

# Online Learning in Changing Environments

Lijun Zhang

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China  
 zhanglj@lamda.nju.edu.cn

## Abstract

The usual goal of online learning is to minimize the regret, which measures the performance of online learner against a fixed comparator. However, it is not suitable for changing environments in which the best decision may change over time. To address this limitation, new performance measures, including dynamic regret and adaptive regret have been proposed to guide the design of online algorithms. In dynamic regret, the learner is compared with a sequence of comparators, and in adaptive regret, the learner is required to minimize the regret over every interval. In this paper, we will review the recent developments in this area, and highlight our contributions. Specifically, we have proposed novel algorithms to minimize the dynamic regret and adaptive regret, and investigated the relationship between them.

## 1 Introduction

Online learning is a well established learning paradigm which has both theoretical and practical appeals [Shalev-Shwartz, 2011]. It is performed in a sequence of consecutive rounds, where at iteration  $t$  the learner chooses a decision  $\mathbf{w}_t$  from a set  $\mathcal{W}$ . After committing to this choice, a cost function  $f_t : \mathcal{W} \mapsto \mathbb{R}$  is revealed and the learner suffers a loss  $f_t(\mathbf{w}_t)$ . The goal of online learning is to minimize the cumulative loss  $\sum_{t=1}^T f_t(\mathbf{w}_t)$ , and the traditional performance measure is regret:

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

defined as the difference between the cumulative loss of the online learner and that of the best decision chosen in hindsight. When both the set  $\mathcal{W}$  and the function  $f_t(\cdot)$  are convex, it becomes online convex optimization (OCO) [Zinkevich, 2003].

In the literature, there exist plenty of algorithms and theories for minimizing the regret [Cesa-Bianchi and Lugosi, 2006; Hazan, 2016]. However, when the environment is changing, regret is not a suitable measure, since it compares the learner against a fixed decision. And for the same

reason, the definition in (1) is also referred to as *static* regret. To address this limitation, recent studies have introduced new performance measures, including *dynamic* regret and *adaptive* regret, to measure the performance of online learner.

In dynamic regret, the learner is compared against a sequence of comparators  $\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{W}$  [Zinkevich, 2003], i.e.,

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

It is well-known that in the worst case, a sublinear dynamic regret is impossible. The dynamic regret is usually bounded in terms of some regularities on the comparator sequence or the function sequence [Besbes *et al.*, 2015; Jadbabaie *et al.*, 2015]. As shown in (2), dynamic regret deals with changing environments from a global perspective, as it measures the performance over the whole interval  $[1, T]$  but allows the comparator to change over time.

On the other hand, adaptive regret takes a local perspective, and aims to minimize the static regret over *every* interval. Given a parameter  $\tau$ , the strong version of adaptive regret is defined as the maximum static regret over intervals of length  $\tau$  [Daniely *et al.*, 2015], i.e.,

$$\begin{aligned} & \text{SA-Regret}(T, \tau) \\ &= \max_{[r, r+\tau-1] \subseteq [T]} \left( \sum_{t=r}^{r+\tau-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=r}^{r+\tau-1} f_t(\mathbf{w}) \right). \end{aligned} \quad (3)$$

Note that the optimal decision for different intervals could be different. By minimizing the adaptive regret, we essentially enforce the learner to compete with a changing comparator.

In this paper, we will review the recent developments on dynamic regret and adaptive regret under the setting of OCO, including our contributions. Specifically, we contribute to this area in the following aspects.

- We have considered the worst-case of dynamic regret in (2), and exploited the curvature of the functions, such as smoothness and strong convexity, to improve the regret bound [Yang *et al.*, 2016; Zhang *et al.*, 2017].
- For the general definition of dynamic regret, we propose a novel algorithm named Ader, which attains an  $O(\sqrt{T(1 + P_T)})$  dynamic regret, where  $P_T$  is the path-length of the comparator sequence [Zhang *et al.*, 2018a].

We also prove an  $\Omega(\sqrt{T(1+P_T)})$  lower bound, showing that our upper bound is optimal.

- For adaptive regret, we reduce the number of gradient evaluations in the  $t$ -th round from  $O(\log t)$  to 1 without affecting the performance [Wang *et al.*, 2018], and make use of smoothness to establish problem-dependent bounds [Zhang *et al.*, 2019a]. We also develop a universal algorithm that is able to minimize the adaptive regret of multiple types of convex functions simultaneously [Zhang *et al.*, 2019b].
- To investigate the relationship between dynamic regret and adaptive regret, we demonstrate that the worst-case dynamic regret can be upper bounded in terms of the adaptive regret and the functional variation [Zhang *et al.*, 2018b]. Furthermore, we develop novel algorithms that are able to minimize the general dynamic regret and adaptive regret simultaneously [Zhang *et al.*, 2020].

## 2 Dynamic Regret

We first discuss the worst-case dynamic regret, and then move to the general one.

### 2.1 Worst-case Dynamic Regret

Under the setting of OCO, most of previous studies about dynamic regret focus on the worst-case of (2), in which the sequence of comparators consists of minimizers of online functions [Jadbabaie *et al.*, 2015], i.e.,

$$\begin{aligned} \text{D-Regret}(\mathbf{w}_1^*, \dots, \mathbf{w}_T^*) &= \sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T f_t(\mathbf{w}_t^*) \\ &= \sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T \min_{\mathbf{w} \in \mathcal{W}} f_t(\mathbf{w}) \end{aligned}$$

where  $\mathbf{w}_t^* \in \operatorname{argmin}_{\mathbf{w} \in \mathcal{W}} f_t(\mathbf{w})$  is a minimizer of  $f_t(\cdot)$  over  $\mathcal{W}$ .

Let  $P_T^*$  be the path-length of the minimizer sequence, i.e.,

$$P_T^* = \sum_{t=2}^T \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2.$$

When the loss functions are strongly convex and smooth, Mokhtari *et al.* [2016] show that online gradient descent (OGD) ensures  $\text{D-Regret}(\mathbf{w}_1^*, \dots, \mathbf{w}_T^*) = O(P_T^*)$ . We have demonstrated that this rate is also attainable for convex and smooth functions under the condition that the minimizers lie in the interior of  $\mathcal{W}$  [Yang *et al.*, 2016]. Furthermore, when the minimizer of each function is unique, a simple baseline which plays the minimizer of the previous round, i.e.,

$$\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w} \in \mathcal{W}} f_t(\mathbf{w}),$$

attains an  $O(P_T^*)$  worst-case dynamic regret.

Besides the path-length, we have introduced the squared path-length [Zhang *et al.*, 2017]:

$$S_T^* = \sum_{t=1}^T \|\mathbf{w}_{t+1}^* - \mathbf{w}_t^*\|_2^2$$

as a new regularity of the comparator sequence. The advantage of  $S_T^*$  is that it could be much smaller than  $P_T^*$  if the minimizer moves slowly. When all the functions are strongly convex and smooth, we propose to apply gradient descent multiple times in each round, and demonstrate that the dynamic regret could be reduced from  $O(P_T^*)$  to  $O(\min(P_T^*, S_T^*))$ . We then extend our theoretical guarantee to semi-strongly convex or self-concordant functions.

Instead of measuring the complexity of the comparator sequence, Besbes *et al.* [2015] propose to evaluate the movement of the loss functions as follows:

$$F_T = \sum_{t=1}^T \sup_{\mathbf{w} \in \mathcal{W}} |f_{t+1}(\mathbf{w}) - f_t(\mathbf{w})|. \quad (4)$$

Besbes *et al.* show that a restarted OGD algorithm equipped with a prior knowledge of an upper bound  $V_T \geq F_T$  achieves  $O(V_T^{1/3} T^{2/3})$  and  $O(\log T \sqrt{V_T T})$  dynamic regret for convex functions and strongly convex functions, respectively. However, these bounds depend on the predetermined  $V_T$  rather than the actual  $F_T$ .

### 2.2 General Dynamic Regret

Although one can show that  $\text{D-Regret}(\mathbf{w}_1^*, \dots, \mathbf{w}_T^*) \geq \text{D-Regret}(\mathbf{u}_1, \dots, \mathbf{u}_T)$  for any  $\mathbf{u}_1, \dots, \mathbf{u}_T \in \mathcal{W}$ , it does not imply the former one is stronger since an upper bound for  $\text{D-Regret}(\mathbf{w}_1^*, \dots, \mathbf{w}_T^*)$  could be very loose for  $\text{D-Regret}(\mathbf{u}_1, \dots, \mathbf{u}_T)$ . In fact, the definition in (2) is more general since it holds for any sequence of comparators, and thus includes the static regret and the worst-case dynamic regret as special cases.

In his seminal work, Zinkevich [2003] shows that OGD with a constant step size attains a dynamic regret of  $O(\sqrt{T}(1+P_T))$  for any sequence  $\mathbf{u}_1, \dots, \mathbf{u}_T$ , where

$$P_T = \sum_{t=1}^T \|\mathbf{u}_{t+1} - \mathbf{u}_t\|_2.$$

This upper bound is adaptive in the sense that it automatically becomes tighter when the comparators change slowly. Hall and Willett [2013] have investigated the general dynamic regret when a dynamic model is available to predict the comparator sequence.

For the general dynamic regret, we have established an  $\Omega(\sqrt{T}(1+P_T))$  lower bound, which indicates that results of Zinkevich [2003] and Hall and Willett [2013] are far away from the optimum. To address this limitation, we develop an optimal algorithm, namely adaptive learning for dynamic environment (Ader), which attains an  $O(\sqrt{T}(1+P_T))$  dynamic regret. In the following, we give a brief description of Ader.

### 2.3 Ader

Ader follows the framework of prediction with expert advice (PAE) [Cesa-Bianchi and Lugosi, 2006], and is inspired by the strategy of maintaining multiple learning rates in Meta-Grad [van Erven and Koolen, 2016]. The basic idea is to run multiple OGD algorithms in parallel, each with a different step size that is optimal for a specific path-length, and combine them with an expert-tracking algorithm.

---

**Algorithm 1** Ader: Meta-algorithm

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

- 1: Activate a set of experts  $\{E^\eta | \eta \in \mathcal{H}\}$  by invoking Algorithm 2 for each step size  $\eta \in \mathcal{H}$
- 2: Sort step sizes in ascending order  $\eta_1 \leq \eta_2 \leq \dots \leq \eta_N$ , and set  $\omega_1^{\eta_i} = \frac{C}{i(i+1)}$
- 3: **for**  $t = 1, \dots, T$  **do**
- 4:   Receive  $\mathbf{w}_t^\eta$  from each expert  $E^\eta$
- 5:   Output

$$\mathbf{w}_t = \sum_{\eta \in \mathcal{H}} \omega_t^\eta \mathbf{w}_t^\eta$$

- 6:   Observe the loss function  $f_t(\cdot)$
- 7:   Update the weight of each expert by

$$\omega_{t+1}^\eta = \frac{\omega_t^\eta e^{-\alpha f_t(\mathbf{w}_t^\eta)}}{\sum_{\mu \in \mathcal{H}} \omega_t^\mu e^{-\alpha f_t(\mathbf{w}_t^\mu)}}$$

- 8:   Send gradient  $\nabla f_t(\mathbf{w}_t^\eta)$  to each expert  $E^\eta$
- 9: **end for**

---

**Meta-algorithm** Our meta-algorithm, summarized in Algorithm 1, is built upon the exponentially weighted average forecaster [Cesa-Bianchi and Lugosi, 2006]. The inputs of the meta-algorithm are its own step size  $\alpha$ , and a set  $\mathcal{H}$  of step sizes for experts. In Step 1, we active a set of experts  $\{E^\eta | \eta \in \mathcal{H}\}$  by invoking the expert-algorithm for each  $\eta \in \mathcal{H}$ . In Step 2, we set the initial weight of each expert. Let  $\eta_i$  be the  $i$ -th smallest step size in  $\mathcal{H}$ . The weight of  $E^{\eta_i}$  is chosen as

$$\omega_1^{\eta_i} = \frac{C}{i(i+1)}, \text{ and } C = 1 + \frac{1}{|\mathcal{H}|}. \quad (5)$$

In each round, the meta-algorithm receives a set of predictions  $\{\mathbf{w}_t^\eta | \eta \in \mathcal{H}\}$  from all experts (Step 4), and outputs the weighted average (Step 5):

$$\mathbf{w}_t = \sum_{\eta \in \mathcal{H}} \omega_t^\eta \mathbf{w}_t^\eta$$

where  $\omega_t^\eta$  is the weight assigned to expert  $E^\eta$ . After observing the loss function, the weights of experts are updated according to the exponential weighting scheme (Step 7). In the last step, we send the gradient  $\nabla f_t(\mathbf{w}_t^\eta)$  to each expert  $E^\eta$  so that they can update their own predictions.

**Expert-algorithm** As shown in Algorithm 2, we use OGD as the expert-algorithm, which takes the step size  $\eta$  as its input. In Step 3 of Algorithm 2, each expert submits its prediction  $\mathbf{w}_t^\eta$  to the meta-algorithm, and receives the gradient  $\nabla f_t(\mathbf{w}_t^\eta)$  in Step 4. Then, in Step 5 it performs gradient descent

$$\mathbf{w}_{t+1}^\eta = \Pi_{\mathcal{W}}[\mathbf{w}_t^\eta - \eta \nabla f_t(\mathbf{w}_t^\eta)]$$

to get the prediction for the next round.

Next, we specify the parameter setting and our dynamic regret. The set  $\mathcal{H}$  is constructed in the way such that for any possible sequence of comparators, there exists a step size that

---

**Algorithm 2** Ader: Expert-algorithm

**Require:** The step size  $\eta$

- 1: Let  $\mathbf{w}_1^\eta$  be any point in  $\mathcal{W}$
- 2: **for**  $t = 1, \dots, T$  **do**
- 3:   Submit  $\mathbf{w}_t^\eta$  to the meta-algorithm
- 4:   Receive gradient  $\nabla f_t(\mathbf{w}_t^\eta)$  from the meta-algorithm
- 5:

$$\mathbf{w}_{t+1}^\eta = \Pi_{\mathcal{W}}[\mathbf{w}_t^\eta - \eta \nabla f_t(\mathbf{w}_t^\eta)]$$

- 6: **end for**

---

is nearly optimal. To control the size of  $\mathcal{H}$ , we use a geometric series with ratio 2:

$$\mathcal{H} = \left\{ \eta_i = \frac{2^{i-1}D}{G} \sqrt{\frac{7}{2T}} \mid i = 1, \dots, N \right\} \quad (6)$$

where  $G$  is the upper bound of the norm of gradients,  $D$  is the upper bound of the diameter of  $\mathcal{W}$ , and  $N = \lceil \frac{1}{2} \log_2(1 + 4T/7) \rceil + 1$ . Then, with an appropriate choice of  $\alpha$ , we can show that Ader is equipped with an optimal dynamic regret:

$$\text{D-Regret}(\mathbf{u}_1, \dots, \mathbf{u}_T) = O\left(\sqrt{T(1 + P_T)}\right).$$

With the help of surrogate losses, we have proposed an improved Ader and reduced the number of gradient evaluations per round from  $N = O(\log T)$  to 1. Furthermore, we have extended Ader to the setting that a sequence of dynamical models is available to characterize the comparators.

### 3 Adaptive Regret

Adaptive regret has been studied under the settings of PAE [Littlestone and Warmuth, 1994; György *et al.*, 2012] and OCO. In this paper, we focus on the later one.

The concept of adaptive regret is formally introduced by Hazan and Seshadhri [2007] to OCO, and later refined by Daniely *et al.* [2015]. To distinguish between them, we refer to the definition of Hazan and Seshadhri as weakly adaptive regret:

$$\text{WA-Regret}(T) = \max_{[r,s] \subseteq [T]} \left( \sum_{t=r}^s f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=r}^s f_t(\mathbf{w}) \right)$$

which is the maximum regret over any contiguous interval. They develop a novel algorithm named as follow the leading history (FLH), which runs an instance of low-regret algorithm in each round as an expert, and then combines them with an expert-tracking method. To improve the efficiency, Hazan and Seshadhri deploy a data-streaming technique to prune the set of experts, and as a result only  $O(\log t)$  experts are stored at round  $t$ . The efficient version of FLH attains  $O(d \log^2 T)$  and  $O(\sqrt{T \log^3 T})$  adaptive regrets for exponentially concave (abbr. exp-concave) functions and convex functions, respectively [Hazan and Seshadhri, 2009]. For strongly convex functions, we have designed an efficient algorithm with an  $O(\log^2 T)$  adaptive regret [Zhang *et al.*, 2018b].

However, the weakly adaptive regret does not respect short intervals well. To avoid this limitation, Daniely *et al.* [2015]

propose the strongly adaptive regret in (3), which emphasizes the dependence on the interval length. For convex functions, Daniely *et al.* [2015] establish an  $O(\sqrt{\tau \log T})$  strongly adaptive regret. In a subsequent work, Jun *et al.* [2017a] improve the strongly adaptive regret to  $O(\sqrt{\tau \log T})$ . Jun *et al.* [2017b] have investigated how to exploit smoothness to further improve the adaptive regret, and we go one step further by making the bound fully problem-dependent [Zhang *et al.*, 2019a]. All the previous algorithms designed for adaptive regret require to query the gradient at least  $O(\log t)$  times in the  $t$ -th round. To address this limitation, we propose to use surrogate losses to reduce the number of gradient evaluations per round to 1 [Wang *et al.*, 2018].

Existing algorithms for adaptive regret lack universality in the sense that they can only handle one type of convex functions and need apriori knowledge of parameters. By contrast, there exist universal algorithms, such as MetaGrad [van Erven and Koolen, 2016], that attain optimal static regret for multiple types of convex functions simultaneously. Along this line of research, we have proposed the *first* universal algorithm for minimizing the adaptive regret of convex functions [Zhang *et al.*, 2019b]. Due to limitations of space, we take exp-concave functions as an example to illustrate the key idea in the next section.

### 3.1 A Universal Algorithm for Adaptive Regret

Inspired by MetaGrad, we constructs surrogate losses parameterized by a learning rate  $\eta$ :

$$\ell_t^\eta(\mathbf{w}) = -\eta \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w} \rangle + \eta^2 \langle \nabla f_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w} \rangle^2.$$

By considering different values of  $\eta$ , we can handle the uncertainty of the modulus of exp-concavity. To minimize the adaptive regret, we adopt the idea of sleeping experts [Freund *et al.*, 1997]. Specifically, we make use of the geometric covering (GC) intervals [Daniely *et al.*, 2015] defined as  $\mathcal{I} = \bigcup_{k \in \mathbb{N} \cup \{0\}} \mathcal{I}_k$ , where

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

Then, we only focus on intervals in  $\mathcal{I}$ . For each interval  $I = [r, s] \in \mathcal{I}$ , we will create  $1 + \lceil \frac{1}{2} \log_2(s-r+1) \rceil$  experts, each of which minimizes one surrogate loss in  $\{\ell_t^\eta(\mathbf{w}) | \eta \in \mathcal{S}(s-r+1)\}$  during  $I$ , where

$$\mathcal{S}(\tau) = \left\{ \frac{2^{-i}}{5DG} \mid i = 0, 1, \dots, \left\lceil \frac{1}{2} \log_2 \tau \right\rceil \right\}.$$

These experts become active in round  $r$  and will be removed forever after round  $s$ . It is easy to verify that the surrogate loss  $\ell_t^\eta(\cdot)$  is exp-concave. So, we can apply online Newton step (ONS) [Hazan *et al.*, 2007] as the expert-algorithm to minimize  $\ell_t^\eta(\cdot)$  during interval  $I$ .

To combine the solutions of active experts, we extend the meta-algorithm of MetaGrad, i.e., Tilted Exponentially Weighted Average (TEWA), to sleeping experts. Let  $\mathcal{A}_t$  be the set consisting of all the active experts, and  $E_I^\eta$  be the expert for an interval  $I \in \mathcal{I}$  with learning rate  $\eta$ . Denote the prediction of expert  $E_I^\eta$  at round  $t$  as  $\mathbf{w}_{t,I}^\eta$ . Specifically, we submit the following decision in the  $t$ -th round

$$\mathbf{w}_t = \frac{1}{\sum_{E_I^\eta \in \mathcal{A}_t} \exp(-L_{t-1,I}^\eta \eta)} \sum_{E_I^\eta \in \mathcal{A}_t} \exp(-L_{t-1,I}^\eta \eta) \mathbf{w}_{t,I}^\eta$$

where  $L_{t-1,I}^\eta$  is the cumulative loss of  $E_I^\eta$ .

Our theoretical analysis shows that the above algorithm attains an  $O(\frac{d}{\alpha} \log \tau \log T)$  strongly adaptive regret for  $\alpha$ -exp-concave functions. By introducing an additional surrogate loss for strongly convex functions, we obtain an  $O(\frac{1}{\lambda} \log \tau \log T)$  strongly adaptive regret for  $\lambda$ -strongly convex functions. Meanwhile, the algorithm also ensures an  $O(\sqrt{\tau \log T})$  bound for convex functions.

## 4 Relationship between Dynamic Regret and Adaptive Regret

While both the dynamic and adaptive regrets aim at coping with changing environments, little is known about their relationship. We make two contributions in this direction.

First, we show that the strongly adaptive regret in (3), together with the functional variation, can be used to upper bound the worst-case dynamic regret [Zhang *et al.*, 2018b]. Let  $I_1 = [r_1, s_1], \dots, I_k = [r_k, s_k]$  be a partition of  $[1, T]$  and for each interval  $I_i$ , and define the local variation of functions as

$$F_T(i) = \sum_{t=r_i}^{s_i-1} \sup_{\mathbf{w} \in \mathcal{W}} |f_{t+1}(\mathbf{w}) - f_t(\mathbf{w})|.$$

We have proved that

$$\begin{aligned} & \text{D-Regret}(\mathbf{w}_1^*, \dots, \mathbf{w}_T^*) \\ & \leq \min_{I_1, \dots, I_k} \sum_{i=1}^k (\text{SA-Regret}(T, |I_i|) + 2|I_i| \cdot F_T(i)). \end{aligned}$$

Thus, an algorithm with a small strongly adaptive regret is automatically equipped with a tight dynamic regret. As a result, we obtain a series of algorithms for minimizing the dynamic regret that do not need any prior knowledge of the functional variation.

Second, we have proposed novel algorithms that are able to minimize the general dynamic regret and adaptive regret simultaneously [Zhang *et al.*, 2020]. In fact, our theoretical guarantee is even stronger in the sense that one algorithm is able to minimize the dynamic regret over any interval.

## 5 Open Problems

Although significant progress has been made in dynamic regret and adaptive regret, there still remain many open problems in this area.

- In the analysis of the worst-case dynamic regret, we can exploit the curvature of functions, such as strong convexity and smoothness, to improve the bound. However, whether these conditions can be used to tighten the general dynamic regret remains open.
- Although we have shown that the worst-case dynamic regret can be upper bounded by the adaptive regret, the relationship between the general dynamic regret and the adaptive regret is unclear.

## Acknowledgments

This work was partially supported by NSFC (61921006), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

## References

- [Besbes *et al.*, 2015] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations Research*, 63(5):1227–1244, 2015.
- [Cesa-Bianchi and Lugosi, 2006] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [Daniely *et al.*, 2015] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1405–1411, 2015.
- [Freund *et al.*, 1997] Yoav Freund, Robert E. Schapire, Yooram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 334–343, 1997.
- [György *et al.*, 2012] András György, Tamás Linder, and Gábor Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012.
- [Hall and Willett, 2013] Eric C. Hall and Rebecca M. Willett. Dynamical models and tracking regret in online convex programming. In *Proceedings of the 30th International Conference on Machine Learning*, pages 579–587, 2013.
- [Hazan and Seshadhri, 2007] Elad Hazan and C. Seshadhri. Adaptive algorithms for online decision problems. *Electronic Colloquium on Computational Complexity*, 88, 2007.
- [Hazan and Seshadhri, 2009] Elad Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 393–400, 2009.
- [Hazan *et al.*, 2007] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- [Hazan, 2016] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.
- [Jadbabaie *et al.*, 2015] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.
- [Jun *et al.*, 2017a] Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Improved strongly adaptive online learning using coin betting. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 943–951, 2017.
- [Jun *et al.*, 2017b] Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Online learning for changing environments using coin betting. *Electronic Journal of Statistics*, 11(2):5282–5310, 2017.
- [Littlestone and Warmuth, 1994] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- [Mokhtari *et al.*, 2016] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 7195–7201, 2016.
- [Shalev-Shwartz, 2011] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.
- [van Erven and Koolen, 2016] Tim van Erven and Wouter M Koolen. MetaGrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29*, pages 3666–3674, 2016.
- [Wang *et al.*, 2018] Guanghui Wang, Dakuan Zhao, and Lijun Zhang. Minimizing adaptive regret with one gradient per iteration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2762–2768, 2018.
- [Yang *et al.*, 2016] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 449–457, 2016.
- [Zhang *et al.*, 2017] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems 30*, pages 732–741, 2017.
- [Zhang *et al.*, 2018a] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31*, pages 1323–1333, 2018.
- [Zhang *et al.*, 2018b] Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.
- [Zhang *et al.*, 2019a] Lijun Zhang, Tie-Yan Liu, and Zhi-Hua Zhou. Adaptive regret of convex and smooth functions. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7414–7423, 2019.
- [Zhang *et al.*, 2019b] Lijun Zhang, Guanghui Wang, Wei-Wei Tu, and Zhi-Hua Zhou. Dual adaptivity: A universal algorithm for minimizing the adaptive regret of convex functions. *ArXiv e-prints*, arXiv:1906.10851, 2019.
- [Zhang *et al.*, 2020] Lijun Zhang, Shiyin Lu, and Tianbao Yang. Minimizing dynamic regret and adaptive regret simultaneously. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, 2020.
- [Zinkevich, 2003] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.