SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

October 2022, Vol. 65 202101:1–202101:15
https://doi.org/10.1007/s11432-020-3273-9

Strongly adaptive online learning over partial intervals

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

[1]*Nationa Key Laboratory for Nove oftware echno ogy Nan ing niversity Nan ing* 210023 *China*
[2] *Paradig Inc Bei ing* 100085 *China*

Received 22 August 2020/Revised 8 February 2021/Accepted 13 April 2021/Published online 26 September 2022

Abstract To cope with changing environments, strongly adaptive algorithms that almost enjoy the optimal performance on every time interval have been proposed for on

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 the 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 environmental changes 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 the environment change regularly. In recommender systems, the environment change 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 interval.

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 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-boxes, 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 the $O(\sqrt{\tau \log T} + \sqrt{\tau \log N})$ bound in the previous work [5];
- $\mathrm{SAR}(T, \tau) = O(\sqrt{\tau \log(\tau_2/\tau_1)} + GD\sqrt{\tau})$ for OCO where D is the diameter of $\mathcal{X}$ and G is the bound of any $\|\nabla f_t(\boldsymbol{x})\|_2$, which is better than the $O(\sqrt{\tau \log T} + GD\sqrt{\tau})$ bound in 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 the learner in changing environments, the pioneering work [26] proposed adaptive regret, which is an extension of static regret and defined as

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

where $[T] = \{1, \ldots, T\}$. 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 an approximate black-box, FLH and AFLH have adaptive regret bounds $O(\sqrt{T \log T})$ and $O(\sqrt{T \log^3 T})$, respectively. Note that these bounds depend on T instead of the length of intervals, which makes no sense for intervals with small length such as $O(\sqrt{T})$.

Table 1 Comparison of strongly adaptive algorithms for LEA and general OCO, where previous regret bounds hold for any $\in [\ ]$ and our bounds hold for $\in [\ _1\ \ _2]$

Ref.	SAR( ) for LEA	SAR( ) for OCO	Number of the instances
[4]	$(\sqrt{\ }\log\ + \sqrt{\ \log\ })$	$(\sqrt{\ }\log\ + GD\sqrt{\ })$	$(\log t)$
[5, 34]	$(\sqrt{\ \log\ } + \sqrt{\ \log\ })$	$(\sqrt{\ \log\ } + GD\sqrt{\ })$	$(\log t)$
[37]	$(\sqrt{\ \log\ } + \sqrt{\ \log\ })$	$(\sqrt{\ \log\ } + GD\sqrt{\ })$	$(\log^2 t)$
[38]	$(\sqrt{\ \log\ } + \sqrt{\ \log\ })$	$(\sqrt{\ \log\ } + GD\sqrt{\ })$	$(\log\ )\quad(\log^2 t)$
This work	$(\sqrt{\ \log(\ _2\ \ _1)} + \sqrt{\ \log\ })$	$(\sqrt{\ \log(\ _2\ \ _1)} + GD\sqrt{\ })$	$(\log(\ _2\ \ _1))$

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 environment, it has $\mathrm{SAR}(T, \tau) = O(\mathrm{poly}(\log T)\, R(\tau))$, where $R(\tau)$ is the minimax regret bound for time intervals with length τ and $R(\tau) = O(\sqrt{\tau})$ 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 $\mathcal{J} = \bigcup_{\in \mathbb{N}\cup\{0\}} \mathcal{J}$, where $\mathcal{J} = \{[i\cdot 2\ , (i+1)\cdot 2\ -1] : i \in \mathbb{N}\}$;
- 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 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 evaluations 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 [4, 5, 34, 37, 38].

3 Main results

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

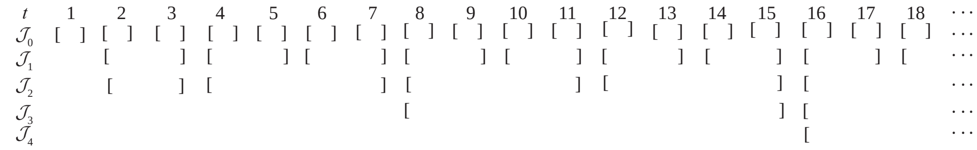


Figure 1 Illustration of GC intervals, where each interval is denoted by [ ].

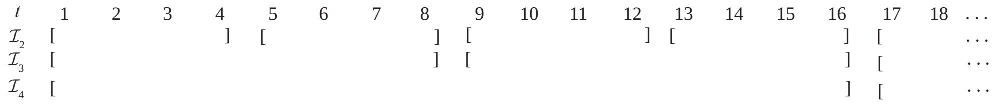


Figure 2 The refined set of intervals where $\tau_1 = 4$ and $\tau_2 = 16$.

3.1 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 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 interval, the key property of GC intervals $\mathcal{J}$ is that any interval can be partitioned into a finite sequence 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 keeping a similar property for focused intervals. Specifically, we define a smaller set of intervals

$$\mathcal{I} = \bigcup_{k=\lceil \log \tau_1 \rceil}^{\lceil \log \tau_2 \rceil} \mathcal{I}_k, \tag{3}$$

where $\mathcal{I}_k = \{[i \cdot 2^k + 1, (i+1) \cdot 2^k] : i \in \mathsf{N} \cup \{0\}\}$.

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 $[2^{\lceil \log \tau_1 \rceil}, 2^{\lceil \log \tau_2 \rceil}]$, 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

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

In each round $t = 1, \ldots, T$, each instance $\mathcal{B}_I$, $\forall I \in \mathrm{Active}(t)$ 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 3.4847(e-723(c)-1.66442(a.930723(o)-29.961(a)-5.8901732.56T[(],)-51

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

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

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

According to the definition of potential function (5), it is easy to derive an upper bound of G_t as $G_t \leqslant \sqrt{3S_t \ln \Phi(G_t, S_t)}$. 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. 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 $|G_t| = \sqrt{3S_t \ln \Phi(G_t, S_t)}$. 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 $\boldsymbol{c}_t(i) = W(G_{t-1}, S_{t-1})$. However, with the new potential function, the value of $\boldsymbol{c}_t(i)$ could be negative, which motivates the following two modifications. First, to ensure $\boldsymbol{x}_t \in \Delta_N$ where $\Delta_N = \{\boldsymbol{x}: \sum_{i=1}^{N} \boldsymbol{x}(i) = 1\}$, the normalized weight is redefined as $\boldsymbol{x}_t(i) \propto [\boldsymbol{c}_t(i)]_+$. Second, to ensure $\sum_{i=1}^{N} \tilde{\boldsymbol{g}}_t(i)\boldsymbol{c}_t(i) \leqslant 0$ that is essential for upper bounding $\Phi(G_t, S_t)$, we redefine

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

and recall $G_{t-1} = \sum_{k=1}^{t-1} \tilde{\boldsymbol{g}}_k(i)$, $S_{t-1} = \sum_{k=1}^{t-1} |\tilde{\boldsymbol{g}}_k(i)|$. 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.

Algorithm 1 Modified AdaNormalHedge

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

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 $\boldsymbol{x}_t^I$ generated by instance $\mathcal{B}_I$, we further define

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

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

$$\boldsymbol{x}_t = \sum_{I \in \text{Active}(t)} p_t^I \boldsymbol{x}_t^I. \tag{11}$$

The detailed procedures of our meta-algorithm are summarized in Algorithm 2 and this algorithm is called 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 $\mathcal{I}_i$, 5Tf3.120Td[(i)8.23599]TJF

Algorithm 2 Strongly adaptive [redacted]

1: **Input:** A black-box algorithm [redacted]

2: $\mathcal{I}$ =0cmB6.6Td [00796.9738/I [redacted]
1 1 1194.421(-210.96-91-3.52 [redacted]

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

Under Assumptions 1–3, we bound the regret of SOGD in Lemma 2.

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

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

Then, combining Theorem 1 and Lemma 2, we can obtain Corollary 2.

Corollary 2. Let $c = 3\ln(2\tau$

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 $\boldsymbol{x} \in \Delta$ and $I = [q, s]$ with length $|I| \in [\tau_1, \tau_2]$, our Algorithm 2 using Algorithm 1 as its black-box satisfies

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

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

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

Because of $s - q' + 1 \leqslant 2^{j+1} \leqslant 4|I|$, we have $\sqrt{\sum_{t=q'}^{s} f_t(\boldsymbol{x})} \leqslant \sqrt{\sum_{t=q'}^{s} 1} = O(\sqrt{|I|})$, 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, $\sqrt{\sum_{t=q'}^{s} f_t(\boldsymbol{x})}$, which is a relaxation of $\sqrt{\sum_{t=q}^{s} f_t(\boldsymbol{x})}$, 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 Assumption 4 about the smoothness of the loss functions.

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

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

Then, we bound the regret of Algorithm 3 in Lemma 4.

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

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

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 $\boldsymbol{x} \in \mathcal{X}$ 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

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

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

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

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

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

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

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

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

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

where the last inequality is due to $\text{Active}(t)\subseteq\mathcal{I}'$, $\boldsymbol{x}_t=\sum_{I'\in\text{Active}(t)}p_t^{I'}\boldsymbol{x}_t^{I'}$ and Jensen's inequality. To bound the last term in (28), we further introduce Lemma 6.

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

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

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

$$\sum_{t=q'}^{s'}\frac{|\tilde{r}_t^{I'}|}{(C_{t-1}^{I'}+1)}\leqslant 1+\int_1^{1+C_{s'}^{I'}}\frac{1}{x}dx=1+\ln(1+C_{s'}^{I'}). \tag{32}$$

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

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

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

$$|R^I|=\sqrt{3C^I\ln\Phi(R^I,C^I)}\leqslant\sqrt{3C^I\ln\sum_{I'=[q',s']\in\mathcal{I}'}\Phi(R^{I'}_{\wedge s'},C^{I'}_{\wedge s'})}\leqslant\sqrt{cC^I}. \tag{34}$$

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

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

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

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

$$\begin{aligned}C^I&=\sum_{t=1}\mathbb{I}_{[t\in I]}|\tilde{r}_t^I|=\sum_{t=1}\mathbb{I}_{[t\in I]}(\tilde{r}_t^I+2[-\tilde{r}_t^I]_+)\\&=R^I+2\sum_{t=1}\mathbb{I}_{[t\in I]}[-\tilde{r}_t^I]_+\leqslant R^I+2L^I,\end{aligned} \tag{36}$$

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

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

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

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

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

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

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

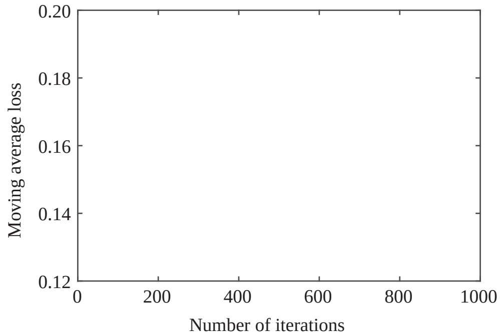
$$\frac{(\mathrm{R}_{t-1}^{I}+\mathrm{r})^2}{\mathrm{C}_{t-1}^{I}+|\mathrm{r}|}=(\mathrm{C}_{t-1}^{I}+\mathrm{r})+\frac{(\mathrm{R}_{t-1}^{I}-\mathrm{C}_{t-1}^{I})^2}{\mathrm{C}_{t-1}^{I}+\mathrm{r}}+2(\mathrm{R}_{t-1}^{I}-\mathrm{C}_{t-1}^{I}), \tag{40}$$

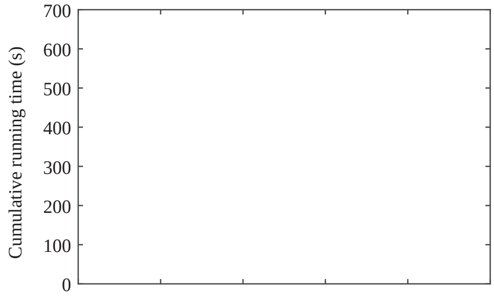
when $\mathrm{r}\in[0,1]$ and

$$\frac{(\mathrm{R}_{t-1}^{I}+\mathrm{r})^2}{\mathrm{C}_{t-1}^{I}+|\mathrm{r}|}=(\mathrm{C}_{t-1}^{I}-\mathrm{r})+\frac{(\mathrm{R}_{t-1}^{I}+\mathrm{C}_{t-1}^{I})^2}{\mathrm{C}_{t-1}^{I}-\mathrm{r}}-2(\mathrm{R}_{t-1}^{I}+\mathrm{C}_{t-1}^{I}), \tag{41}$$

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

Furthermore, we defi(ne a function $\mathrm{h}(\mathrm{r})$:97)) $^{I}_{-1}$)j|))Id–


0.20
0.18
0.16
0.14
0.12
Moving average loss
0
200
400
600
800
1000
Number of iterations


700
600
500
400
300
200
100
0
Cumulative running time (s)

18 Hou B J, Zhang L J, Zhou Z H. Learning with feature evolvable streams. In: Proceedings of 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 Marketing 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. FNT Optim, 2015, 2: 157–325

24 Shalev-Shwartz S. Online learning and online convex optimization. FNT 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: Proceedings of 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, Pál 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: Proceedings of 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: Proceedings of 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: Proceedings of 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