y}_{t+1} - \mathbf{y}_t, \mathbf{u}_t \rangle + \frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \end{aligned} \tag{22}$$

sum all,

$$\begin{aligned}\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) &\le \frac{1}{2\eta}\|\mathbf{y}_1\|_2^2 + \frac{1}{\eta}\sum_{t=1}^{T}\langle \mathbf{y}_{t+1} - \mathbf{y}_t, \mathbf{u}_t\rangle + \sum_{t=1}^{T}\frac{\eta}{2}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \\ &= \frac{1}{2\eta}\|\mathbf{y}_1\|_2^2 + \frac{1}{\eta}\left(\mathbf{y}_{T+1}^T\mathbf{u}_T - \mathbf{y}_1^T\mathbf{u}_1\right) + \frac{1}{\eta}\sum_{t=2}^{T}\langle \mathbf{u}_{t-1} - \mathbf{u}_t, \mathbf{y}_t\rangle \\ &\quad + \frac{\eta}{2}\sum_{t=1}^{T}\|\nabla \hat{f}_t(\mathbf{y}_{t+1}) - \nabla f_t(\mathbf{x}_t)\|_2^2 \\ &\le \frac{D^2}{\eta} + \frac{D}{\eta}\sum_{t=2}^{T}\|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2 + 2\eta T G^2\end{aligned} \tag{23}$$

where the last step makes of

$$\mathbf{y}_1 = \mathbf{0} \tag{24}$$

$$\mathbf{y}_{T+1}^T\mathbf{u}_T \le \|\mathbf{y}_{T+1}\|_2\|\mathbf{u}_T\| \le D^2 \tag{25}$$

$$\langle \mathbf{u}_{t-1} - \mathbf{u}_t, \mathbf{y}_t\rangle \le \|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2\|\mathbf{y}_t\|_2 \le D\|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2 \tag{26}$$

$$\|\nabla \hat{f}_t(\cdot)\|_2 \le G,\ \|\nabla f_t(\cdot)\|_2 \le G,\ \|\nabla \hat{f}_t(\cdot) - \nabla f_t(\cdot)\|_2 < 2G \tag{27}$$

□

3.3 Comparison

According to the last subsection, we derive the Regret with prediction is

$$\begin{aligned}\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) &\le \frac{D^2}{\eta} + \frac{D}{\eta}\sum_{t=2}^{T}\|\mathbf{u}_{t-1} - \mathbf{u}_t\|_2 + 2\eta T G^2 \\ &= \frac{D^2}{\eta} + \frac{D}{\eta}P_T + 2\eta T G^2\end{aligned} \tag{28}$$

by design the best stepsize as

$$\eta^*(P_T) = \sqrt{\frac{D^2 + DP_T}{2TG^2}} \tag{29}$$

the Regret reach

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \le \sqrt{2TG^2(D^2 + DP_T)} = \mathcal{O}\left(\sqrt{T(1 + P_T)}\right). \tag{30}$$

Compared with origin Regret at (22) in origin paper, i.e.

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \le \frac{7D^2}{4\eta} + \frac{D}{\eta}P_T + \frac{\eta T G^2}{2} = \mathcal{O}\left(\sqrt{T(1 + P_T)}\right) \tag{31}$$

even under the worst prediction, our Regret also reach the same order as $\mathcal{O}\left(\sqrt{T(1 + P_T)}\right)$.

If the prediction is perfect,i.e. $\nabla \hat{f}_t(\mathbf{y}_{t+1}) = \nabla f_t(\mathbf{x}_t)$, we get

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t) \le \frac{D^2}{\eta} + \frac{D}{\eta}P_T \tag{32}$$

which do not rely on T.

4 Empirical Exploration

We did a simulation based on Ader and Improved Ader algorithm in *Adaptive Online Learning in Dynamic Environments*. The two algorithms proved to have good performance in dynamic environment. During the simulation, we found that the performance of two algorithms are sensitive to the choice to G and D, which are estimated before the algorithm starts. We provided two solutions to this problem: increasing number of experts and using a dynamic set of experts.

4.1 Introduction to our simulation

The dataset we used is *Daily Temperature of Major Cities* in Kaggle. We did a temperature forecast based on temperature of previous days. Each day we make a prediction of tomorrow's temperature, and then we can receive the real temperature. The loss function is:

$$L(T_{predict}) = (T_{real} - T_{predict})^2$$

4.2 Comparasion between dynamic and statistic environments

We applied Ader and Improved Ader in different environments, and they proved to have good performance in dynamic environments, which is better than Online Mirror Descent.

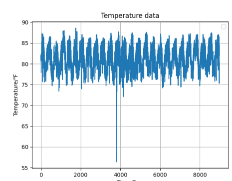

Figure 1: Temperatures of Cotonou, Benin.

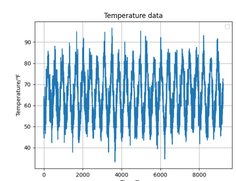

Figure 2: Temperatures of Algiers, Algeria.

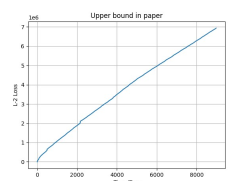

Figure 3: Upper bound of Cotonou, Benin.

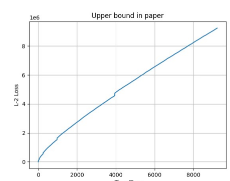

Figure 4: Upper bound of Algiers, Algeria.

Figure 1 and **Figure 2** are the temperatures of two cities. **Figure 1** has a more stationary environment while **Figure 2** has a more dynamic environment.

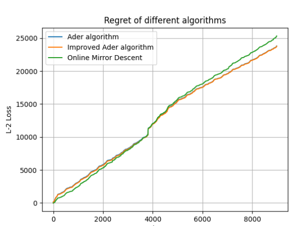

Figure 5: Regret of Cotonou, Benin.

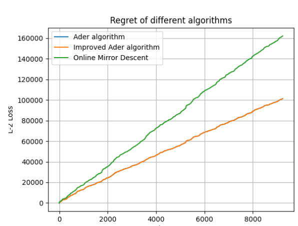

Figure 6: Regret of Algiers, Algeria.

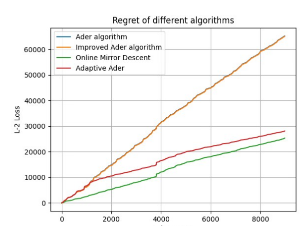

Figure 7: Regret of Ader with dynamic set of experts

Figure 5 and **Figure 6** are the regret curve of Ader, Improved Ader and Online Mirror Descent. The curve of Ader and Improved Ader are almost coincident. They proves to have a better performance than Online Mirror Descent in a more dynamic environment.

Figure 3 and **Figure 4** are the regret upper bound in paper:

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(u_t) \leq \frac{3G}{4}\sqrt{2T(7D^2 + 4DP_T)} + \frac{c\sqrt{2T}}{4}[1 + 2\ln(k+1)]$$

The difference of their upper bound are mainly determined by their path length, where $\frac{P_T}{T} = 1.23$, $D = 25.3$, $G = 50.6$ in **Figure 3** and $\frac{P_T}{T} = 2.34$, $D = 21.2$, $G = 42.4$ in **Figure 4**.

4.3 Sensitivity to the choice of G and D

In paper, it is assumed that the values of G and D are known to the learner. However, we can not acctually know the bound of the gradient of all functions and the bound of domain $\mathcal{X}$. We need to do an estimation of G and D. If our estimated $\frac{D}{G}$ is not accurate, the set of learning rate will be influenced: $\mathcal{H} = \{\eta_i = \frac{2^{i-1}D}{G}\sqrt{\frac{7}{2T}} | i = 1, \cdots, N\}$ Therefore, the best expert for this environment may not be included in the expert set.

In our simulation, we maintained D unchanged and increased the value of G(**Figure 8**, **Figure 9** and **Figure 10**). As the estimation of $\frac{D}{G}$ becomes more inaccurate, the regret increased.

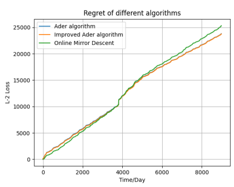

Figure 8: D=25.3 G=50.6 N=7

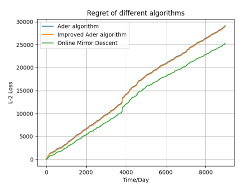

Figure 9: D=25.3 G=500 N=7

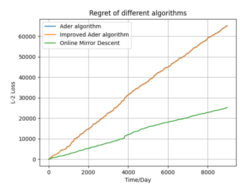

Figure 10: D=25.3 G=5000 N=7

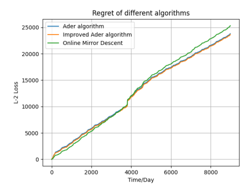

Figure 11: D=25.3 G=5000 N=12

Solution 1: Increase the number of experts If our estimation of $\frac{D}{G}$ is smaller than the real $\frac{D}{G}$, each learning rate is smaller. Therefore, we can increase N(the number of experts) to assure that the best expert is still in the expert set. In our simulation, we added N from 7 to 12 when $D = 25.3$, $G = 5000$. In this way, the regret is close to the regret with good estimation of $\frac{D}{G}$. (**Figure 11**)

However, this solution only works when the estimation of $\frac{D}{G}$ is smaller than the real $\frac{D}{G}$. If it is larger, the best expert's learning rate is smaller than any one in learning rate set.

Solution 2: A dynamic set of experts We found a solution to get good regret even with inaccurate estimation of D and G. We can use a dynamic set of experts, where we can add new experts and delete old experts in this algorithm. Our main idea is that if the best expert now has the highest/lowest learning rate, we can add a new expert with a higher/lower learning rate in the expert set, which may perform better in this environment. At the same time, we can delete an old expert to maintain the same time complexity with Ader algorithm.

Specifically, if the best expert has the highest learning rate, we set the new expert with double learning rate, its initial weight is $\frac{1}{N}$ and we assume that its prediction is same to the prediction of the best expert. Then we deleted the expert with lowest learning rate, and normalized the weight of each expert. If the best expert has the lowest learning rate, it is similar. Due to page limit, we leave the complete algorithm in the Appendix B. As shown in Figure 7, this algorithm is proved to have a good performance with inaccurate estimation of G and D.

5 Conclusion

In conclusion, this report has provided an in-depth review and analysis of the Adaptive Dynamic Expert Regret (Ader) algorithm as presented in the paper "Adaptive Online Learning in Dynamic Environments". We have delved into the theoretical aspects of the algorithm, examining its performance enhancement when an additional prediction is incorporated during the model updating process. We have also conducted empirical studies to understand the algorithm's performance under different dynamic environments. Our findings have highlighted the robustness of the Ader algorithm, while also revealing potential limitations when certain prior information about the environment is not available. To address these limitations, we have proposed several novel algorithms that dynamically adjust parameters based on environmental feedback. Through this comprehensive review and our explorations, we hope to contribute to the ongoing research in the field of online learning in dynamic environments, and inspire further improvements and innovations in this critical area.

References

[1] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. *Advances in neural information processing systems*, 2018.

[2] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, 2003.

A Appendix

A.1 Collaboration

This project is jointly completed by Zhitong Gao, Haotian Tian and Tianyue Zhou, with all team members playing an active role at different stages of the project. These include selecting the project paper, gaining an in-depth understanding of the selected work, exploring possible extensions of the study, preparing presentations, and writing the final report. Besides, each member undertook distinct individual responsibilities, detailed as follows:

- **Zhitong Gao: Literature Review and Report Organization**
  Zhitong Gao was entrusted with the comprehensive review of the project paper, encapsulating the problem statement, the proposed methodology, and possible extensions identified in the paper. Besides, she coordinated the structure and content of the project report, meticulously crafting the abstract, introduction, and conclusion sections. Zhitong also played a key role in critically analyzing the limitations of the method, suggesting novel algorithms to bolster its performance.
- **Haotian Tian: Theoretical Exploration and Predictive Analysis)**
  Haotian Tian explores the theoretical improvement of the original Ader by considering a prediction-enriched variant. Specifically, he offered a theoretical validation of the new algorithm's regret, demonstrating that even under worst-case prediction scenarios, the performance remained equivalent to the version without prediction, but offered tighter bounds given accurate predictions.
- **Tianyue Zhou: Empirical Exploration and Algorithm Improvement**
  Tianyue Zhou conducted empirical experiments to test the effectiveness of the paper's proposed algorithms, during which he identified certain drawbacks and devised two innovative solutions to overcome these limitations. Specifically, he proposed an advanced algorithm, that utilized a dynamic set of experts, improving upon the original Ader method.

B Appendix

Algorithm 2 Ader with dynamic set of experts: Meta algorithm

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

Activate a set of experts $\{E^\eta | \eta \in \mathcal{H}\}$ by invoking Expert Algorithm for each step size $\eta \in \mathcal{H}$

Sort step sizes in ascending order $\eta_1 \leq \eta_2 \leq \cdots \leq \eta_N$ and set $w_1^{\eta_i} = \frac{C}{i(i+1)}$

for $t = 1, ..., T$ **do**

  Receive x_t^η from each expert E^η

  Output

$$x_t = \sum_{\eta \in \mathcal{H}} w_t^\eta x_t^\eta$$

  Observe the loss function $f_t(\cdot)$

  Update the weight of each expert by

$$w_{t+1}^\eta = \frac{w_t^\eta e^{-\alpha f_t(x_t^\eta)}}{\sum_{\mu \in \mathcal{H}} w_t^\mu e^{-\alpha f_t(x_t^\mu)}}$$

  if $\text{argmax}_{i=1,\cdots,N} w_{t+1}^{\eta_i} = N$ **then**

    for $i = 1, \cdots, N-1$ **do**

      $\eta_i \leftarrow 2\eta_i$

      $w_{t+1}^{\eta_i} \leftarrow \frac{w_{i+1}}{\sum_{j=2}^N w_j + \frac{1}{N}}$

      $x_t^{\eta_i} \leftarrow x_t^{\eta_{i+1}}$

    $\eta_N \leftarrow 2\eta_N$

    $w_{t+1}^{\eta_N} \leftarrow \frac{\frac{1}{N}}{\sum_{j=2}^N w_j + \frac{1}{N}}$

    $x_t^{\eta_N} \leftarrow x_t^{\eta_N}$

  if $\text{argmax}_{i=1,\cdots,N} w_{t+1}^{\eta_i} = 1$ **then**

    for $i = 2, \cdots, N$ **do**

      $\eta_i \leftarrow \frac{1}{2}\eta_i$

      $w_{t+1}^{\eta_i} \leftarrow \frac{w_{i-1}}{\sum_{j=1}^{N-1} w_j + \frac{1}{N}}$

      $x_t^{\eta_i} \leftarrow x_t^{\eta_{i-1}}$

    $\eta_1 \leftarrow \frac{1}{2}\eta_1$

    $w_{t+1}^{\eta_1} \leftarrow \frac{\frac{1}{N}}{\sum_{j=1}^{N-1} w_j + \frac{1}{N}}$

    $x_t^{\eta_1} \leftarrow x_t^{\eta_1}$

  Send η, x_t^η and gradient $\nabla f_t(x_t^\eta)$ to each expert E_η

end for =0

Algorithm 3 Ader with dynamic set of experts: Expert-algorithm

Require: The step size η

Let x_t^η be any point in $\mathcal{X}$

for $t = 1, ..., T$ **do**

  Submit x_t^η to the meta-algorithm

  Receive η, x_t^η and gradient $\nabla f_t(x_t^\eta)$ from the meta-algorithm

$$x_{t+1}^\eta = \Pi_{\mathcal{X}}[x_t^\eta - \eta \nabla f_t(x_t^\eta)]$$

end for =0