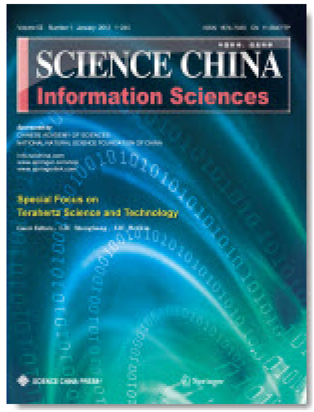


Strongly Adaptive Online Learning over Partial Intervals

Journal:	*SCIENCE CHINA Information Sciences*
Manuscript ID	SCIS-2020-0974.R2
Manuscript Type:	Research Paper
Date Submitted by the Author:	16-Apr-2021
Complete List of Authors:	Wan, Yuanyu; Nanjing University, Department of Computer Science and Technology Tu, Wei-Wei; 4Paradigm Inc. Zhang, LiJun; Nanjing University, Department of Computer Science and Technology
Keywords:	onling learning, strongly adaptive, changing environments, weighing method, regret bound
Speciality:	Artificial Intelligence\Pattern Recognition\Database and Knowledge-Based Systems < Computer Science & Technology

SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

Strongly Adaptive Online Learning over Partial Intervals

Yuanyu WAN[1], Wei-Wei TU[2] & Lijun ZHANG[1*]

[1]*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023*, China;*
[2]*4Paradigm Inc., Beijing* 100000*, China*

Abstract To cope with changing environments, strongly adaptive algorithms that almost enjoy the optimal performance on every time intervals have been proposed for online learning. However, the best regret bound of existing algorithms on each time interval with length τ is $O(\sqrt{\tau \log T})$, and their complexities are increasing with a factor of $O(\log T)$, where T is the time horizon. In real-world applications, T could go to infinity, which means that even these logarithmic factors are unacceptable. In this paper, we propose to remove the logarithmic factors of existing algorithms by utilizing prior information of environments. Specifically, we assume a lower bound τ_1 and an upper bound τ_2 on how long the environment changes are given, and only focus on the performance over time intervals with length in $[\tau_1, \tau_2]$. Then, we propose a new algorithm with a refined set of intervals that can reduce the complexity and a simple weighting method that can cooperate with our intervals set. Theoretical analysis reveals that the regret bound of our algorithm on any focused interval is optimal up to a constant factor. Both the regret bound and the computational cost per iteration are independent from T. Experimental results show that our algorithm outperforms the state-of-the-art algorithm.

Keywords onling learning, strongly adaptive, changing environments, weighing method, regret bound

Citation Wan Y Y, Tu W-W, Zhang L J. Strongly Adaptive Online Learning over Partial Intervals. Sci China Inf Sci, for review

1 Introduction

Online learning is a general framework that covers various problems such as learning with experts advice (LEA) [1] and online convex optimization (OCO) [2]. This framework can be viewed as a repeated game between a learner and an environment. In each round t, the learner picks a decision $\mathbf{x}_t \in \mathcal{X}$, where $\mathcal{X}$ is the decision space. Then, the environment reveals a loss function $f_t(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$ and the learner suffers a loss $f_t(\mathbf{x}_t)$. Traditionally, the performance of the learner is measured by so-called static regret with respect to the best fixed decision, which is defined as $\mathrm{R}(T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x})$. However, environments of real-world applications may change, and the best decision for different time intervals could be different. In this case, static regret can no longer reflect the hardness of the problem [3].

Recently, strongly adaptive regret [4] has been proposed to measure the performance of the learner on every time intervals, which is defined as

$$\mathrm{SAR}(T,\tau) = \max_{[s,s+\tau-1]\subseteq[T]} \left(\sum_{t=s}^{s+\tau-1} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=s}^{s+\tau-1} f_t(\mathbf{x}) \right) \tag{1}$$

for intervals of length τ, where $[T] = \{1, \cdots, T\}$. Accordingly, strongly adaptive algorithms [4, 5] have been developed to minimize the strongly adaptive regret $\mathrm{SAR}(T,\tau)$ for any τ. Unlike traditional algorithms [6–17] being designed for a specific problem, these algorithms are meta-algorithms that can use any online algorithm as a black-box and convert it into a strongly adaptive one. Specifically, these meta-algorithms consist of a set of intervals, each of which is associated with an instance of the black-box,

* Corresponding author (email: zhanglj@lamda.nju.edu.cn)

and a weighting method that aggregates decisions from the active instances. With an appropriate set of intervals, in each round t, these meta-algorithms maintain $O(\log t)$ instances of the black-box. So the complexities of these strongly adaptive algorithms are increasing with a factor of $O(\log t)$. Moreover, the strongly adaptive regret can be decomposed as the sum of the meta regret caused by meta-algorithm and the black-box regret. While the black-box regret could be bounded by $O(\sqrt{\tau})$ for any length τ, the best meta regret bound is $O(\sqrt{\tau \log T})$ established by Jun et al. [5], which has an additional factor $\sqrt{\log T}$.

Because of the ability to cope with changing environments, strongly adaptive algorithms are more appropriate for real-world applications than traditional online algorithms. However, in many real-world applications, the scale of data grows continuously and explosively, which means even logarithmic factors $O(\sqrt{\log T})$ and $O(\log T)$ cannot be ignored. Therefore, their increasing complexities and the gap between their bounds and the optimal one are unacceptable, which significantly limits their applications. To tackle this limitation, this paper aims to improve strongly adaptive algorithms by utilizing prior information of environments. In many applications, the occurrence of environment changing is related to other regular events, and is knowable to the learner. For example, in moving tag detection [18, 19], the sensors used to collect data are regularly expired and replaced by new ones, which causes that the environment changes regularly. In recommender systems, the environment changing could be mainly caused by the change of the purchasing behaviors of customers. According to previous studies [20–22], the purchasing behaviors of customers could change regularly under the impact of their life stages. Without loss of generality, we assume a lower bound τ_1 and an upper bound τ_2 on how long the environment changes are given as the prior information of applications. Our proposed algorithm only focuses on the performance over time intervals with length in $[\tau_1, \tau_2]$, instead of every intervals.

Specifically, by utilizing this prior information, we propose a new meta-algorithm for strongly adaptive online learning, which consists of two parts:

- A refined set of intervals, which is carefully designed to reduce the number of the instances;
- A simple weighting method, which can cooperate with our refined set of intervals.

Compared with existing meta-algorithms, we only maintain $O(\log(\tau_2/\tau_1))$ instead of $O(\log t)$ instances of the black-box in each round t, and reduce the meta regret bound from $O(\sqrt{\tau \log T})$ to $O(\sqrt{\tau \log(\tau_2/\tau_1)})$ for any focused interval with length τ. Combining with appropriate black-box, we establish the following bounds:

- $\mathrm{SAR}(T,\tau) = O(\sqrt{\tau \log(\tau_2/\tau_1)} + \sqrt{\tau \log N})$ for LEA where N is the number of experts, which is better than $O(\sqrt{\tau \log T} + \sqrt{\tau \log N})$ of the previous work [5];
- $\mathrm{SAR}(T,\tau) = O(\sqrt{\tau \log(\tau_2/\tau_1)} + GD\sqrt{\tau})$ for OCO where D is diameter of $\mathcal{X}$ and G is bound of any $\|\nabla f_t(\mathbf{x})\|_2$, which is better than $O(\sqrt{\tau \log T} + GD\sqrt{\tau})$ of the previous work [5].

Moreover, our meta regret and strongly adaptive regret for LEA also have problem-dependent bounds, which could be much tighter when the loss of the competitor is small. Similarly, when the loss functions are smooth, we can further improve our strongly adaptive regret bound for OCO to a problem-dependent one. To verify the efficiency and effectiveness of our algorithm, we conduct numerical experiments on LEA and OCO, respectively. The results demonstrate that our algorithm outperforms the state-of-the-art algorithm.

2 Related Work

In this section, we only review related work in strongly adaptive regret for brevity. More related work in static regret can be found in surveys of online learning [23–25].

To measure the performance of learner in changing environments, the pioneering work [26] proposed adaptive regret, which is an extension of static regret and defined as

$$\mathrm{AR}(T) = \max_{[q,s]\subseteq[T]} \left(\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{X}} \sum_{t=q}^{s} f_t(\mathbf{x}) \right) \tag{2}$$

where $[T] = \{1, \cdots, T\}$. Accordingly, Hazan and Seshadhri [26] proposed two meta-algorithms named as follow the leading history (FLH) with $O(T)$ complexity and advanced follow the leading history (AFLH) with $O(\log T)$ complexity to minimize the adaptive regret $\mathrm{AR}(T)$. For general convex functions, with approximate black-box, FLH and AFLH have adaptive regret bounds $O(\sqrt{T \log T})$ and $O(\sqrt{T \log^3 T})$,

Table 1 Comparison of strongly adaptive algorithms for LEA and general OCO, where previous regret bounds hold for any $\tau \in [T]$ and our bounds hold for $\tau \in [\tau_1, \tau_2]$.

Reference	$\mathrm{SAR}(T,\tau)$ for LEA	$\mathrm{SAR}(T,\tau)$ for OCO	Number of the instances
[4]	$O(\sqrt{\tau}\log T + \sqrt{\tau \log N})$	$O(\sqrt{\tau}\log T + GD\sqrt{\tau})$	$O(\log t)$
[5,34]	$O(\sqrt{\tau \log T} + \sqrt{\tau \log N})$	$O(\sqrt{\tau \log T} + GD\sqrt{\tau})$	$O(\log t)$
[37]	$O(\sqrt{\tau \log T} + \sqrt{\tau \log N})$	$O(\sqrt{\tau \log T} + GD\sqrt{\tau})$	$O(\log^2 t)$
[38]	$O(\sqrt{\tau \log T} + \sqrt{\tau \log N})$	$O(\sqrt{\tau \log T} + GD\sqrt{\tau})$	$O(\log T), O(\log^2 t)$
This work	$O(\sqrt{\tau \log(\tau_2/\tau_1)} + \sqrt{\tau \log N})$	$O(\sqrt{\tau \log(\tau_2/\tau_1)} + GD\sqrt{\tau})$	$O(\log(\tau_2/\tau_1))$

respectively. Note that these bounds depend on T instead of the length of intervals, which make no sense for intervals with small length such as $O(\sqrt{T})$.

To overcome this limitation, Daniely et al. [4] proposed strongly adaptive regret $\mathrm{SAR}(T,\tau)$ defined in (1) and argued that an algorithm is strongly adaptive if for every environments, it has $\mathrm{SAR}(T,\tau) = O(\mathrm{poly}(\log T)\,\mathrm{R}(\tau))$, where $\mathrm{R}(\tau)$ is the minimax regret bound for time intervals with length τ and $\mathrm{R}(\tau) = O(\sqrt{\tau})$ for general convex functions [27]. Compared with adaptive regret, strongly adaptive regret is a refined measure, because it emphasizes the dependency on the interval length, which is meaningful even for intervals with small length. For general convex functions, Daniely et al. [4] proposed the first strongly adaptive meta-algorithm and established a meta regret bound as $O(\sqrt{\tau}\log T)$. The two key parts of the meta-algorithm are:

- The geometric covering (GC) intervals defined as $\mathcal{J} = \bigcup_{j \in \mathbb{N} \cup \{0\}} \mathcal{J}_j$, where $\mathcal{J}_j = \{[i \cdot 2^j, (i+1) \cdot 2^j - 1] : i \in \mathbb{N}\}$;
- The weighting method, which is an extension of multiplicative weights (MW) [28] in the sleeping expert setting [29].

According to the definition and illustration of $\mathcal{J}$ shown in Figure 1, it is easy to verify that any time t is contained by at most $O(\log t)$ intervals, which is equal to the number of instances of the black-box. By respectively choosing MW and online gradient descent [2] as the black-box, Daniely et al. [4] established $\mathrm{SAR}(T,\tau) = O(\sqrt{\tau}\log T + \sqrt{\tau \log N})$ for LEA and $\mathrm{SAR}(T,\tau) = O(\sqrt{\tau}\log T + GD\sqrt{\tau})$ for OCO. Later, Jun et al. [5] proposed a new meta-algorithm named as coin betting for changing environment (CBCE) by replacing MW with coin betting (CB) [30], which reduces the meta regret bound to $O(\sqrt{\tau \log T})$ and could accordingly improve the strongly adaptive regret bound for both LEA and OCO. Recently, Zhang et al. [31] utilized the smoothness of loss functions to improve the strongly adaptive regret bound for OCO to a problem-dependent bound by choosing AdaNormalHedge [32] and scale-free online gradient descent (SOGD) [33] as the weighting method and the black-box, respectively.

However, the number of instances maintained by all the previous methods increases at least as $O(\log t)$, which is unacceptable, especially for real-world applications where T could go to infinity. To accelerate these algorithms, Wang et al. [34] proposed a series of algorithms that reduce the number of gradient evaluation from $O(\log t)$ to 1 by carefully designing surrogate losses [35]. Although their algorithms are much more efficient than previous strongly adaptive algorithms when the evaluation of gradients is expensive, they only partially overcame the limitation of complexity because the number of the instances is still $O(\log t)$. Moreover, for general convex functions, the factor $\sqrt{\log T}$ in the strongly adaptive regret bounds of previous algorithms [5, 34] also limits their applications.

We also note that strongly adaptive algorithms for exponentially concave and strongly convex functions have been proposed by Hazan and Seshadhri [26] and Zhang et al. [36] respectively. Recently, Zhang et al. [37] further proposed a universal algorithm to minimize the strongly adaptive regret for different types of convex functions. Although their algorithm enjoys the same strongly adaptive regret as that of CBCE [5], it needs to maintain $O(\log^2 t)$ instances in each round t. Moreover, Zhang et al. [38] have proposed two algorithms to simultaneously minimize the strongly adaptive regret and dynamic regret, where the latter is another performance measure for changing environments [2]. However, in each round t, the first algorithm needs to maintain $O(\log T)$ instances, and the second algorithm needs to maintain $O(\log^2 t)$ instances. In this paper, we focus on general convex functions and the strongly adaptive regret. To facilitate comparisons, the strongly adaptive regret and the computational complexity of different strongly adaptive algorithms for LEA and general OCO are summarized in Table 1.

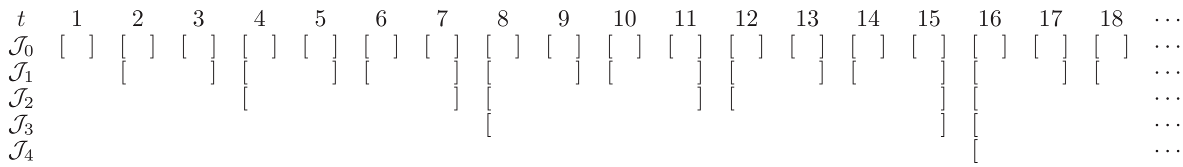


Figure 1 Illustration of GC intervals, where each interval is denoted by [ ].

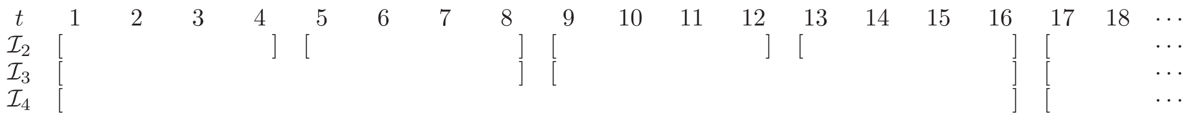


Figure 2 The refined set of intervals where $\tau_1 = 4$ and $\tau_2 = 16$.

3 Main Results

In this section, we present our algorithm for changing environments and the corresponding theoretical results.

3.1 Algorithms

From previous studies [4,5], we know that strongly adaptive algorithms are composed of a set of intervals, which decides the starting and ending time of instances of the black-box, and a weighting method that weights these instances according to their performance in the history. To ensure the performance on time intervals with length in $[\tau_1, \tau_2]$, we propose a new strongly adaptive algorithm with a refined set of intervals and a simple weighting method, as explained below.

The Set of Intervals $\mathcal{I}$. To ensure the optimal performance on every intervals, the key property of GC intervals $\mathcal{J}$ is that any interval can be partitioned into a finite sequences of disjoint and consecutive intervals in $\mathcal{J}$ (Lemma 5 of Daniely et al. [4]). Because we only focus on intervals with length in $[\tau_1, \tau_2]$, it is reasonable to remove unnecessary intervals from $\mathcal{J}$ while keep a similar property for focused intervals. Specifically, we define a smaller set of intervals

$$\mathcal{I} = \bigcup_{j=\lceil \log \tau_1 \rceil, \cdots, \lceil \log \tau_2 \rceil} \mathcal{I}_j \tag{3}$$

where $\mathcal{I}_j = \left\{ [i \cdot 2^j + 1, (i+1) \cdot 2^j] : i \in \mathbb{N} \cup \{0\} \right\}$.

Comparing GC intervals $\mathcal{J}$ with our $\mathcal{I}$, the most obvious difference is that the length of intervals in $\mathcal{I}$ is bounded in $\left[2^{\lceil \log \tau_1 \rceil}, 2^{\lceil \log \tau_2 \rceil}\right]$, instead of being unbounded in $\mathcal{J}$. Furthermore, because the absence of intervals shorter than τ_1 affects the partition of intervals, our $\mathcal{I}$ only ensures that any focused interval can be contained by two disjoint and consecutive intervals in it. Figure 2 gives an illustration of our $\mathcal{I}$ with $\tau_1 = 4, \tau_2 = 16$. We maintain an instance $\mathcal{B}_I$ of the black-box $\mathcal{B}$ during each time interval $I \in \mathcal{I}$ and define the active set at time t as

$$\text{Active}(t) = \{I \in \mathcal{I} : t \in I\}. \tag{4}$$

In each round $t = 1, \cdots, T$, each instance $\mathcal{B}_I, \forall I \in \text{Active}(t)$ is working to generate a decision. To aggregate decisions from active instances, we regard these instances as experts and utilize appropriate methods to weight these experts.

The Weighting Method. To cooperate with GC intervals, AdaNormalHedge [32], CB [30] and MW [28] have been extended to the sleeping expert setting by Zhang et al. [31], Jun et al. [5] and Daniely et al. [4], respectively. Comparing these methods, we prefer to choose AdaNormalHedge to cooperate with our refined set of intervals $\mathcal{I}$, because it achieves a problem-dependent regret bound which could be used to establish a problem-dependent bound for strongly adaptive regret and is more tighter than the regret bounds of MW and CB that are related to the length of interval, when the loss of the competitor is small.

Algorithm 1 Modified AdaNormalHedge

Input: Active interval $I=[q,s]$, number of experts N
for $t=q,\cdots,s$ **do**
  $\mathbf{c}_t^I(i)=W\left(\sum_{k=q}^{t-1}\tilde{\mathbf{g}}_k^I(i),\sum_{k=q}^{t-1}|\tilde{\mathbf{g}}_k^I(i)|\right)$ and $\mathbf{x}_t^I(i)\propto[\mathbf{c}_t^I(i)]_+,\forall i\in[N]$
  Receive loss vector $\boldsymbol{\ell}_t\in[0,1]^N$
  $\forall i\in[N]$, $\tilde{\mathbf{g}}_t^I(i)=\begin{cases}\langle\boldsymbol{\ell}_t,\mathbf{x}_t^I\rangle-\boldsymbol{\ell}_t(i),\mathbf{c}_t^I(i)>0\\ [\langle\boldsymbol{\ell}_t,\mathbf{x}_t^I\rangle-\boldsymbol{\ell}_t(i)]_+,\mathbf{c}_t^I(i)\leqslant 0\end{cases}$
end for

Let N be the number of experts, and $\boldsymbol{\ell}_t(i)\in[0,1]$ be the loss of expert i in the t-th round. AdaNormalHedge is mainly composed of the potential function defined as

$$\Phi(x,y)=\exp\left(\frac{[x]_+^2}{3y}\right) \tag{5}$$

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

$$W(x,y)=\frac{1}{2}\left(\Phi(x+1,y+1)-\Phi(x-1,y+1)\right). \tag{6}$$

In the t-th round, AdaNormalHedge sets and normalizes the weight of expert i as

$$\mathbf{x}_t(i)\propto\mathbf{c}_t(i)=W\left(G_{t-1}^i,S_{t-1}^i\right) \tag{7}$$

where $G_{t-1}^i=\sum_{k=1}^{t-1}(\langle\boldsymbol{\ell}_k,\mathbf{x}_k\rangle-\boldsymbol{\ell}_k(i))$ is the regret with respect to expert i over the first $t-1$ iterations and $S_{t-1}^i=\sum_{k=1}^{t-1}|\langle\boldsymbol{\ell}_k,\mathbf{x}_k\rangle-\boldsymbol{\ell}_k(i)|$ is the cumulative magnitude of the instantaneous regret over the first $t-1$ iterations. For brevity, we define $\tilde{\mathbf{g}}_t(i)=\langle\boldsymbol{\ell}_t,\mathbf{x}_t\rangle-\boldsymbol{\ell}_t(i)$.

According to the definition of potential function (5), it is easy to derive an upper bound of G_t^i as $G_t^i\leqslant\sqrt{3S_t^i\ln\Phi(G_t^i,S_t^i)}$. However, because our $\mathcal{I}$ only ensures that any focused interval can be contained by two disjoint and consecutive intervals in it, to cooperate with $\mathcal{I}$, we need to bound the absolute value of G_t^i. To this end, we redefine the potential function (5) with slight modifications as

$$\Phi(x,y)=\exp\left(\frac{x^2}{3y}\right) \tag{8}$$

where $\Phi(0,0)=1$ and $|G_t^i|=\sqrt{3S_t^i\ln\Phi(G_t^i,S_t^i)}$. The weight function with respect to the new potential function is still defined as (6) and the weight of each expert i is still set as $\mathbf{c}_t(i)=W\left(G_{t-1}^i,S_{t-1}^i\right)$. However, with the new potential function, the value of $\mathbf{c}_t(i)$ could be negative, which motivates the following two modifications. First, to ensure $\mathbf{x}_t\in\Delta^N$ where $\Delta^N=\{\mathbf{x}:\sum_{i=1}^N\mathbf{x}(i)=1\}$, the normalized weight is redefined as $\mathbf{x}_t(i)\propto[\mathbf{c}_t(i)]_+$. Second, to ensure $\sum_{i=1}^N\tilde{\mathbf{g}}_t(i)\mathbf{c}_t(i)\leqslant 0$ that is essential for upper bounding $\Phi(G_t^i,S_t^i)$, we redefine

$$\tilde{\mathbf{g}}_t(i)=\begin{cases}\langle\boldsymbol{\ell}_t,\mathbf{x}_t\rangle-\boldsymbol{\ell}_t(i),\mathbf{c}_t(i)>0\\ [\langle\boldsymbol{\ell}_t,\mathbf{x}_t\rangle-\boldsymbol{\ell}_t(i)]_+,\mathbf{c}_t(i)\leqslant 0\end{cases} \tag{9}$$

and recall $G_{t-1}^i=\sum_{k=1}^{t-1}\tilde{\mathbf{g}}_k(i),S_{t-1}^i=\sum_{k=1}^{t-1}|\tilde{\mathbf{g}}_k(i)|$. We call this new algorithm as modified AdaNormalHedge and summarize its detailed procedures in Algorithm 1, where the superscript I is used to distinguish its instances on different intervals.

Based on the modified AdaNormalHedge, we proposed our weighting method that can aggregate decisions from active instances of the black-box. The potential function and the corresponding weight function are still defined as (8) and (6), respectively. To calculate the weight w_t^I of each decision $\mathbf{x}_t^I$ generated by instance $\mathcal{B}_I$, we further define

$$R_t^I=\sum_{k=1}^t\mathbb{I}_{[k\in I]}\tilde{r}_k^I,C_t^I=\sum_{k=1}^t\mathbb{I}_{[k\in I]}|\tilde{r}_k^I|,\tilde{r}_k^I=\begin{cases}f_k(\mathbf{x}_k)-f_k(\mathbf{x}_k^I),w_k^I>0\\ [f_k(\mathbf{x}_k)-f_k(\mathbf{x}_k^I)]_+,w_k^I\leqslant 0\end{cases} \tag{10}$$

Algorithm 2 Strongly Adaptive Online Learning over Partial Intervals

Input: A black-box algorithm $\mathcal{B}$, prior information τ_1, τ_2
$\mathcal{I} = \bigcup_j \mathcal{I}_j, j = \lceil \log \tau_1 \rceil, \lceil \log \tau_1 \rceil + 1, \cdots, \lceil \log \tau_2 \rceil$
for $t = 1, 2, \cdots, T$ **do**
  $w_t^I = W\left(\sum_{k=1}^{t-1} \mathbb{I}_{[k \in I]} \tilde{r}_k^I, \sum_{k=1}^{t-1} \mathbb{I}_{[k \in I]} |\tilde{r}_k^I|\right)$ and $p_t^I \propto [w_t^I]_+$, $\forall I \in \text{Active}(t)$
  Run $\mathcal{B}_I$ to generate $\mathbf{x}_t^I \in \mathcal{X}$, $\forall I \in \text{Active}(t)$
  $\mathbf{x}_t = \sum_{I \in \text{Active}(t)} p_t^I \mathbf{x}_t^I$
  Observe the loss $f_t(\mathbf{x}_t^I) \in [0, 1]$, $\forall I \in \text{Active}(t)$ and $f_t(\mathbf{x}_t)$
  $\forall I \in \text{Active}(t)$,
$$\tilde{r}_t^I = \begin{cases} f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^I), w_t^I > 0 \\ [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^I)]_+, w_t^I \leqslant 0 \end{cases}$$
end for

Algorithm 3 Scale-free Online Gradient Descent

Input: Active interval $I = [q, s]$ and parameters δ, α
Initialize $\mathbf{x}_q^I \in \mathcal{X}$ arbitrarily
for $t = q, \cdots, s$ **do**
  $\eta_t^I = \alpha / \sqrt{\delta + \sum_{i=q}^{t} \|\nabla f_i(\mathbf{x}_i^I)\|_2^2}$
  $\mathbf{x}_{t+1}^I = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} \|\mathbf{x} - (\mathbf{x}_t^I - \eta_t^I \nabla f_t(\mathbf{x}_t^I))\|_2^2$
end for

for $I \in \mathcal{I}$, where $\mathbb{I}_{[k \in I]} = 1$ if $k \in I$, and $\mathbb{I}_{[k \in I]} = 0$ if $k \notin I$. Then, in each round $t, \forall I \in \text{Active}(t)$, the weight is calculated as $w_t^I = W(R_{t-1}^I, C_{t-1}^I)$ and normalized as $p_t^I \propto [w_t^I]_+$. Finally, the decision of meta-algorithm is calculated as
$$\mathbf{x}_t = \sum_{I \in \text{Active}(t)} p_t^I \mathbf{x}_t^I. \tag{11}$$
The detailed procedures of our meta-algorithm are summarized in Algorithm 2 and this algorithm is named as strongly adaptive online learning over partial intervals (SAOL-PI).

Remark. With our refined $\mathcal{I}$, any time t is contained in only one interval in each $\mathcal{I}_j$, where $j = \lceil \log \tau_1 \rceil, \lceil \log \tau_1 \rceil + 1, \cdots, \lceil \log \tau_2 \rceil$. As a result, our Algorithm 2 only needs to maintain $O(\log(\tau_2/\tau_1))$ instances in each round t, which is better than the increasing factor $O(\log t)$ of previous algorithms. Even if we do not get accurate prior information in some cases, we can simply set $\tau_1 = 1$ and $\tau_2 = T$ in Algorithm 2, and the complexity reduces to that of previous algorithms.

3.2 General Regret Bounds

Following previous studies [4, 5], we introduce the following assumption.

Assumption 1. The loss function f_t is general convex and $0 \leqslant f_t(\mathbf{x}) \leqslant 1$, for any t.

We first bound the meta regret of our Algorithm 2 with respect to an instance $\mathcal{B}_I$ as below.

Theorem 1. (General Meta Regret Bound) Under Assumption 1, for any $I \in \mathcal{I}$ and $[q, s] \subseteq I$, Algorithm 2 with the black-box $\mathcal{B}$ satisfies
$$\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}_t^I) \leqslant 2\sqrt{c|I|} \tag{12}$$
where $c = 3\ln(2\tau_2(3 + \ln(1 + 2\tau_2))/\tau_1)$.

Compared with the meta regret bound $O(\sqrt{|I| \log T})$ of Jun et al. [5], our improved bound in Theorem 1 is independent from T and can yield a number of improvements for different problems by choosing an appropriate black-box for our Algorithm 2.

For brevity, we take LEA and OCO as examples. First, for LEA where $f_t(\mathbf{x}) = \langle \boldsymbol{\ell}_t, \mathbf{x} \rangle$ and $\mathcal{X} = \Delta^N$, we choose the modified AdaNormalHedge shown in Algorithm 1 as the black-box of Algorithm 2. Before establishing a regret bound for LEA, we bound the regret of Algorithm 1 in the following lemma.

Lemma 1. For any $I \in \mathcal{I}$, $[q,s] \subseteq I$ and $\mathbf{x} \in \Delta^N$, Algorithm 1 satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant 2\sqrt{\tilde{c}(|I|)|I|} \tag{13}$$

where $\tilde{c}(x) = 3\ln(N(3+\ln(1+x))/2)$.

Then, combining Theorem 1 and Lemma 1, we can obtain the following corollary.

Corollary 1. Let $c = 3\ln(2\tau_2(3+\ln(1+2\tau_2))/\tau_1)$ and $\tilde{c}(x) = 3\ln(N(3+\ln(1+x))/2)$. Under the setting of LEA, for any $\mathbf{x} \in \mathcal{X}$ and $I = [q,s]$ with length $|I| \in [\tau_1, \tau_2]$, our Algorithm 2 using Algorithm 1 as its black-box satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant 4\sqrt{2|I|c} + 4\sqrt{2|I|\tilde{c}(2|I|)}. \tag{14}$$

Second, for OCO, we choose SOGD [33] shown in Algorithm 3 as the black-box of Algorithm 2 and further introduce the following two common assumptions.

Assumption 2. The domain $\mathcal{X}$ is convex and $\|\mathbf{x}-\mathbf{y}\|_2 \leqslant D$ for any $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{X}$.

Assumption 3. The gradient satisfies $\|\nabla f_t(\mathbf{x})\|_2 \leqslant G$ for any $\mathbf{x} \in \mathcal{X}$ and t.

Under Assumptions 1, 2 and 3, we bound the regret of SOGD in the following lemma.

Lemma 2. Under Assumptions 1, 2 and 3, for any $I \in \mathcal{I}$, $[q,s] \subseteq I$ and $\mathbf{x} \in \mathcal{X}$, Algorithm 3 with $\delta > 0, \alpha = D/\sqrt{2}$ satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - f_t(\mathbf{x}) \leqslant \sqrt{2}D\sqrt{\delta + G^2|I|}. \tag{15}$$

Then, combining Theorem 1 and Lemma 2, we can obtain the following corollary.

Corollary 2. Let $c = 3\ln(2\tau_2(3+\ln(1+2\tau_2))/\tau_1)$. Under Assumptions 1, 2 and 3, for any $\mathbf{x} \in \mathcal{X}$ and $I = [q,s]$ with length $|I| \in [\tau_1, \tau_2]$, our Algorithm 2 using Algorithm 3 with $\delta > 0$ and $\alpha = D/\sqrt{2}$ as its black-box satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant 4\sqrt{2|I|c} + 2\sqrt{2}D\sqrt{\delta + 2G^2|I|}. \tag{16}$$

Note that the parameter $\delta > 0$ is used to avoid the operation of being divided by 0. So it is reasonable to choose a small value such as $\delta = 10^{-4}$ that does not affect the regret bound in Corollary 2.

Remark. Corollary 1 shows that our Algorithm 2 with the modified AdaNormalHedge is a strongly adaptive algorithm for LEA, and enjoys

$$\mathrm{SAR}(T,\tau) \leqslant O(\sqrt{\tau\log(\tau_2/\tau_1)} + \sqrt{\tau\log N}) \tag{17}$$

for $\tau \in [\tau_1, \tau_2]$. Corollary 2 shows that our Algorithm 2 with SOGD is a strongly adaptive algorithm for OCO, and enjoys

$$\mathrm{SAR}(T,\tau) \leqslant O(\sqrt{\tau\log(\tau_2/\tau_1)} + GD\sqrt{\tau}) \tag{18}$$

for $\tau \in [\tau_1, \tau_2]$. Compared with the bounds of Jun et al. [5], our bounds are better when $\log(\tau_2/\tau_1)$ is a small constant. When there is no prior knowledge, we set $\tau_1 = 1$ and $\tau_2 = T$, then the $\log(\tau_2/\tau_1)$ term reduces to $\log T$ and we recover their bounds.

3.3 Problem-dependent Regret Bounds

It is worth mentioning that the meta regret of our Algorithm 2 also enjoys a problem-dependent bound in the following theorem.

Theorem 2. (Problem-dependent Meta Regret Bound) Under Assumption 1, for any $I \in \mathcal{I}$ and $[q,s] \subseteq I$, Algorithm 2 with the black-box $\mathcal{B}$ satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}_t^I) \leqslant 2c + 2\sqrt{2c\sum_{t=1}^{s} \mathbb{I}_{[t\in I]} f_t(\mathbf{x}_t^I)} \tag{19}$$

where $c = 3\ln(2\tau_2(3+\ln(1+2\tau_2))/\tau_1)$.

Because of $\sum_{t=1}^{s} \mathbb{I}_{[t\in I]} f_t(\mathbf{x}_t^I) \leqslant |I|$, the upper bound in Theorem 2 is comparable with that in Theorem 1 in the worst case and could be much smaller when the loss of the competitor is small. According to Theorem 2, we can improve the upper bounds in Corollary 1 and Corollary 2 to problem-dependent ones.

Specifically, for LEA, with more careful analysis, we can improve the upper bound in Lemma 1 to the following problem-dependent bound.

Lemma 3. For any $I \in \mathcal{I}$, $[q,s] \subseteq I$ and $\mathbf{x} \in \Delta^N$, Algorithm 1 satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant 2\tilde{c}(|I|) + 2\sqrt{2\tilde{c}(|I|)\sum_{t=1}^{s} \mathbb{I}_{[t\in I]} f_t(\mathbf{x})} \tag{20}$$

where $\tilde{c}(x) = 3\ln(N(3+\ln(1+x))/2)$.

Combining (19) with (20), we further obtain the following regret bound without introducing any additional assumption.

Corollary 3. Let $c = 3\ln(2\tau_2(3+\ln(1+2\tau_2))/\tau_1)$ and $\tilde{c}(x) = 3\ln(N(3+\ln(1+x))/2)$. Under the setting of LEA, for any $\mathbf{x} \in \Delta^N$ and $I = [q,s]$ with length $|I| \in [\tau_1, \tau_2]$, our Algorithm 2 using Algorithm 1 as its black-box satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant a(I) + b(I)\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}. \tag{21}$$

where $q' = \lfloor \frac{q-1}{2^j} \rfloor \cdot 2^j + 1$, $j = \lceil \log |I| \rceil$, $a(I) = 4c + 8\sqrt{2c\tilde{c}(2^j)} + 4\tilde{c}(2^j)$ and $b(I) = 4\sqrt{2c} + 4\sqrt{\tilde{c}(2^j)}$.

Remark. We first note that $a(I) = O(\log(\tau_2/\tau_1) + \log N)$ and $b(I) = O(\sqrt{\log(\tau_2/\tau_1)} + \sqrt{\log N})$ where we treat the double logarithmic factors as constant following [32]. Therefore, the upper bound in Corollary 3 is on the order of

$$O\left((\sqrt{\log(\tau_2/\tau_1)} + \sqrt{\log N})\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})} \right). \tag{22}$$

Because of $s - q' + 1 \leqslant 2^{j+1} \leqslant 4|I|$, we have $\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})} \leqslant \sqrt{\sum_{t=q'}^{s} 1} = O(\sqrt{|I|})$, which means that the above upper bound is comparable to the upper bound in Corollary 1 in the worst case. Moreover, when the loss of the competitor is small, $\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}$, which is a relaxation of $\sqrt{\sum_{t=q}^{s} f_t(\mathbf{x})}$, could be much smaller than $O(\sqrt{|I|})$ in Corollary 1.

To achieve a problem-dependent regret bound for OCO, inspired by previous studies [31,39], we further introduce the following assumption about the smoothness of the loss functions.

Assumption 4. For any t, the loss function f_t is H-smooth, that is, for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$

$$\|\nabla f_t(\mathbf{x}) - \nabla f_t(\mathbf{x}')\|_2 \leqslant H\|\mathbf{x} - \mathbf{x}'\|_2. \tag{23}$$

Then, we bound the regret of Algorithm 3 in the following lemma.

Lemma 4. Under Assumptions 1, 2 and 4, for any $I \in \mathcal{I}$, $[q,s] \subseteq I$ and $\mathbf{x} \in \mathcal{X}$, Algorithm 3 with $\delta > 0, \alpha = D/\sqrt{2}$ satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant 8HD^2 + D\sqrt{2\delta + 8H\sum_{t=1}^{s} \mathbb{I}_{[t\in I]} f_t(\mathbf{x})}. \tag{24}$$

Combining (19) with (24), we can obtain the following regret bound.

Corollary 4. Let $c = 3\ln(2\tau_2(3+\ln(1+2\tau_2))/\tau_1)$. Under Assumptions 1, 2 and 4, for any $\mathbf{x} \in \mathcal{X}$ and $I = [q, s]$ with length $|I| \in [\tau_1, \tau_2]$, our Algorithm 2 using Algorithm 3 with $\delta > 0, \alpha = D/\sqrt{2}$ as its black-box satisfies

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant \tilde{a}(I) + \tilde{b}(I)\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}. \tag{25}$$

where $q' = \lfloor \frac{q-1}{2^j} \rfloor \cdot 2^j + 1$, $j = \lceil \log |I| \rceil$, $\tilde{a}(I) = 6c + 56HD^2 + 6D\sqrt{2\delta}$ and $\tilde{b}(I) = 4\sqrt{2c} + 4\sqrt{HD^2}$.

Remark. Similar as $a(I)$ and $b(I)$ in Corollary 3, we note that $\tilde{a}(I) = O(\log(\tau_2/\tau_1) + HD^2)$ and $\tilde{b}(I) = O(\sqrt{\log(\tau_2/\tau_1)} + \sqrt{HD^2})$. Therefore, the upper bound in Corollary 4 is on the order of

$$O\left((\sqrt{\log(\tau_2/\tau_1)} + \sqrt{HD^2}) \sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})} \right) \tag{26}$$

which is comparable to the upper bound in Corollary 2 in the worst case and could be much tighter when the loss of the competitor is small.

4 Theoretical Analysis

Due to the limitation of space, we only provide the proof of Theorems 1 and 2, and the omitted proofs can be found in the supplementary material.

4.1 Proof of Theorems 1 and 2

We first introduce an essential lemma about the property of our potential function (8), which is a variant of Lemma 5 in Luo and Schapire [32].

Lemma 5. For any $I \in \mathcal{I}$ and $t \in I$, Algorithm 2 has

$$\Phi(R_t^I, C_t^I) \leqslant \Phi(R_{t-1}^I, C_{t-1}^I) + w_t^I \tilde{r}_t^I + \frac{|\tilde{r}_t^I|}{2(C_{t-1}^I + 1)}. \tag{27}$$

For any $I \in \mathcal{I}$, there must be an integer $i \geqslant 0$ such that $I \subseteq [i \cdot 2^{\lceil \log \tau_2 \rceil} + 1, (i+1) \cdot 2^{\lceil \log \tau_2 \rceil}]$, due to the definition of $\mathcal{I}$. Therefore, we define $\mathcal{I}' = \{I' \in \mathcal{I} : I' \subseteq [t_1, t_2]\}$, where $t_1 = i \cdot 2^{\lceil \log \tau_2 \rceil} + 1$ and $t_2 = (i+1) \cdot 2^{\lceil \log \tau_2 \rceil}$. Repeatedly applying Lemma 5, for any $k \in I$, we have

$$\begin{aligned}
&\sum_{I'=[q',s'] \in \mathcal{I}'} \Phi(R_{k\wedge s'}^{I'}, C_{k\wedge s'}^{I'}) \\
\leqslant &\sum_{I'=[q',s'] \in \mathcal{I}'} \left(\Phi(R_{k\wedge s'-1}^{I'}, C_{k\wedge s'-1}^{I'}) + w_{k\wedge s'}^{I'} \tilde{r}_{k\wedge s'}^{I'} + \frac{|\tilde{r}_{k\wedge s'}^{I'}|}{2(C_{k\wedge s'-1}^{I'} + 1)} \right) \\
\leqslant &|\mathcal{I}'| + \sum_{t=t_1}^{k} \sum_{I' \in \mathcal{I}'} \mathbb{I}_{[t \in I']} \tilde{r}_t^{I'} w_t^{I'} + \sum_{I'=[q',s'] \in \mathcal{I}'} \sum_{i=q'}^{k\wedge s'} \frac{|\tilde{r}_i^{I'}|}{2(C_{i-1}^{I'} + 1)} \\
\leqslant &|\mathcal{I}'| + \sum_{t=t_1}^{k} \sum_{I' \in \mathcal{I}'} \mathbb{I}_{[t \in I']} \tilde{r}_t^{I'} w_t^{I'} + \sum_{I'=[q',s'] \in \mathcal{I}'} \sum_{i=q'}^{s'} \frac{|\tilde{r}_i^{I'}|}{2(C_{i-1}^{I'} + 1)}
\end{aligned} \tag{28}$$

where $k \wedge s' = \min(k, s')$. It is easy to verify that

$$|\mathcal{I}'| = \sum_{j=\lceil \log \tau_1 \rceil}^{\lceil \log \tau_2 \rceil} \frac{2^{\lceil \log \tau_2 \rceil}}{2^j} = 2^{\lceil \log \tau_2 \rceil - \lceil \log \tau_1 \rceil + 1} - 1 \leqslant \frac{4\tau_2}{\tau_1}. \tag{29}$$

Moreover, because of $p_t^{I'} \propto [w_t^{I'}]_+$, for any $t \in [t_1, t_2]$,we have

$$\begin{aligned}
&\sum_{I'\in\mathcal{I}'} \mathbb{I}_{[t\in I']}\tilde{r}_t^{I'} w_t^{I'} \\
=&\sum_{I'\in\mathcal{I}':\mathbb{I}_{[t\in I']}w_t^{I'}>0} [w_t^{I'}]_+(f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^{I'})) + \sum_{I'\in\mathcal{I}':\mathbb{I}_{[t\in I']}w_t^{I'}\leqslant 0} \mathbb{I}_{[t\in I']}w_t^{I'}[f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^{I'})]_+ \\
\leqslant&\left(\sum_{I'\in\text{Active}(t)}[w_t^{I'}]_+\right)\sum_{I'\in\mathcal{I}':\mathbb{I}_{[t\in I']}w_t^{I'}>0} p_t^{I'}(f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^{I'}))\leqslant 0
\end{aligned} \tag{30}$$

where the last inequality is due to $\text{Active}(t)\subseteq\mathcal{I}'$, $\mathbf{x}_t=\sum_{I'\in\text{Active}(t)}p_t^{I'}\mathbf{x}_t^{I'}$ and Jensen's inequality. To bound the last term in (28), we further introduce the following lemma.

Lemma 6. **(Lemma 14 of Gaillard et al. [40])** Let $a_0>0$ and $a_1,\cdots,a_m\in[0,1]$ be real numbers and let $f:(0,+\infty)\mapsto[0,+\infty)$ be a nonincreasing function. Then

$$\sum_{i=1}^m a_i f\left(\sum_{j=0}^{i-1}a_j\right)\leqslant f(a_0)+\int_{a_0}^{\sum_{j=0}^m a_j} f(x)dx. \tag{31}$$

Applying Lemma 6 with $f(x)=1/x$, for any $I'=[q',s']\in\mathcal{I}'$, we have

$$\sum_{i=q'}^{s'}\frac{|\tilde{r}_i^{I'}|}{(C_{i-1}^{I'}+1)}\leqslant 1+\int_1^{1+C_{s'}^{I'}}\frac{1}{x}dx=1+\ln(1+C_{s'}^{I'}). \tag{32}$$

Substituting (29), (30) and (32) into (28), we have

$$\begin{aligned}
\sum_{I'=[q',s']\in\mathcal{I}'}\Phi(R_{k\wedge s'}^{I'},C_{k\wedge s'}^{I'})\leqslant&\frac{4\tau_2}{\tau_1}+\frac{1}{2}\sum_{I'=[q',s']\in\mathcal{I}'}(1+\ln(1+C_{s'}^{I'})) \\
\leqslant&\frac{4\tau_2(3+\ln(1+t_2-t_1+1))}{2\tau_1} \\
\leqslant&\frac{2\tau_2(3+\ln(1+2\tau_2))}{\tau_1} \\
=&\exp(c/3)
\end{aligned} \tag{33}$$

where $c=3\ln(2\tau_2(3+\ln(1+2\tau_2))/\tau_1)$. According to the definition and $I\in\mathcal{I}'$, we further have

$$|R_k^I|=\sqrt{3C_k^I\ln\Phi(R_k^I,C_k^I)}\leqslant\sqrt{3C_k^I\ln\sum_{I'=[q',s']\in\mathcal{I}'}\Phi(R_{k\wedge s'}^{I'},C_{k\wedge s'}^{I'})}\leqslant\sqrt{cC_k^I}. \tag{34}$$

Then, for any $[q,s]\subseteq I$, we have

$$\begin{aligned}
\sum_{t=q}^s f_t(\mathbf{x}_t)-\sum_{t=q}^s f_t(\mathbf{x}_t^I)\leqslant\sum_{t=q}^s\tilde{r}_t^I=&\sum_{t=1}^s\mathbb{I}_{[t\in I]}\tilde{r}_t^I-\sum_{t=1}^{q-1}\mathbb{I}_{[t\in I]}\tilde{r}_t^I \\
\leqslant&|R_s^I-R_{q-1}^I|\leqslant|R_s^I|+|R_{q-1}^I|\leqslant 2\sqrt{cC_s^I}.
\end{aligned} \tag{35}$$

It is easy to obtain (12) in Theorem 1 due to $C_s^I\leqslant|I|$.

For brevity, let $L_k^I=\sum_{t=1}^k\mathbb{I}_{[t\in I]}f_t(\mathbf{x}_t^I)$ for any $k\in I$. To prove (19) in Theorem 2, we note that for any $k\in I$

$$\begin{aligned}
C_k^I=&\sum_{t=1}^k\mathbb{I}_{[t\in I]}|\tilde{r}_t^I|=\sum_{t=1}^k\mathbb{I}_{[t\in I]}(\tilde{r}_t^I+2[-\tilde{r}_t^I]_+) \\
=&R_k^I+2\sum_{t=1}^k\mathbb{I}_{[t\in I]}[-\tilde{r}_t^I]_+\leqslant R_k^I+2L_k^I.
\end{aligned} \tag{36}$$

where the last inequality is due to $[-\tilde{r}_t^I]_+ = f_t(\mathbf{x}_t^I) - f_i(\mathbf{x}_t) \leqslant f_t(\mathbf{x}_t^I)$ when $\tilde{r}_t^I < 0$ and $[-\tilde{r}_t^I]_+ = 0 \leqslant f_t(\mathbf{x}_t^I)$ when $\tilde{r}_t^I \geqslant 0$. Plugging the above inequality into (34) and taking square on both sides, we have $(R_k^I)^2 \leqslant cR_k^I + 2cL_k^I$ which implies that

$$|R_k^I| \leqslant \frac{c+\sqrt{c^2+8cL_k^I}}{2} \leqslant c + \sqrt{2cL_k^I}. \tag{37}$$

Replacing the last inequality in (35) with the above inequality, we have

$$\begin{aligned}\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}_t^I) \leqslant& |R_s^I| + |R_{q-1}^I| \\ \leqslant& 2c + 2\sqrt{2cL_s^I}.\end{aligned} \tag{38}$$

4.2 Proof of Lemma 5

Lemma 5 can be derived by following the proof of Lemma 5 in Luo and Schapire [32] with slight modifications to deal with our potential function (8). We include this proof for completeness.

It is easy to derive that

$$\Phi(R_{t-1}^I + r, C_{t-1}^I + |r|) = \exp\left(\frac{(R_{t-1}^I + r)^2}{3\left(C_{t-1}^I + |r|\right)}\right) \tag{39}$$

as a function of r is convex on $r \in [-1,0]$ and $r \in [0,1]$ respectively, due to

$$\frac{(R_{t-1}^I + r)^2}{C_{t-1}^I + |r|} = (C_{t-1}^I + r) + \frac{(R_{t-1}^I - C_{t-1}^I)^2}{C_{t-1}^I + r} + 2(R_{t-1}^I - C_{t-1}^I) \tag{40}$$

when $r \in [0,1]$ and

$$\frac{(R_{t-1}^I + r)^2}{C_{t-1}^I + |r|} = (C_{t-1}^I - r) + \frac{(R_{t-1}^I + C_{t-1}^I)^2}{C_{t-1}^I - r} - 2(R_{t-1}^I + C_{t-1}^I) \tag{41}$$

when $r \in [-1,0]$.

Furthermore, we define a function $h(r) = \Phi(R_{t-1}^I + r, C_{t-1}^I + |r|) - w_t^I r$, and it is also convex on $r \in [-1,0]$ and $r \in [0,1]$, respectively. According to the definition of weight function (6), we have

$$\begin{aligned}h(1) &= \Phi(R_{t-1}^I + 1, C_{t-1}^I + |1|) - w_t^I \\ &= \Phi(R_{t-1}^I + 1, C_{t-1}^I + |1|) - \frac{1}{2}\left(\Phi(R_{t-1}^I + 1, C_{t-1}^I + 1) - \Phi(R_{t-1}^I - 1, C_{t-1}^I + 1)\right) \\ &= \frac{1}{2}\left(\Phi(R_{t-1}^I + 1, C_{t-1}^I + 1) + \Phi(R_{t-1}^I - 1, C_{t-1}^I + 1)\right).\end{aligned} \tag{42}$$

Similarly, we have

$$h(-1) = \frac{1}{2}\left(\Phi(R_{t-1}^I + 1, C_{t-1}^I + 1) + \Phi(R_{t-1}^I - 1, C_{t-1}^I + 1)\right) = h(1). \tag{43}$$

Due to the property of convex functions, we have

$$h(r) \leqslant \begin{cases}(1-r)h(0) + rh(1), r \in [0,1]; \\ (1+r)h(0) - rh(-1), r \in [-1,0].\end{cases} \tag{44}$$

Therefore, when $h(0) \geqslant h(1) = h(-1)$, we have $h(r) \leqslant h(0)$, which implies

$$\Phi(R_{t-1}^I + r, C_{t-1}^I + |r|) \leqslant \Phi(R_{t-1}^I, C_{t-1}^I) + w_t^I r \tag{45}$$

for $r \in [-1,1]$. When $h(0) \leqslant h(1) = h(-1)$, we have

$$h(r) \leqslant \max\{h(0) + (h(0) - h(-1))r, h(0) + (h(1) - h(0))r\} = h(0) + (h(1) - h(0))|r|. \tag{46}$$

To bound $h(1)-h(0)$, we further define a function as

$$g(R) =(\Phi(R+1, C_{t-1}^I+1)+\Phi(R-1, C_{t-1}^I+1))-2\Phi(R, C_{t-1}^I). \tag{47}$$

It is easy to verify

$$h(1)-h(0)=\frac{1}{2}g(R_{t-1}^I) \tag{48}$$

which implies that $h(1)-h(0)$ can be bounded by the maximum of $\frac{1}{2}g(R_{t-1}^I)$. Therefore, we introduce the following lemma, which is derived from the proof of Lemma 2 of Luo and Schapire [41].

Lemma 7. Let $F(s)=\exp\left(\frac{(s+1)^2}{3a}\right)+\exp\left(\frac{(s-1)^2}{3a})\right)-2\exp\left(\frac{s^2}{3(a-1)}\right)$, where $a>1$ is a constant, the derivatives of $F(s)$ satisfy

$$\begin{cases} F'(s)\geqslant 0, s<0; \\ F'(s)=0, s=0; \\ F'(s)\leqslant 0, s>0. \end{cases} \tag{49}$$

When $C_{t-1}^I>0$, applying Lemma 7, we have that $g'(0)=0$, $g'(R)\geqslant 0$ for $R<0$ and $g'(R)\leqslant 0$ for $R>0$, which implies that $g(0)$ is the maximum. So, when $C_{t-1}^I>0$, we have

$$h(1)-h(0)=\frac{1}{2}g(R_{t-1}^I)\leqslant\frac{1}{2}g(0)=\exp\left(\frac{1}{3(C_{t-1}^I+1)}\right)-1\leqslant\frac{1}{2(C_{t-1}^I+1)} \tag{50}$$

where the last inequality is due to $e^x-1\leqslant\frac{e^z-1}{z}x$ for $x\in[0,z]$. It is easy to verify that this inequality (50) still holds when $C_{t-1}^I=0$, because $C_{t-1}^I=0$ implies $R_{t-1}^I=0$. Combining (50) with (46), for $r\in[-1,1]$, we have

$$\Phi(R_{t-1}^I+r, C_{t-1}^I+|r|)\leqslant\Phi(R_{t-1}^I, C_{t-1}^I)+w_t^I r+\frac{|r|}{2(C_{t-1}^I+1)}. \tag{51}$$

Finally, combining the fact $\tilde{r}_t^I\in[-1,1]$ with (45) and (51), we complete the proof.

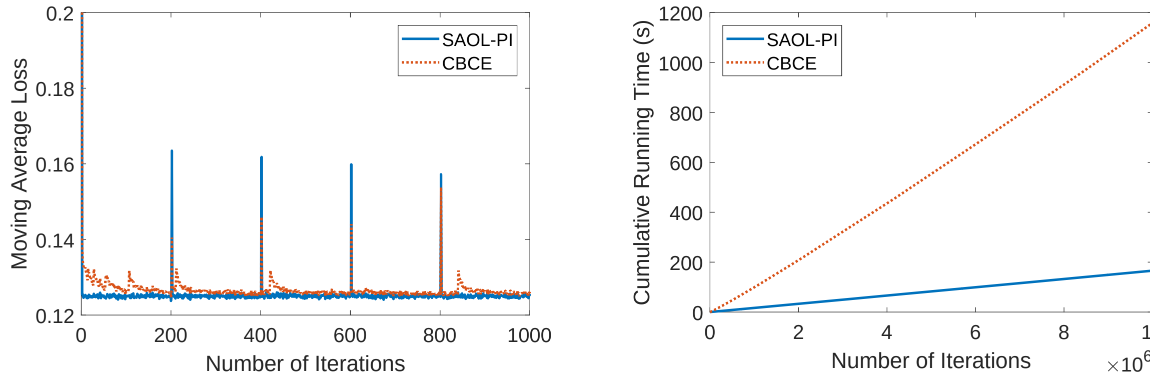
5 Experiments

In this section, we perform numerical experiments on LEA and OCO to verify the efficiency and effectiveness of our proposed algorithm. We compare our SAOL-PI with CBCE [5] that enjoys the best strongly adaptive regret bound. Following Jun et al. [5], we set the prior weight over instances $\mathcal{B}_I$ to uniform distribution for CBCE.

5.1 Learning with Experts Advice

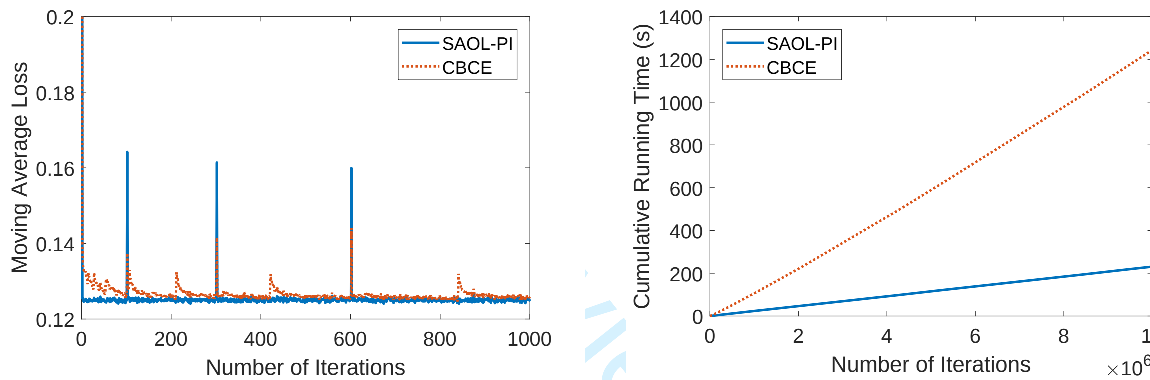
In the setting of LEA, we set $T=1e7$ and $N=10$. Following Jun et al. [5], we first draw loss $\boldsymbol{\ell}_t(i), \forall t=1,2,\cdots,1e7$ for each expert i independently from $[0,1]$ by uniform sampling. Then, we consider two kinds of changing environments including a fixed frequency setting and a varying frequency setting. In the fixed frequency setting, for $t\in[2(i-1)\cdot 1e6+1, 2i\cdot 1e6]$ where $i=1,2,\cdots,5$, we reduce loss of expert i by $\boldsymbol{\ell}_t(i)=[\boldsymbol{\ell}_t(i)-0.5]_+$ and reduce loss of expert $i-1$ by $\boldsymbol{\ell}_t(i-1)=[\boldsymbol{\ell}_t(i-1)-0.3]_+$ when $i-1>0$. As a result, during time interval $[2(i-1)\cdot 1e6+1, 2i\cdot 1e6]$, where $i=1,2,\cdots,5$, the best expert is i, which changes after each $2e6$ rounds. In the varying frequency setting, for $t\in[s_i\cdot 1e6+1, s_{i+1}\cdot 1e6]$ where $s_i=i\cdot(i-1)/2$ and $i=1,2,3,4$, we reduce loss of expert i by $\boldsymbol{\ell}_t(i)=[\boldsymbol{\ell}_t(i)-0.5]_+$ and reduce loss of expert $i-1$ by $\boldsymbol{\ell}_t(i-1)=[\boldsymbol{\ell}_t(i-1)-0.3]_+$ when $i-1>0$. In this way, the best expert is i during time interval $[s_i\cdot 1e6+1, s_{i+1}\cdot 1e6]$, where $i=1,2,3,4$, which changes without a fixed frequency. To utilize the prior information, for our SAOL-PI, we set $\tau_1=1e6, \tau_2=2e6$ in the fixed frequency setting, and set $\tau_1=1e6, \tau_2=4e6$ in the varying frequency setting. For each meta-algorithm, we use our Algorithm 1 as the black-box. We repeat the experiment 50 times and report the average results of all algorithms.

Figures 3 and 4 show the comparison of moving average loss with window size $1e4$ and cumulative running time among different algorithms in the above two settings, respectively. Compared with CBCE, we observe that our SAOL-PI has smaller loss when the shift has settled down and is very stable. Furthermore, our SAOL-PI achieves 7.0 times speed-up in the fixed frequency setting and 5.4 times



(a) Comparison of moving average loss (b) Comparison of cumulative running time

Figure 3 Experimental results for learning with experts advice in the fixed frequency setting.



(a) Comparison of moving average loss (b) Comparison of cumulative running time

Figure 4 Experimental results for learning with experts advice in the varying frequency setting.

speed-up in the varying frequency setting, and these ratios will increase with T due to the $O(\log T)$ complexity of CBCE. We also note that when the environment changes, our SAOL-PI may catch up with it slower than CBCE in the beginning, because CBCE maintains more instances of the black-box. However, from Figures 3(a) and 4(a), we observe that SAOL-PI converges much faster, and its loss becomes smaller than that of CBCE very quickly.

5.2 Online Convex Optimization

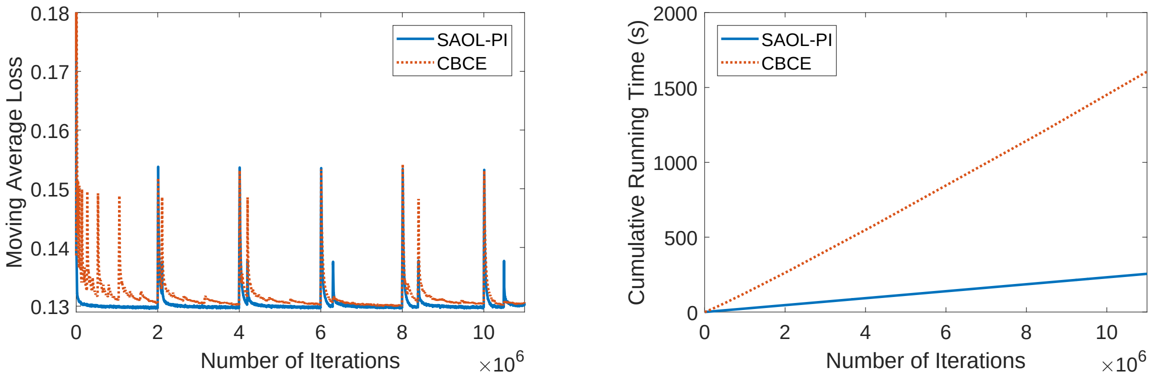
Then, we consider the problem of online logistic regression with two real-world datasets from LIBSVM repository [42]: HIGGS and SUSY. At each round $t = 1, 2, \cdots, T$, the leaner receives a single example $(\mathbf{z}_t, y_t)$ and suffers the logistic loss $f_t(\mathbf{x}) = \beta \ln(1 + \exp(-y_t \mathbf{x}^\top \mathbf{z}_t))$, where $\mathbf{z}_t \in \mathbb{R}^d$, $y_t \in \{-1, 1\}$ and β is a constant. According to the size of each dataset, we have $T = 1.1e7, d = 28$ for HIGGS and $T = 5e6, d = 18$ for SUSY.

Following Wang et al. [34], a scenario of changing environments is created for each dataset, as follows.

- For HIGGS, we flip the labels of samples in $t \in [2(i-1) \cdot 1e6 + 1, 2i \cdot 1e6], i = 2, 4$ and $t \in [1e7+1, 1.1e7]$ as $y_t = -y_t$.
- For SUSY, we flip the labels of samples in $t \in [2e6 + 1, 4e6]$ as $y_t = -y_t$.

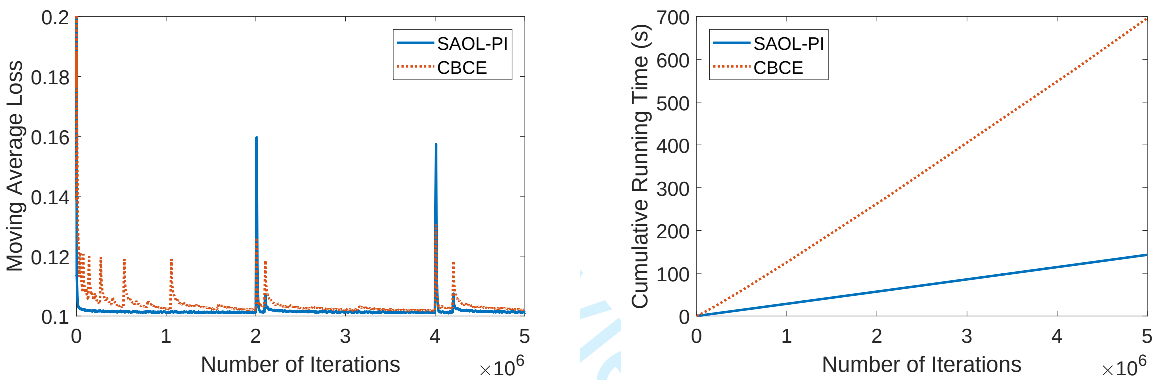
In this way, the optimal predictor will change after each $2e6$ rounds. To make Assumptions 1, 2 and 3 satisfied, we first bound the convex domain in a d-dimensional ball with radius 10, which means $\|\mathbf{x} - \mathbf{y}\|_2 \leqslant D = 20$ for any $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{X}$. Then, we set $\beta = 1/5$ and cap the loss $f_t(\mathbf{x})$ above at 1 to ensure $f_t(\mathbf{x}) \in [0, 1]$. We further normalize each data $\mathbf{z}_t$ by $\mathbf{z}_t = \mathbf{z}_t / \|\mathbf{z}_t\|_2$, which leads to $\|\nabla f_t(\mathbf{x})\|_2 \leqslant G = \beta = 1/5$. We set $\tau_1 = 1e6$ and $\tau_2 = 2e6$ for our SAOL-PI. For each meta-algorithm, we use Algorithm 3 as the black-box and set $\delta = 10^{-4}, \alpha = D/\sqrt{2} = 10\sqrt{2}$. We repeat the experiment 50 times with random permutations of each dataset and report the average results of all algorithms.

Figures 5 and 6 show the comparison of moving average loss with window size $1e4$ and cumulative



(a) Comparison of moving average loss (b) Comparison of cumulative running time

Figure 5 Experimental results for online logistic regression on HIGGS.



(a) Comparison of moving average loss (b) Comparison of cumulative running time

Figure 6 Experimental results for online logistic regression on SUSY.

running time among different algorithms on HIGGS and SUSY, respectively. We find that our SAOL-PI has smaller loss when the shift has settled down, and is much faster than CBCE on both datasets. Specifically, our SAOL-PI achieves 6.3 times speed-up on HIGGS and 4.9 times speed-up on SUSY, and these ratios will increase with T due to the $O(\log T)$ complexity of CBCE.

6 Conclusions

In this paper, we propose an improved strongly adaptive algorithm by utilizing prior information. Given the lower bound τ_1 and upper bound τ_2 on how long the environment changes, in each round t, our SAOL-PI only need to maintain $O(\log(\tau_2/\tau_1))$ instances of the black-box, which could be far less than $O(\log t)$ needed by previous strongly adaptive algorithms. Theoretical analysis shows that the regret bound of our meta-algorithm on any time interval with length $\tau \in [\tau_1, \tau_2]$ is $O(\sqrt{\tau \log(\tau_2/\tau_1)})$, which is better than $O(\sqrt{\tau \log T})$ established by CBCE. Furthermore, we improve the meta regret bound to a problem-dependent one, which could be much tighter when the loss of the competitor is small. Numerical experiments demonstrate the efficiency and effectiveness of our proposed algorithm.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (61976112), Jiangsu Science Foundation (BK20200064), and Open Research Projects of Zhejiang Lab (NO. 2021KB0AB02).

Supporting information Appendix A ∼ Appendix G. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Cesa-Bianchi N, Freund Y, Haussler D, et al. How to use expert advice. J ACM, 1997, 44: 427–485

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

3 Zhang L J. Online learning in changing environments. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, Online, 2020. 5178-5182

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

5 Jun K-S, Orabona F, Wright S, et al. Improved strongly adaptive online learning using coin betting. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, 2017. 943–951

6 Auer P, Cesa-Bianchi N, Gentile C. Adaptive and self-confident on-line learning algorithms. J Comput Syst Sci, 2002, 64: 48–75

7 Shalev-Shwartz S. Online learning: Theory, algorithms, and applications. Dissertation for the Doctoral Degree. Jerusalem: The Hebrew University of Jerusalem, 2007. 1–162

8 Hazan E, Agarwal A, Kale S. Logarithmic regret algorithms for online convex optimization. Mach Learn, 2007, 69: 169–192

9 Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res, 2011, 12: 2121–2159

10 Hazan E, Kale S. Projection-free online learning. In: Proceedings of the 29th International Conference on Machine Learning, Edinburgh, 2012. 1843–1850

11 Zhang L J, Jin R, Chen C, Bu J J, He X F. Efficient online learning for large-scale sparse kernel logistic regression. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, 2012. 1219–1225

12 Zhang L J, Yi J F, Jin R, Lin M, He X F. Online kernel learning with a near optimal sparsity bound. In: Proceedings of the 30th International Conference on Machine Learning, Atlanta, 2013. 621–629

13 Oiwa H, Matsushima S, Nakagawa H. Feature-aware regularization for sparse online learning. Sci China Inf Sci, 2014, 57: 052104(21)

14 Wan Y Y, Wei N, Zhang L J. Efficient adaptive online learning via frequent directions. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, 2018. 2748–2754

15 Wang Y H, Lin P, Hong Y G. Distributed regression estimation with incomplete data in multi-agent networks. Sci China Inf Sci, 2018, 61(9): 092202

16 Wan Y Y, Tu W-W, Zhang L J. Projection-free distributed online convex optimization with $O(\sqrt{T})$ communication complexity. In: Proceedings of the 37th International Conference on Machine Learning, Online, 2020. 9818–9828

17 Wan Y Y, Zhang L J. Projection-free online learning over strongly convex sets. 2020. ArXiv:2010.08177

18 Hou B-J, Zhang L J, Zhou Z-H. Learning with feature evolvable streams. In: Advances in Neural Information Processing Systems 30, Long Beach, 2017. 1416–1426

19 Wang C Y, Xie L, Wang W, et al. Moving tag detection via physical layer analysis for large-scale RFID systems. In: Proceedings of the 35th Annual IEEE International Conference on Computer Communications, Calcutta, 2016. 1–9

20 Wells W D, Gubar G. Life cycle concept in marketing research. J Mark Res, 1966, 3: 355–363

21 Yang J-W, Yu Y, Zhang X-P. Life-stage modeling by customer-manifold embedding. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, 2017. 3259–3265

22 Bojanic D C. The impact of age and family life experiences on mexican visitor shopping expenditures. Tourism Manage, 2011, 32: 406–414

23 Hazan E. Introduction to online convex optimization. Found Trends Optim, 2016, 2: 157–325

24 Shalev-Shwartz S. Online learning and online convex optimization. Found Trends Mach Learn, 2011, 4: 107–194

25 Cesa-Bianchi N, Orabona F. Online learning algorithms. Annu Rev Stat Appl, 2020, 8: 1–26

26 Hazan E, Seshadhri C. Adaptive algorithms for online decision problems. Electron Colloq Comput Complex, 2007, 14: 88

27 Abernethy J D, Bartlett P L, Rakhlin A, et al. Optimal stragies and minimax lower bounds for online convex games. In: Proceedings of the 21st Annual Conference on Learning Theory, Helsinki, 2008. 415–424

28 Arora S, Hazan E, Kale S. The multiplicative weights update method: A meta-algorithm and applications. Theor Comput, 2012, 8: 121–164

29 Freund Y, Schapire R E, Singer Y, et al. Using and combining predictors that specialize. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, 1997. 334–343

30 Orabona F, Pal D. Coin betting and parameter-free online learning. In: Advances in Neural Information Processing Systems 29, Barcelona, 2016. 577–585

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

32 Luo H P, Schapire R E. Achieving all with no parameters: AdaNormalHedge. In: Proceedings of the 28th Conference on Learning Theory, Paris, 2015. 1286–1304

33 Orabona F, Pal D. Scale-free online learning. Theor Comput Sci, 2018, 716: 50–69

34 Wang G H, Zhao D K, Zhang L J. Minimizing adaptive regret with one gradient per iteration. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, 2018. 2762–2768

35 van Erven T, Koolen W M. MetaGrad: Multiple learning rates in online learning. In: Advances in Neural Information Processing Systems 29, Barcelona, 2016. 3666–3674

36 Zhang L J, Yang T B, Jin R, et al. Dynamic regret of strongly adaptive methods. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 5877–5886

37 Zhang L J, Wang G H, Tu W-W, et al. Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. 2019. ArXiv:1906.10851

38 Zhang L J, Lu S Y, Yang T B. Minimizing dynamic regret and adaptive regret simultaneously. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, Palermo, 2020. 309–319

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

40 Gaillard P, Stoltz G, van Erven T. A second-order bound with excess losses. In: Proceedings of the 27th Annual Conference on Learning Theory, Barcelona, 2014. 176–196

41 Luo H P, Schapire R E. A drifting-games analysis for online learning and applications to boosting. In: Advances in Neural Information Processing Systems 27, Montreal, 2014. 1368–1376

42 Chang C-C, Lin C-J. LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol, 2011, 2: 1–27

SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Strongly Adaptive Online Learning over Partial Intervals

Yuanyu WAN[1], Wei-Wei TU[2] & Lijun ZHANG[1*]

[1] *National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210023*, China;*
[2] *4Paradigm Inc., Beijing* 100000*, China*

Appendix A Proof of Lemmas 1 and 3

Because the weighting method used in Algorithm 2 can be reduced to the modified AdaNormalHedgeg shown in Algorithm 1 by keeping all experts active, Theorems 1 and 2 can also be reduced to Lemmas 1 and 3, respectively. Following the proof of Theorems 1 and 2, for any $i \in [N]$, it is easy to verify that

$$\sum_{t=q}^{s}\langle \boldsymbol{\ell}_t, \mathbf{x}_t^I\rangle - \sum_{t=q}^{s}\boldsymbol{\ell}_t(i) \leqslant 2\sqrt{\tilde{c}(|I|)|I|} \tag{A1}$$

and

$$\sum_{t=q}^{s}\langle \boldsymbol{\ell}_t, \mathbf{x}_t^I\rangle - \sum_{t=q}^{s}\boldsymbol{\ell}_t(i) \leqslant 2\tilde{c}(|I|) + 2\sqrt{2\tilde{c}(|I|)\sum_{t=1}^{s}\mathbb{I}_{[t\in I]}\boldsymbol{\ell}_t(i)} \tag{A2}$$

where $\tilde{c}(|I|) = 3\ln\frac{N(3+\ln(1+|I|))}{2}$. Because of $\mathbf{x} \in \Delta^N$, multiplying both sides of (A1) by $\mathbf{x}(i)$ and summing over N, we have

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) = \sum_{t=q}^{s}\langle \boldsymbol{\ell}_t, \mathbf{x}_t^I\rangle - \sum_{t=q}^{s}\langle \boldsymbol{\ell}_t, \mathbf{x}\rangle \leqslant 2\sqrt{\tilde{c}(|I|)|I|}.$$

Similarly, multiplying both sides of (A2) by $\mathbf{x}(i)$ and summing over N, we have

$$\begin{aligned}\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) =& \sum_{t=q}^{s}\langle \boldsymbol{\ell}_t, \mathbf{x}_t^I\rangle - \sum_{t=q}^{s}\langle \boldsymbol{\ell}_t, \mathbf{x}\rangle \\ \leqslant& 2\tilde{c}(|I|) + 2\sqrt{2\tilde{c}(|I|)}\sum_{i=1}^{N}\mathbf{x}(i)\sqrt{\sum_{t=1}^{s}\mathbb{I}_{[t\in I]}\boldsymbol{\ell}_t(i)} \\ \leqslant& 2\tilde{c}(|I|) + 2\sqrt{2\tilde{c}(|I|)}\sqrt{\sum_{i=1}^{N}\mathbf{x}(i)\sum_{t=1}^{s}\mathbb{I}_{[t\in I]}\boldsymbol{\ell}_t(i)} \\ \leqslant& 2\tilde{c}(|I|) + 2\sqrt{2\tilde{c}(|I|)\sum_{t=1}^{s}\mathbb{I}_{[t\in I]}f_t(\mathbf{x})}\end{aligned} \tag{A3}$$

where the second inequality is due to Jensens inequality.

Appendix B Proof of Lemmas 2 and 4

The regret bound of SOGD over the interval I has been analyzed by Orabona and Pal [33] for online linear optimization and further refined by Zhang et al. [31] for online convex optimization with smooth loss functions. However, we need to bound the regret over any subinterval $[q, s] \subseteq I$, which requires additional analysis. For the sake of completeness, we include the detailed proof.

For brevity, let $\hat{\mathbf{x}}_{t+1}^I = \mathbf{x}_t^I - \eta_t^I\nabla f_t(\mathbf{x}_t^I)$ and assume $I = [t_1, t_2]$. Because f_t is convex function, for any $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned}f_t(\mathbf{x}_t^I) - f_t(\mathbf{x}) \leqslant& \langle\nabla f_t(\mathbf{x}_t^I), \mathbf{x}_t^I - \mathbf{x}\rangle = \frac{1}{\eta_t^I}\langle \mathbf{x}_t - \hat{\mathbf{x}}_{t+1}^I, \mathbf{x}_t - \mathbf{x}\rangle \\ =& \frac{1}{2\eta_t^I}\left(\|\mathbf{x}_t^I - \mathbf{x}\|_2^2 - \|\hat{\mathbf{x}}_{t+1}^I - \mathbf{x}\|_2^2 + \|\mathbf{x}_t^I - \hat{\mathbf{x}}_{t+1}^I\|_2^2\right) \\ \leqslant& \frac{1}{2\eta_t^I}\left(\|\mathbf{x}_t^I - \mathbf{x}\|_2^2 - \|\mathbf{x}_{t+1}^I - \mathbf{x}\|_2^2\right) + \frac{\eta_t^I}{2}\|\nabla f_t(\mathbf{x}_t^I)\|_2^2.\end{aligned} \tag{B1}$$

* Corresponding author (email: zhanglj@lamda.nju.edu.cn)

For any $[q,s] \subseteq I = [t_1, t_2]$, summing the inequalities of iterations during $[q,s]$, we have

$$\begin{aligned}\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant& \frac{1}{2\eta_q^I}\|\mathbf{x}_q^I - \mathbf{x}\|_2^2 + \sum_{t=q+1}^{s}\left(\frac{1}{\eta_t^I} - \frac{1}{\eta_{t-1}^I}\right)\frac{\|\mathbf{x}_t^I - \mathbf{x}\|_2^2}{2} + \frac{1}{2}\sum_{t=q}^{s}\eta_t^I\|\nabla f_t(\mathbf{x}_t^I)\|_2^2 \\ \leqslant& \frac{D^2}{2\eta_q^I} + \sum_{t=q+1}^{s}\left(\frac{1}{\eta_t^I} - \frac{1}{\eta_{t-1}^I}\right)\frac{D^2}{2} + \frac{1}{2}\sum_{t=1}^{s}\eta_t^I\|\nabla f_t(\mathbf{x}_t^I)\|_2^2 \\ =& \frac{D^2}{2\eta_s^I} + \frac{1}{2}\sum_{t=t_1}^{s}\eta_t^I\|\nabla f_t(\mathbf{x}_t^I)\|_2^2\end{aligned} \tag{B2}$$

where the second inequality is due to Assumption 2. To bound $\sum_{t=t_1}^{s}\eta_t^I\|\nabla f_t(\mathbf{x}_t^I)\|_2^2$, we introduce the following lemma.

Lemma 8. (Lemma 3.5 of Auer et al. [6]) Let $a_1, \cdots, a_T$ and δ be non-negative real numbers. Then

$$\sum_{t=1}^{T}\frac{a_t}{\sqrt{\delta + \sum_{i=1}^{t} a_i}} \leqslant 2\left(\sqrt{\delta + \sum_{t=1}^{T} a_t} - \sqrt{\delta}\right) \tag{B3}$$

where $0/\sqrt{0} = 0$.

According to the definition of η_t^I shown in Algorithm 3 and Lemma 8, we have

$$\sum_{t=t_1}^{s}\eta_t^I\|\nabla f_t(\mathbf{x}_t^I)\|_2^2 = \alpha\sum_{t=t_1}^{s}\frac{\|\nabla f_t(\mathbf{x}_t^I)\|_2^2}{\sqrt{\delta + \sum_{i=t_1}^{t}\|\nabla f_i(\mathbf{x}_i^I)\|_2^2}} \leqslant 2\alpha\sqrt{\delta + \sum_{t=t_1}^{s}\|\nabla f_t(\mathbf{x}_t^I)\|_2^2}. \tag{B4}$$

Substituting (B4) and $\alpha = D/\sqrt{2}$ into (B2), we have

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant \sqrt{2}D\sqrt{\delta + \sum_{t=t_1}^{s}\|\nabla f_t(\mathbf{x}_t^I)\|_2^2}. \tag{B5}$$

When Assumption 3 is satisfied, we have $\|\nabla f_t(\mathbf{x})\|_2 \leqslant G$ for any $\mathbf{x} \in \mathcal{X}$ and t. Combining with $s - t_1 + 1 \leqslant |I|$, it is easy to obtain (15) in Lemma 2 from (B5).

To further utilize the smoothness shown in Assumption 4, we introduce the self-bounding property of smooth functions.

Lemma 9. (Lemma 3.1 of Srebro et al. [39]) For an H-smooth and nonnegative function $f : \mathcal{X} \mapsto \mathbb{R}$,

$$\|\nabla f(\mathbf{x})\|_2 \leqslant \sqrt{4Hf(\mathbf{x})}, \forall \mathbf{x} \in \mathcal{X}. \tag{B6}$$

According to Lemma 9, Assumptions 1 and 4, we have

$$\|\nabla f_t(\mathbf{x})\|_2^2 \leqslant 4Hf_t(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}. \tag{B7}$$

Combining (B5) and (B7), we have

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant \sqrt{2}D\sqrt{\delta + 4H\sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I)} \leqslant \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I)}. \tag{B8}$$

To replace $\sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I)$ with $\sum_{t=t_1}^{s} f_t(\mathbf{x})$, we use the following lemma.

Lemma 10. (Lemma 19 of Shalev-Shwartz [7]) Let $x, b, c \in \mathbb{R}_+$. Then,

$$x - c \leqslant b\sqrt{x} \Rightarrow x - c \leqslant b^2 + b\sqrt{c}. \tag{B9}$$

Note that (B8) holds for any $[q,s] \subseteq I = [t_1, t_2]$, which implies

$$\left(\frac{\delta}{4H} + \sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I)\right) - \left(\frac{\delta}{4H} + \sum_{t=t_1}^{s} f_t(\mathbf{x})\right) \leqslant \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I)}. \tag{B10}$$

Applying Lemma 10 into the above inequality, we have

$$\begin{aligned}\sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=t_1}^{s} f_t(\mathbf{x}) \leqslant& 8HD^2 + \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{s} f_t(\mathbf{x})} \\ =& 8HD^2 + D\sqrt{2\delta + 8H\sum_{t=t_1}^{s} f_t(\mathbf{x})}.\end{aligned} \tag{B11}$$

Then, if $\sum_{t=t_1}^{q-1} f_t(\mathbf{x}_t^I) - \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) \geqslant 0$, from the above inequality, it is easy to obtain

$$\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant 8HD^2 + D\sqrt{2\delta + 8H\sum_{t=t_1}^{s} f_t(\mathbf{x})}. \tag{B12}$$

In the case $\sum_{t=t_1}^{q-1} f_t(\mathbf{x}_t^I) - \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) < 0$, from (B8), we have

$$\begin{aligned}\sum_{t=q}^{s} f_t(\mathbf{x}_t^I) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant & \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{s} f_t(\mathbf{x}_t^I)} \\ \leqslant & \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^I)}\end{aligned} \tag{B13}$$

which implies

$$\begin{aligned}&\left(\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^I)\right) - \left(\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x})\right) \\ &\leqslant \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^I)}.\end{aligned} \tag{B14}$$

Applying Lemma 10 again, we have

$$\begin{aligned}&\left(\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^I)\right) - \left(\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x})\right) \\ \leqslant & 8HD^2 + \sqrt{8HD^2}\sqrt{\frac{\delta}{4H} + \sum_{t=t_1}^{q-1} f_t(\mathbf{x}) + \sum_{t=q}^{s} f_t(\mathbf{x})} \\ = & 8HD^2 + D\sqrt{2\delta + 8H\sum_{t=t_1}^{s} f_t(\mathbf{x})}.\end{aligned} \tag{B15}$$

Combining (B12) and (B15) and $\sum_{t=t_1}^{s} f_t(\mathbf{x}) = \sum_{t=1}^{s} \mathbb{I}_{[t\in I]} f_t(\mathbf{x})$, we complete the proof for (24) in Lemma 4.

Appendix C Proof of Lemma 7

Lemma 7 is derived from the proof of Lemma 2 of Luo and Schapire [41], and we include its proof for completeness.

Let $h(s,c) = \frac{\partial \exp(s^2/c)}{\partial s} = \frac{2s}{c}\exp\left(\frac{s^2}{c}\right)$. Taking the derivative of $F(s)$, we have

$$F'(s) = h(s+1,c) + h(s-1,c) - 2h(s,c') \tag{C1}$$

where $c = 3a$, $c' = 3(a-1)$. Then, applying Taylor expansion to $h(s+1,c)$ and $h(s-1,c)$ around s, and $h(s,c')$ around c, we have

$$\begin{aligned}F'(s) =& \sum_{k=1}^{\infty}\frac{1}{k!}\frac{\partial^k h(s,c)}{\partial s^k} + \sum_{k=1}^{\infty}\frac{(-1)^k}{k!}\frac{\partial^k h(s,c)}{\partial s^k} - 2\sum_{k=1}^{\infty}\frac{(c'-c)^k}{k!}\frac{\partial^k h(s,c)}{\partial c^k} \\ =& 2\sum_{k=1}^{\infty}\left(\frac{1}{(2k)!}\frac{\partial^{2k} h(s,c)}{\partial s^{2k}} - \frac{(-3)^k}{k!}\frac{\partial^k h(s,c)}{\partial c^k}\right).\end{aligned} \tag{C2}$$

To further analyze $F'(s)$, we introduce the following two lemmas.

Lemma 11. **(Lemma 3 of Luo and Schapire [41])** Let $h(s,c) = \frac{2s}{c}\exp\left(\frac{s^2}{c}\right)$. The partial derivatives of $h(s,c)$ satisfy

$$\begin{aligned}\frac{\partial^k h(s,c)}{\partial c^k} &= \exp\left(\frac{s^2}{c}\right)\sum_{j=0}^{k}(-1)^k\alpha_{k,j}\cdot\frac{s^{2j+1}}{c^{k+j+1}} \\ \frac{\partial^{2k} h(s,c)}{\partial s^{2k}} &= \exp\left(\frac{s^2}{c}\right)\sum_{j=0}^{k}\beta_{k,j}\cdot\frac{s^{2j+1}}{c^{k+j+1}}\end{aligned} \tag{C3}$$

where $\alpha_{k,j}$ and $\beta_{k,j}$ are recursively defined as

$$\begin{aligned}\alpha_{k+1,j} &= \alpha_{k,j-1} + (k+j+1)\alpha_{k,j} \\ \beta_{k+1,j} &= 4\beta_{k,j-1} + (8j+6)\beta_{k,j} + (2j+3)(2j+2)\beta_{k,j+1}\end{aligned} \tag{C4}$$

with initial values $\alpha_{0,0} = \beta_{0,0} = 2$.

Lemma 12. **(Lemma 4 of Luo and Schapire [41])** Let $\alpha_{k,j}$ and $\beta_{k,j}$ be defined as in (C4). Then $\frac{\beta_{k,j}}{(2k)!} \leqslant \frac{(d)^k\alpha_{k,j}}{k!}$ holds for all $k \geqslant 0$ and $j \in \{0,\cdots,k\}$ when $d \geqslant 3$.

Substituting (C3) into (C2), we have

$$F'(s) = 2\exp\left(\frac{s^2}{c}\right)\sum_{k=1}^{\infty}\sum_{j=0}^{k}\frac{s^{2j+1}}{c^{k+j+1}}\left(\frac{\beta_{k,j}}{(2k)!} - \frac{(3)^k\alpha_{k,j}}{k!}\right). \tag{C5}$$

Note that $\exp\left(s^2/c\right) > 0$ and $c = 3a > 0$. Then, applying Lemma 12 with $d = 3$, we complete the proof.

Appendix D Proof of Corollary 1

Because $\tau_1 \leqslant |I| \leqslant \tau_2$, we have $2^{\lceil \log \tau_1 \rceil - 1} < \tau_1 \leqslant |I| \leqslant \tau_2 \leqslant 2^{\lceil \log \tau_2 \rceil}$. Therefore, we can find a $j \in \{\lceil \log \tau_1 \rceil, \lceil \log \tau_1 \rceil + 1, \cdots, \lceil \log \tau_2 \rceil\}$ such that $2^{j-1} < |I| \leqslant 2^j$.

Then, because of $|I| \leqslant 2^j$, there must be an integer $k \geqslant 0$ such that

$$k \cdot 2^j + 1 \leqslant q \leqslant s \leqslant (k+2) \cdot 2^j \tag{D1}$$

where $[k \cdot 2^j + 1, (k+2) \cdot 2^j]$ can be divided as two consecutive intervals

$$I_1 = [k \cdot 2^j + 1, (k+1) \cdot 2^j] \text{ and } I_2 = [(k+1) \cdot 2^j + 1, (k+2) \cdot 2^j]. \tag{D2}$$

Due to $j \in \{\lceil \log \tau_1 \rceil, \lceil \log \tau_1 \rceil + 1, \cdots, \lceil \log \tau_2 \rceil\}$, we have $I_1 \in \mathcal{I}$ and $I_2 \in \mathcal{I}$. If $[q, s] \subseteq I_v, v \in \{1, 2\}$, according to (12) in Theorem 1 and (13) in Lemma 1, for any $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned}
&\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) \\
=&\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}_t^{I_v}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^{I_v}) - \sum_{t=q}^{s} f_t(\mathbf{x}) \\
\leqslant& 2\sqrt{3|I_v| \ln \frac{2\tau_2(3 + \ln(1 + 2\tau_2))}{\tau_1}} + 2\sqrt{3|I_v| \ln \frac{N(3 + \ln(1 + |I_v|))}{2}}.
\end{aligned} \tag{D3}$$

If $q \in I_1$ and $s \in I_2$, similarly, due to (12) in Theorem 1 and (13) in Lemma 1, for any $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned}
&\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) \\
=&\sum_{t \in I_1 : t \geqslant q} (f_t(\mathbf{x}_t) - f_t(\mathbf{x})) + \sum_{t \in I_2 : t \leqslant s} (f_t(\mathbf{x}_t) - f_t(\mathbf{x})) \\
\leqslant& 2\sqrt{3|I_1| \ln \frac{2\tau_2(3 + \ln(1 + 2\tau_2))}{\tau_1}} + 2\sqrt{3|I_1| \ln \frac{N(3 + \ln(1 + |I_1|))}{2}} \\
&+ 2\sqrt{3|I_2| \ln \frac{2\tau_2(3 + \ln(1 + 2\tau_2))}{\tau_1}} + 2\sqrt{3|I_2| \ln \frac{N(3 + \ln(1 + |I_2|))}{2}}.
\end{aligned} \tag{D4}$$

The proof is completed with $|I_1| = |I_2| \leqslant 2|I|$.

Appendix E Proof of Corollary 2

We complete the proof by replacing (13) used in the proof of Corollary 1 with (15) in Lemma 2.

Appendix F Proof of Corollary 3

It is easy to verify $2^{\lceil \log |I| \rceil - 1} < |I| \leqslant 2^{\lceil \log |I| \rceil}$. For brevity, let $j = \lceil \log |I| \rceil$, $k = \lfloor \frac{q-1}{2^j} \rfloor$ and $q' = k \cdot 2^j + 1$. We have

$$k \cdot 2^j + 1 \leqslant q \leqslant (k+1) \cdot 2^j \tag{F1}$$

where the first inequality is due to $k \leqslant \frac{q-1}{2^j}$ and the second inequality is due to $k + 1 = \lceil \frac{q}{2^j} \rceil \geqslant \frac{q}{2^j}$, which implies $q \in [k \cdot 2^j + 1, (k+1) \cdot 2^j]$. Combining with $s - q + 1 = |I| \leqslant 2^j$, we further have

$$k \cdot 2^j + 1 \leqslant q \leqslant s < (k+2) \cdot 2^j \tag{F2}$$

which implies $s \in [k \cdot 2^j + 1, (k+1) \cdot 2^j]$ or $s \in [(k+1) \cdot 2^j + 1, (k+2) \cdot 2^j]$. For brevity, let $I_1 = [k \cdot 2^j + 1, (k+1) \cdot 2^j]$ and $I_2 = [(k+1) \cdot 2^j + 1, (k+2) \cdot 2^j]$. Moreover, because of $|I| \in [\tau_1, \tau_2]$, we have

$$j = \lceil \log |I| \rceil \in \{\lceil \log \tau_1 \rceil, \lceil \log \tau_1 \rceil + 1, \cdots, \lceil \log \tau_2 \rceil\} \tag{F3}$$

which implies that $I_1 \in \mathcal{I}$ and $I_2 \in \mathcal{I}$.

For $s \in I_v$ where $v \in \{1, 2\}$, according to (20) in Lemma 3, for any $\mathbf{x} \in \Delta^N$, we have

$$\begin{aligned}
\sum_{t=1}^{s} \mathbb{I}_{[t \in I_v]} \left(f_t(\mathbf{x}_t^{I_v}) - f_t(\mathbf{x}) \right) \leqslant& 2\tilde{c}(|I_v|) + 2\sqrt{2\tilde{c}(|I_v|) \sum_{t=1}^{s} \mathbb{I}_{[t \in I_v]} f_t(\mathbf{x})} \\
\leqslant& 4\tilde{c}(|I_v|) + \sum_{t=1}^{s} \mathbb{I}_{[t \in I_v]} f_t(\mathbf{x}).
\end{aligned} \tag{F4}$$

If $s \in I_1$, according to (19) in Theorem 2 and (20) in Lemma 3, for any $\mathbf{x} \in \Delta^N$, we have

$$\begin{aligned}
&\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) \\
=&\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}_t^{I_1}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^{I_1}) - \sum_{t=q}^{s} f_t(\mathbf{x}) \\
\leqslant&2c + 2\sqrt{2c\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x}_t^{I_1}) + 2\tilde{c}(|I_1|)} + 2\sqrt{2\tilde{c}(|I_1|)\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x})} \\
\leqslant&2c + 2\sqrt{2c\left(4\tilde{c}(|I_1|) + 2\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x})\right) + 2\tilde{c}(|I_1|)} + 2\sqrt{2\tilde{c}(|I_1|)\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x})} \\
\leqslant&2c + 4\sqrt{2c\tilde{c}(|I_1|)} + 2\tilde{c}(|I_1|) + \left(4\sqrt{c} + 2\sqrt{2\tilde{c}(|I_1|)}\right)\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})} \\
=&\frac{a(I)}{2} + \frac{b(I)}{\sqrt{2}}\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}
\end{aligned} \tag{F5}$$

where the second inequality is due to (F4) and the last equality is due to $|I_1| = 2^j$ and the definitions of $a(I)$ and $b(I)$. Similarly, if $s \in I_2$, for any $\mathbf{x} \in \Delta^N$, we have

$$\begin{aligned}
\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) =& \sum_{t\in I_1:t\geqslant q}(f_t(\mathbf{x}_t) - f_t(\mathbf{x})) + \sum_{t\in I_2:t\leqslant s}(f_t(\mathbf{x}_t) - f_t(\mathbf{x})) \\
\leqslant&\frac{a(I)}{2} + \frac{b(I)}{\sqrt{2}}\sqrt{\sum_{t=q'}^{q'+2^j} f_t(\mathbf{x})} + \frac{a(I)}{2} + \frac{b(I)}{\sqrt{2}}\sqrt{\sum_{t=q'+2^j+1}^{s} f_t(\mathbf{x})} \\
\leqslant&a(I) + b(I)\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}
\end{aligned} \tag{F6}$$

where the last inequality is due to Cauchy-Schwarz inequality.

Appendix G Proof of Corollary 4

Let $j = \lceil \log|I| \rceil$, $k = \lfloor \frac{q-1}{2^j} \rfloor$, $q' = k\cdot 2^j + 1$, $I_1 = [k\cdot 2^j + 1, (k+1)\cdot 2^j]$ and $I_2 = [(k+1)\cdot 2^j + 1, (k+2)\cdot 2^j]$. From the proof of Corollary 3, we have $I_1, I_2 \in \mathcal{I}$, $q \in I_1$ and $s \in I_1 \cup I_2$. For $s \in I_v$ where $v \in \{1,2\}$, according to (24) in Lemma 4, for any $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned}
\sum_{t=1}^{s}\mathbb{I}_{[t\in I_v]}\left(f_t(\mathbf{x}_t^{I_v}) - f_t(\mathbf{x})\right) \leqslant& 8HD^2 + D\sqrt{2\delta + 8H\sum_{t=1}^{s}\mathbb{I}_{[t\in I_v]} f_t(\mathbf{x})} \\
\leqslant&10HD^2 + D\sqrt{2\delta} + \sum_{t=1}^{s}\mathbb{I}_{[t\in I_v]} f_t(\mathbf{x}).
\end{aligned} \tag{G1}$$

If $s \in I_1$, according to (19) in Theorem 2 and (24) in Lemma 4, for any $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned}
\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) =& \sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}_t^{I_1}) + \sum_{t=q}^{s} f_t(\mathbf{x}_t^{I_1}) - \sum_{t=q}^{s} f_t(\mathbf{x}) \\
\leqslant&2c + 2\sqrt{2c\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x}_t^{I_1})} + 8HD^2 + D\sqrt{2\delta + 8H\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x})}.
\end{aligned} \tag{G2}$$

Then, combining the above inequality with (G1), we have

$$\begin{aligned}
\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) \leqslant& 2c + 2\sqrt{2c\left(10HD^2 + D\sqrt{2\delta} + 2\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x})\right)} + 8HD^2 \\
&+ D\sqrt{2\delta + 8H\sum_{t=1}^{s}\mathbb{I}_{[t\in I_1]} f_t(\mathbf{x})} \\
\leqslant&2c + 2\sqrt{2c(10HD^2 + D\sqrt{2\delta})} + 8HD^2 + D\sqrt{2\delta} \\
&+ \left(4\sqrt{c} + \sqrt{8HD^2}\right)\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})} \\
\leqslant&3c + 28HD^2 + 3D\sqrt{2\delta} + \frac{\tilde{b}(I)}{\sqrt{2}}\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})} \\
\leqslant&\frac{\tilde{a}(I)}{2} + \frac{\tilde{b}(I)}{\sqrt{2}}\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}
\end{aligned} \tag{G3}$$

where the last two inequalities are due to the definitions of $\tilde{b}(I)$ and $\tilde{a}(I)$.

Similarly, if $s \in I_2$, for any $\mathbf{x} \in \mathcal{X}$, we have

$$\begin{aligned}\sum_{t=q}^{s} f_t(\mathbf{x}_t) - \sum_{t=q}^{s} f_t(\mathbf{x}) &= \sum_{t\in I_1: t\geqslant q} (f_t(\mathbf{x}_t) - f_t(\mathbf{x})) + \sum_{t\in I_2: t\leqslant s} (f_t(\mathbf{x}_t) - f_t(\mathbf{x})) \\ &\leqslant \frac{\tilde{a}(I)}{2} + \frac{\tilde{b}(I)}{\sqrt{2}}\sqrt{\sum_{t=q'}^{q'+2^j} f_t(\mathbf{x})} + \frac{\tilde{a}(I)}{2} + \frac{\tilde{b}(I)}{\sqrt{2}}\sqrt{\sum_{t=q'+2^j+1}^{s} f_t(\mathbf{x})} \\ &\leqslant \tilde{a}(I) + \tilde{b}(I)\sqrt{\sum_{t=q'}^{s} f_t(\mathbf{x})}\end{aligned} \tag{G4}$$

where the last inequality is due to Cauchy-Schwarz inequality.