Trading-Off Static and Dynamic Regret in Online Least-Squares and Beyond[*]

Jianjun Yuan,[1] **Andrew Lamperski**[1]
[1]Department of Electrical and Computer Engineering, University of Minnesota
200 Union Street SE, 4-174 Keller Hall
Minneapolis, MN 55455, US
{yuanx270, alampers}@umn.edu

Abstract

Recursive least-squares algorithms often use forgetting factors as a heuristic to adapt to non-stationary data streams. The first contribution of this paper rigorously characterizes the effect of forgetting factors for a class of online Newton algorithms. For exp-concave and strongly convex objectives, the algorithms achieve the dynamic regret of $\max\{O(\log T), O(\sqrt{TV})\}$, where V is a bound on the path length of the comparison sequence. In particular, we show how classic recursive least-squares with a forgetting factor achieves this dynamic regret bound. By varying V, we obtain a trade-off between static and dynamic regret. In order to obtain more computationally efficient algorithms, our second contribution is a novel gradient descent step size rule for strongly convex functions. Our gradient descent rule recovers the order optimal dynamic regret bounds described above. For smooth problems, we can also obtain static regret of $O(T^{1-\beta})$ and dynamic regret of $O(T^{\beta}V^*)$, where $\beta \in (0, 1)$ and V^* is the path length of the sequence of minimizers. By varying β, we obtain a trade-off between static and dynamic regret.

Introduction

Online learning algorithms are designed to solve prediction and learning problems for streaming data or batch data whose volume is too large to be processed all at once. Applications include online auctions (Blum et al. 2004), online classification and regression (Crammer et al. 2006), online subspace identification (Yuan and Lamperski 2019), as well as online resource allocation (Yuan and Lamperski 2018).

The general procedure for online learning algorithms is as follows: at each time t, before the true time-dependent objective function $f_t(\theta)$ is revealed, we need to make the prediction, θ_t, based on the history of the observations $f_i(\theta)$, $i < t$. Then the value of $f_t(\theta_t)$ is the loss suffered due to the lack of the knowledge for the true objective function $f_t(\theta)$. Our prediction of θ is then updated to include the information of $f_t(\theta)$. This whole process is repeated until termination. The functions, $f_t(\theta)$, can be chosen from a function class in an arbitrary, possibly adversarial manner.

The performance of an online learning algorithm is typically assessed using various notions of *regret*. *Static regret*, $\mathcal{R}_s$, measures the difference between the algorithm's cumulative loss and the cumulative loss of the best fixed decision in hindsight (Cesa-Bianchi and Lugosi 2006):

$$\mathcal{R}_s = \sum_{t=1}^{T} f_t(\theta_t) - \min_{\theta \in \mathcal{S}} \sum_{t=1}^{T} f_t(\theta),$$

where $\mathcal{S}$ is a constraint set. For convex functions, variations of gradient descent achieve static regret of $O(\sqrt{T})$, while for strongly convex functions these can be improved to $O(\log T)$ (Hazan 2016). However, when the underlying environment is changing, due to the fixed comparator the algorithms converge to, static regret is no longer appropriate.

In order to better track the changes of the underlying environment, *dynamic regret* is proposed to compare the cumulative loss against that incurred by a comparison sequence, $z_1, \ldots, z_T \in \mathcal{S}$:

$$\mathcal{R}_d = \sum_{t=1}^{T} f_t(\theta_t) - \sum_{t=1}^{T} f_t(z_t)$$

The classic work on online gradient descent (Zinkevich 2003) achieves dynamic regret of the order $O(\sqrt{T}(1 + V))$, where V is a bound on the path length of the comparison sequence:

$$\sum_{t=2}^{T} \|z_t - z_{t-1}\| \leq V.$$

This has been improved to $O(\sqrt{T(1 + V)})$ in (Zhang, Lu, and Zhou 2018) by applying a meta-optimization over step sizes.

In works such as (Mokhtari et al. 2016; Yang et al. 2016), it is assumed that $z_t = \theta_t^* = \operatorname{argmin}_{\theta \in \mathcal{S}} f_t(\theta)$. We denote that particular version of dynamic regret by:

$$\mathcal{R}_d^* = \sum_{t=1}^{T} f_t(\theta_t) - \sum_{t=1}^{T} f_t(\theta_t^*)$$

[*]Work supported in part by NSF CMMI 1727096 and a University of MnDRIVE Informatics Graduate Fellowship.

In particular, if V^* is the corresponding path length:

$$V^* = \sum_{t=2}^{T} \left\| \theta_t^* - \theta_{t-1}^* \right\|, \tag{1}$$

then (Mokhtari et al. 2016) shows that for strongly convex functions, $\mathcal{R}_d^*$ of order $O(V^*)$ is obtained by gradient descent. However, as pointed out by (Zhang, Lu, and Zhou 2018), V^* metric is too pessimistic and unsuitable for stationary problems, which will result in poor generalization due to the random perturbation caused by sampling from the *same* distribution. Thus, a trade-off between static regret $\mathcal{R}_s$ and dynamic regret $\mathcal{R}_d^*$ is desired to maintain the abilities of both generalization to stationary problem and tracking to the local changes.

Adaptive regret (Hazan and Seshadhri 2009) is another metric when dealing with changing environment, which is defined as the maximum static regret over any contiguous time interval. Although it shares the similar goal as the dynamic regret, their relationship is still an open question.

Closely related to the problem of online learning is adaptive filtering, in which time series data is predicted using a filter that is designed from past data (Sayed 2011). The performance of adaptive filters is typically measured in an average case setting under statistical assumptions. One of the most famous adaptive filtering techniques is recursive least squares, which bears strong resemblance to the online Newton method of (Hazan, Agarwal, and Kale 2007). The work in (Hazan, Agarwal, and Kale 2007) proves a static regret bound of $O(\log T)$ for online Newton methods, but dynamic regret bounds are not known.

In order to have an algorithm that adapts to non-stationary data, it is common to use a forgetting factor. For the recursive least squares, (Guo, Ljung, and Priouret 1993) analyzed the effect of the forgetting factor in terms of the tracking error covariance matrix, and (Zhao et al. 2019) made the tracking error analysis with the assumptions that the noise is sub-Gaussian and the parameter follows a drifting model. However, none of the analysis mentioned is done in terms of the regret, which eliminates any noise assumption. For the online learning, (Garivier and Moulines 2011) analyzed the discounted UCB, which uses the discounted empirical average as the estimate for the upper confidence bound. (Russac, Vernade, and Cappé 2019) used the weighted least-squares to update the linear bandit's underlying parameter.

The contributions of this paper are:

1. For exp-concave and strongly convex problems, we propose a discounted Online Newton algorithm which generalizes recursive least squares with forgetting factors and the original online Newton method of (Hazan, Agarwal, and Kale 2007). We show how tuning the forgetting factor can achieve a dynamic regret bound of $\mathcal{R}_d \leq \max\{O(\log T), O(\sqrt{TV})\}$. This gives a rigorous analysis of forgetting factors in recursive least squares and improves the bounds described in (Zhang, Lu, and Zhou 2018). However, this choice requires a bound on the path length, V. For an alternative choice of forgetting factors, which does not require path length knowledge, we can simultaneously bound static regret by $\mathcal{R}_s \leq O(T^{1-\beta})$ and dynamic regret by $\mathcal{R}_d \leq \max\{O(T^{1-\beta}), O(T^{\beta}V)\}$. Note that tuning β produces a trade-off between static and dynamic regret.
2. Based on the analysis of discounted recursive least squares, we derive a novel step size rule for online gradient descent. Using this step size rule for smooth, strongly convex functions we obtain a static regret bound of $\mathcal{R}_s \leq O(T^{1-\beta})$ and a dynamic regret bound against $\theta_t = \operatorname{argmin}_{\theta \in \mathcal{S}} f_t(\theta)$ of $\mathcal{R}_d^* \leq O(T^{\beta}(1+V^*))$. This improves the trade-off obtained in the exp-concave case, since static regret or dynamic regret can be made small by appropriate choice of $\beta \in (0, 1)$.
3. We show how the step size rule can be modified further so that gradient descent recovers the $\max\{O(\log T), O(\sqrt{TV})\}$ dynamic regret bounds obtained by discounted Online Newton methods. However, as above, these bounds require knowledge of the bound on the path length, V.
4. Finally, we describe a meta-algorithm, similar to that used in (Zhang, Lu, and Zhou 2018), which can recover the $\max\{O(\log T), O(\sqrt{TV})\}$ dynamic regret bounds without knowledge of V. These bounds are tighter than those in (Zhang, Lu, and Zhou 2018), since they exploit exp-concavity to reduce the loss incurred by running an experts algorithm. Furthermore, we give a lower bound for the corresponding problems, which matches the obtained upper bound for certain range of V.

Notation. For the n dimensional vector $\theta \in \mathbb{R}^n$, we use $\|\theta\|$ to denote the ℓ_2-norm. The gradient of the function f_t at time step t in terms of the θ is denoted as $\nabla f_t(\theta)$.

For the matrix $A \in \mathbb{R}^{m \times n}$, its transpose is denoted by $A^\top$ and $A^\top A$ denotes the matrix multiplication. The inverse of A is denoted as A^{-1}. When $m = n$, we use $\|A\|_2$ to represent the induced 2 norm of the square matrix. For the two square matrix $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$, $A \preceq B$ means $A - B$ is negative semi-definite, while $A \succeq B$ means $A - B$ is positive semi-definite. For a positive definite matrix, M, let $\|x\|_M^2 = x^\top M x$. The standard inner product between matrices is given by $\langle A, B \rangle = \operatorname{Tr}(A^\top B)$. The determinant of a square matrix, A is denoted by $|A|$. We use I to represent the identity matrix.

Discounted Online Newton Algorithm

As described above, the online Newton algorithm from (Hazan, Agarwal, and Kale 2007) strongly resembles the classic recursive least squares algorithm from adaptive filtering (Sayed 2011). Currently, only the static regret of the online Newton method is studied. To obtain more adaptive performance, forgetting factors are often used in recursive least squares. However, the regret of forgetting factor algorithms has not been analyzed. This section proposes a class of algorithms that encompasses recursive least squares with forgetting factors and the online Newton algorithm. We show how dynamic regret bounds for these methods can be obtained by tuning the forgetting factor.

First we describe the problem assumptions. Throughout the paper we assume that $f_t : \mathcal{S} \to \mathbb{R}$ are convex, differentiable functions, $\mathcal{S}$ is a compact convex set, $\|x\| \leq D$ for all $x \in \mathcal{S}$, and $\|\nabla f_t(x)\| \leq G$ for all $x \in \mathcal{S}$. Without loss of generality, we assume throughout the paper that $D \geq 1$.

In this section we assume that all of the objective functions, $f_t : \mathcal{S} \to \mathbb{R}$ are α-exp-concave for some $\alpha > 0$. This means that $e^{-\alpha f_t(\theta)}$ is concave.

If f_t is twice differentiable, it can be shown that f_t is α-exp-concave if and only if

$$\nabla^2 f_t(x) \succeq \alpha \nabla f_t(x) \nabla f_t(x)^\top \tag{2}$$

for all $x \in \mathcal{S}$.

For an α-exp-concave function f_t, Lemma 4.2 of (Hazan 2016) implies that for all $\rho \leq \frac{1}{2}\min\{\frac{1}{4GD}, \alpha\}$, the following bound holds for all x and y in $\mathcal{S}$:

$$\begin{aligned} f_t(y) \geq f_t(x) + \nabla f_t(x)^\top (y-x) + \\ \frac{\rho}{2}(x-y)^\top \nabla f_t(x) \nabla f_t(x)^\top (x-y). \end{aligned} \tag{3a}$$

In some variations on the algorithm, we will require extra conditions on the function, f_t. In particular, in one variation we will require ℓ-strong convexity. which means that there is a number $\ell > 0$ such that

$$f_t(y) \geq f_t(x) + \nabla f_t(x)^\top (y-x) + \frac{\ell}{2}\|x-y\|^2 \tag{3b}$$

for all x and y in $\mathcal{S}$. For twice-differentiable functions, strong convexity implies α-exp-concavity for $\alpha \leq \ell/G^2$ on $\mathcal{S}$.

In another variant, we will require that the following bound holds for all x and y in $\mathcal{S}$:

$$f_t(y) \geq f_t(x) + \nabla f_t(x)^\top (y-x) + \frac{1}{2}\|x-y\|^2_{\nabla^2 f_t(x)}. \tag{3c}$$

This bound does not correspond to a commonly used convexity class, but it does hold for the important special case of quadratic functions: $f_t(x) = \frac{1}{2}\|y_t - A_t x\|^2$. This fact will be important for analyzing the classic discounted recursive least-squares algorithm. Note that if y_t and A_t are restricted to compact sets, α can be chosen so that f_t is α-exp-concave.

Additionally, the algorithms for strongly convex functions and those satisfying (3c) will require that the gradients $\nabla f_t(x)$ are u-Lipschitz for all $x \in \mathcal{S}$ (equivalently, $f_t(x)$ is u-smooth), which means the gradient $\nabla f_t(x)$ satisfies the relation

$$\|\nabla f_t(x) - \nabla f_t(y)\| \leq u\, \|x-y\|\,, \forall t.$$

This smoothness condition is equivalent to $f_t(y) \leq f_t(x) + \nabla f_t(x)^T(y-x) + \frac{u}{2}\|y-x\|^2$ and implies, in particular, that $\nabla^2 f_t(x) \preceq uI$.

To accommodate these three different cases, we propose Algorithm 1, in which $\Pi_{\mathcal{S}}^{P_t}(y) = \operatorname{argmin}_{z \in \mathcal{S}} \|z-y\|^2_{P_t}$ is the projection onto $\mathcal{S}$ with respect to the norm induced by P_t.

By using Algorithm 1, the following theorem can be obtained:

Algorithm 1 Discounted Online Newton Step

```
Given constants $\epsilon > 0$, $\eta > 0$, and $\gamma \in (0,1)$.
Let $\theta_1 \in \mathcal{S}$ and $P_0 = \epsilon I$.
for t=1,...,T do
    Play $\theta_t$ and incur loss $f_t(\theta_t)$
    Observe $\nabla_t = \nabla f_t(\theta_t)$ and $H_t = \nabla^2 f_t(\theta_t)$ (if needed)
    Update $P_t$:
        $P_t = \gamma P_{t-1} + \nabla_t \nabla_t^\top$    (Quasi-Newton)    (4a)
        $P_t = \gamma P_{t-1} + H_t$    (Full-Newton)    (4b)
    Update $\theta_t$: $\theta_{t+1} = \Pi_{\mathcal{S}}^{P_t}\left(\theta_t - \frac{1}{\eta}P_t^{-1}\nabla_t\right)$
end for
```

Theorem 1. *Consider the following three cases of Algorithm 1:*

1. *f_t is α-exp-concave. The algorithm uses $\eta \leq \frac{1}{2}\min\{\frac{1}{4GD}, \alpha\}$, $\epsilon = 1$ [1], and* (4a).
2. *f_t is α-exp-concave and ℓ-strongly convex while $\nabla f_t(x)$ is u-Lipschitz. The algorithm uses $\eta \leq \ell/u$, $\epsilon = 1$, and* (4b).
3. *f_t is α-exp-concave and satisfy (3c) while $\nabla f_t(x)$ is u-Lipschitz. The algorithm uses $\eta \leq 1$, $\epsilon = 1$, and* (4b).

For each of these cases, there are positive constants $a_1, \dots a_4$ such that

$$\begin{aligned} \textstyle\sum_{t=1}^T (f_t(\theta_t) - f_t(z_t)) \quad &\leq -a_1 T \log\gamma - a_2 \log(1-\gamma) \\ &\quad + \tfrac{a_3}{1-\gamma}V + a_4 \end{aligned}$$

for all $z_1, \dots, z_T \in \mathcal{S}$ such that $\sum_{t=2}^T \|z_t - z_{t-1}\| \leq V$.

Due to space limits, the proof is in the arxiv's version. Now we describe some consequences of the theorem.

Corollary 1. *Setting $\gamma = 1 - T^{-\beta}$ with $\beta \in (0,1)$ leads to the following form:*

$$\begin{aligned} &\textstyle\sum_{t=1}^T (f_t(\theta_t) - f_t(z_t)) \\ &\leq O(T^{1-\beta} + \beta \log T + T^\beta V) \end{aligned}$$

Proof. The first term is bounded as:

$$\begin{aligned} -T\log\gamma &= -T\log(1 - T^{-\beta}) \\ &\leq \frac{T^{1-\beta}}{1 - T^{-\beta}} = O(T^{1-\beta}), \end{aligned}$$

where the inequality follows from $-\log(1-x) \leq \frac{x}{1-x}$ for $0 \leq x < 1$.

The other terms follow by direct calculation. □

This corollary guarantees that the static regret is bounded in the order of $O(T^{1-\beta})$ since $V = 0$ in that case. The dynamic regret is of order $O(T^{1-\beta} + T^\beta V)$. By choosing $\beta \in (0,1)$, we are guaranteed that both the static and dynamic regrets are both sublinear in T as long as $V < O(T)$. Also, small static regret can be obtained by setting β near 1.

[1] The value used here is only for proof simplicity, please see Meta-algorithm Section for more discussion.

In the setting of Corollary 1, the algorithm parameters do not depend on the path length V. Thus, the bounds hold for any path length, whether or not it is known a priori. The next corollary shows how tighter bounds could be obtained if knowledge of V were exploited in choosing the discount factor, γ.

Corollary 2. *Setting* $\gamma = 1 - \frac{1}{2}\sqrt{\frac{\max\{V, \log^2 T/T\}}{2DT}}$ *leads to the form:*

$$\sum_{t=1}^{T}(f_t(\theta_t) - f_t(z_t)) \leq \max\{O(\log T), O(\sqrt{TV})\}$$

The proof is similar to the proof of Corollary 1.

Note that Corollary 2 implies that the discounted Newton method achieves logarithmic static regret by setting $V = 0$. This matches the bounds obtained in (Hazan, Agarwal, and Kale 2007). For positive path lengths bounded by V, we improve the $O(\sqrt{T(1+V)})$ dynamic bounds from (Zhang, Lu, and Zhou 2018). However, the algorithm above current requires knowing a bound on the path length, whereas (Zhang, Lu, and Zhou 2018) achieves its bound without knowing the path length, a priori.

If we view V as the variation budget that $z_1^T = z_1, \dots, z_T$ can vary over $\mathcal{S}$ like in (Besbes, Gur, and Zeevi 2015), and use this as a pre-fixed value to allow the comparator sequence to vary arbitrarily over the set of admissible comparator sequence $\{z_1^T \in \mathcal{S} : \sum_{t=2}^{T} \|z_t - z_{t-1}\| \leq V\}$, we can tune γ in terms of V.

In order to bound the dynamic regret without knowing a bound on the path length, the method of (Zhang, Lu, and Zhou 2018) runs a collection of gradient descent algorithms in parallel with different step sizes and then uses a meta-optimization (Cesa-Bianchi and Lugosi 2006) to weight their solutions. In a later section, we will show how a related meta-optimization over the discount factor leads to $\max\{O(\log T), O(\sqrt{TV})\}$ dynamic regret bounds for unknown V.

For the Algorithm 1, we need to invert P_t, which can be achieved in time $O(n^2)$ for the Quasi-Newton case in (4a) by utilizing the matrix inversion lemma. However, for the Full-Newton step (4b), the inversion requires $O(n^3)$ time.

From Forgetting Factors to a Step Size Rule

In the next few sections, we aim to derive gradient descent rules that achieve similar static and regret bounds to the discounted Newton algorithm, without the cost of inverting matrices. We begin by analyzing the special case of quadratic functions of the form:

$$f_t(\theta) = \frac{1}{2}\|\theta - y_t\|^2, \tag{5}$$

where $y_t \in \mathcal{S}$. In this case, we will see that discounted recursive least squares can be interpreted as online gradient descent with a special step size rule. We will show how this step size rule achieves a trade-off between static regret and dynamic regret with the specific comparison sequence $\theta_t^* = y_t = \operatorname{argmin}_{\theta \in \mathcal{S}} f_t(\theta)$. For a related analysis of more general quadratic functions, $f_t(\theta) = \frac{1}{2}\|A_t\theta - y_t\|^2$, please see the arxiv's version.

Note that the previous section focused on dynamic regret for arbitrary comparison sequences, $z_1^T \in \mathcal{S}$. The analysis techniques in this and the next section are specialized to comparisons against $\theta_t^* = \operatorname{argmin}_{\theta \in \mathcal{S}} f_t(\theta)$, as studied in works such as (Mokhtari et al. 2016; Yang et al. 2016).

Classic discounted recursive least squares corresponds to Algorithm 1 running with full Newton steps, $\eta = 1$, and initial matrix $P_0 = 0$. When f_t is defined as in (5), we have that $P_t = \sum_{k=0}^{t-1}\gamma^k I$. Thus, the update rule can be expressed in the following equivalent ways:

$$\theta_{t+1} = \operatorname*{argmin}_{\theta \in \mathcal{S}} \sum_{i=1}^{t} \gamma^{i-1} f_{t+1-i}(\theta) \tag{6a}$$

$$= \frac{\gamma - \gamma^t}{1 - \gamma^t}\theta_t + \frac{1-\gamma}{1-\gamma^t}y_t \tag{6b}$$

$$= \theta_t - P_t^{-1}\nabla f_t(\theta_t) \tag{6c}$$

$$= \theta_t - \eta_t \nabla f_t(\theta_t), \tag{6d}$$

where $\eta_t = \frac{1-\gamma}{1-\gamma^t}$. Note that since $y_t \in \mathcal{S}$, no projection steps are needed.

The above update is the ubiquitous gradient descent with a changing step size. The only difference from the standard methods is the choice of η_t, which will lead to the useful trade-off between dynamic regret $\mathcal{R}_d^*$ and static regret to maintain the abilities of both generalization to stationary problem and tracking to the local changes.

By using the above update, we can get the relationship between $\theta_{t+1} - \theta_t^*$ and $\theta_t - \theta_t^*$ as the following result:

Lemma 1. *Let* $\theta_t^* = \operatorname{argmin}_{\theta \mathcal{S}} f_t(\theta)$ *in Eq.*(5)*. When using the discounted recursive least-squares update in Eq.*(6)*, we have the following relation:*

$$\theta_{t+1} - \theta_t^* = \frac{\gamma - \gamma^t}{1 - \gamma^t}(\theta_t - \theta_t^*)$$

Proof. Since $\theta_t^* = \operatorname{argmin} f_t(\theta) = y_t$, for $\theta_{t+1} - \theta_t^*$, we have:

$$\begin{aligned}\theta_{t+1} - \theta_t^* &= \theta_{t+1} - y_t \\ &= \tfrac{\gamma-\gamma^t}{1-\gamma^t}\theta_t + \tfrac{1-\gamma}{1-\gamma^t}y_t - y_t \\ &= \tfrac{\gamma-\gamma^t}{1-\gamma^t}(\theta_t - y_t) = \tfrac{\gamma-\gamma^t}{1-\gamma^t}(\theta_t - \theta_t^*)\end{aligned}$$

□

Recall from (1) that the path length of optimizer sequence is denoted by V^*. With the help of Lemma 1, we can upper bound the dynamic regret $\mathcal{R}_d^*$ in the next theorem:

Theorem 2. *Let* θ_t^* *be the solution to* $f_t(\theta)$ *in Eq.*(5)*. When using the discounted recursive least-squares update in Eq.*(6) *with* $1 - \gamma = 1/T^\beta, \beta \in (0, 1)$*, we can upper bound the dynamic regret as:*

$$\mathcal{R}_d^* \leq 2DT^\beta\big(\|\theta_1 - \theta_1^*\| + V^*\big)$$

Proof. According to the Mean Value Theorem, there exists a vector $x \in \{v|v = \delta\theta_t + (1-\delta)\theta_t^*, \delta \in [0,1]\}$ such that $f_t(\theta_t) - f_t(\theta_t^*) = \nabla f_t(x)^T(\theta_t - \theta_t^*) \leq \|\nabla f_t(x)\| \, \|\theta_t - \theta_t^*\|$. For our problem, $\|\nabla f_t(x)\| = \|x - y_t\| \leq \|x\| + \|y_t\|$. For $\|x\|$, we have:

$$\begin{aligned}\|x\| &= \|\delta\theta_t + (1-\delta)\theta_t^*\| \\ &\leq \delta \,\|\theta_t\| + (1-\delta)\,\|y_t\| \\ &= \delta \left\| \frac{\sum_{i=1}^{t-1}\gamma^{i-1}y_{t-i}}{\sum_{i=1}^{t-1}\gamma^{i-1}} \right\| + (1-\delta)\,\|y_t\| \\ &\leq D\end{aligned}$$

where the second inequality is due to $\|y_i\| \leq D, \forall i$.

As a result, the norm of the gradient can be upper bounded as $\|\nabla f_t(x)\| \leq 2D$. Then we have $\mathcal{R}_d^* = \sum_{t=1}^{T}\Big(f_t(\theta_t) - f_t(\theta_t^*)\Big) \leq 2D\sum_{t=1}^{T}\|\theta_t - \theta_t^*\|$. Now we could instead upper bound $\sum_{t=1}^{T}\|\theta_t - \theta_t^*\|$, which can be achieved as follows:

$$\begin{aligned}&\sum_{t=1}^{T}\|\theta_t - \theta_t^*\| \\ &= \|\theta_1 - \theta_1^*\| + \sum_{t=2}^{T}\left\|\theta_t - \theta_{t-1}^* + \theta_{t-1}^* - \theta_t^*\right\| \\ &\leq \|\theta_1 - \theta_1^*\| + \sum_{t=1}^{T-1}\|\theta_{t+1} - \theta_t^*\| + \sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\| \\ &= \|\theta_1 - \theta_1^*\| + \sum_{t=1}^{T-1}\tfrac{\gamma - \gamma^t}{1-\gamma^t}\|\theta_t - \theta_t^*\| + \sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\| \\ &\leq \|\theta_1 - \theta_1^*\| + \sum_{t=1}^{T}\tfrac{\gamma - \gamma^t}{1-\gamma^t}\|\theta_t - \theta_t^*\| + \sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\|\end{aligned}$$

where in the second equality, we substitute the result from Lemma 1.

From the above inequality, we get

$$\sum_{t=1}^{T}\Big(1 - \frac{\gamma - \gamma^t}{1-\gamma^t}\Big)\|\theta_t - \theta_t^*\| \leq \|\theta_1 - \theta_1^*\| + \sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\|$$

Since $\Big(1 - \frac{\gamma - \gamma^t}{1-\gamma^t}\Big) = \frac{1-\gamma}{1-\gamma^t} \geq 1 - \gamma$, we get

$$\begin{aligned}\sum_{t=1}^{T}\|\theta_t - \theta_t^*\| &\leq \tfrac{1}{1-\gamma}\left\|\theta_1 - \theta_1^*\right\| + \tfrac{1}{1-\gamma}\sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\| \\ &= T^{\beta}(\left\|\theta_1 - \theta_1^*\right\| + \sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\|)\end{aligned}$$

Thus, $\mathcal{R}_d \leq 2D\sum_{t=1}^{T}\|\theta_t - \theta_t^*\| \leq 2DT^{\beta}(\|\theta_1 - \theta_1^*\| + \sum_{t=2}^{T}\left\|\theta_t^* - \theta_{t-1}^*\right\|)$. □

Theorem 2 shows that if we choose the discounted factor $\gamma = 1 - T^{-\beta}$ we obtain a dynamic regret of $O(T^{\beta}(1+V^*))$. This is a refinement of the Corollary 1 since the bound no longer has the $T^{1-\beta}$ term. Thus, the dynamic regret can be made small by choosing a small β.

In the next theorem, we will show that this carefully chosen γ can also lead to useful static regret, which can give us a trade-off and solve the dilemma of generalization for stationary problems versus the tracking for local changes.

Theorem 3. *Let θ^* be the solution to* $\min \sum_{t=1}^{T} f_t(\theta)$. *When using the discounted recursive least-squares update in Eq.*(6) *with* $1 - \gamma = 1/T^{\beta}, \beta \in (0,1)$, *we can upper bound the static regret as:*

$$\mathcal{R}_s \leq O(T^{1-\beta})$$

Recall that the algorithm of this section can be interpreted both as a discounted recursive least squares method, and as a gradient descent method. As a result, this theorem is actually a direct consequence of Corollary 1, by setting $V = 0$. However, we will give a separate proof in the arxiv, since the techniques extend naturally to the analysis of more general work on gradient descent methods of the next section.

Our Theorems 2 and 3 build a trade-off between dynamic and static regret by the carefully chosen discounted factor γ. Compared with the result from the last section, there are two improvements: 1. The two regrets are decoupled so that we could reduce the β to make the dynamic regret $\mathcal{R}_d^*$ result smaller than bound from Corollary 1; 2. The update is the first-order gradient descent, which is computationally more efficient than second order methods.

In the next section, we will consider the strongly convex and smooth case, whose result is inspired by this section's analysis.

Online Gradient Descent for Smooth, Strongly Convex Problems

In this section, we generalize the results of the previous section idea to functions which are ℓ-strongly convex and u-smooth. We will see that similar bounds on $\mathcal{R}_s$ and $\mathcal{R}_d^*$ can be obtained.

Our proposed update rule for the prediction θ_{t+1} at time step $t+1$ is:

$$\theta_{t+1} = \operatorname*{argmin}_{\theta \in \mathcal{S}} \left\|\theta - (\theta_t - \eta_t \nabla f_t(\theta_t))\right\|^2 \qquad (7)$$

where $\eta_t = \frac{1-\gamma}{\ell(\gamma - \gamma^t) + u(1-\gamma)}$ and $\gamma \in (0,1)$.

This update rule generalizes the step size rule from the last section.

Before getting to the dynamic regret, we will first derive the relation between $\|\theta_{t+1} - \theta_t^*\|$ and $\|\theta_t - \theta_t^*\|$ to try to mimic the result in Lemma 1 of the quadratic case:

Lemma 2. *Let $\theta_t^* \in \mathcal{S}$ be the solution to $f_t(\theta)$ which is strongly convex and smooth. When we use the update in Eq.*(7), *the following relation is obtained:*

$$\|\theta_{t+1} - \theta_t^*\| \leq \sqrt{1 - \frac{l(1-\gamma)}{u(1-\gamma) + l\gamma}}\,\|\theta_t - \theta_t^*\|$$

Due to space limits, the proof is in the arxiv's version.

Now we are ready to present the dynamic regret result:

Theorem 4. *Let θ_t^* be the solution to $f_t(\theta), \theta \in \mathcal{S}$. When using the update in Eq.*(7) *with $1-\gamma = 1/T^\beta, \beta \in (0,1)$, we can upper bound the dynamic regret:*

$$\mathcal{R}_d^* \leq G\big(2(T^\beta - 1) + u/l\big)(\|\theta_1 - \theta_1^*\| + V^*)$$

The proof follows the similar steps in the proof of Theorem 2. Due to space limits, please refer to the version in arxiv.

Theorem 4's result seems promising in achieving the trade-off, since it has a similar form of the result from quadratic problems in Theorem 2. Next, we will present the static regret result, which assures that the desired trade-off can be obtained.

Theorem 5. *Let θ^* be the solution to $\min\limits_{\theta \in \mathcal{S}} \sum\limits_{t=1}^{T} f_t(\theta)$. When using the update in Eq.*(7) *with $1-\gamma = 1/T^\beta, \beta \in (0,1)$, we can upper bound the static regret:*

$$\mathcal{R}_s \leq O(T^{1-\beta})$$

The proof follows the similar steps in the proof of Theorem 3. Due to space limits, please refer to the version in arxiv.

The regret bounds of this section are similar to those obtained for simple quadratics. Thus, this gradient descent rule maintains all of the advantages over the discounted Newton method that were described in the previous section and the advantages of trading off static regret and dynamic regret $\mathcal{R}_d^*$.

Online Gradient Descent for Strongly Convex Problems

In this section, we extend step size idea from previous section to problems which are ℓ-strongly convex, but not necessarily smooth. We obtain a dynamic regret of $\mathcal{R}_d \leq \max\{O(\log T), O(\sqrt{TV})\}$, similar to the discounted online Newton method. However, our analysis does not lead to the clean trade-off of $\mathcal{R}_s \leq O(T^{1-\beta})$ and $\mathcal{R}_d^* \leq O(T^\beta(1+V^*))$ obtained when smoothness is also used.

The update rule is online gradient descent:

$$\theta_{t+1} = \operatorname*{argmin}_{\theta \in \mathcal{S}} \|\theta - (\theta_t - \eta_t \nabla f_t(\theta_t))\|^2 \tag{8}$$

where $\eta_t = \frac{1-\gamma}{\ell(1-\gamma^t)}$, and $\gamma \in (0,1)$.

We can see that the update rule is the same as the one in Eq.(7) while the step size η_t is replaced with $\frac{1-\gamma}{\ell(1-\gamma^t)}$.

By using the new step size with the update rule in Eq.(8), we can obtain the following dynamic regret bound:

Theorem 6. *If using the update rule in Eq.*(8) *with $\eta_t = \frac{1-\gamma}{\ell(1-\gamma^t)}$ and $\gamma \in (0,1)$, the following dynamic regret can be obtained:*

$$\sum_{t=1}^{T}\Big(f_t(\theta_t) - f_t(z_t)\Big) \leq 2D\ell\frac{1}{1-\gamma}V + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t$$

Proof. According to the non-expansive property of the projection operator and the update rule in Eq.(8), we have

$$\begin{aligned}\|\theta_{t+1} - z_t\|^2 &\leq \|\theta_t - \eta_t\nabla f_t(\theta_t) - z_t\|^2 \\ &= \|\theta_t - z_t\|^2 - 2\eta_t \nabla f_t(\theta_t)^T(\theta_t - z_t) \\ &\quad + \eta_t^2 \|\nabla f_t(\theta_t)\|^2\end{aligned}$$

The reformulation gives us

$$\begin{aligned}\nabla f_t(\theta_t)^T(\theta_t - z_t) &\leq \tfrac{1}{2\eta_t}\big(\|\theta_t - z_t\|^2 - \|\theta_{t+1} - z_t\|^2\big) \\ &\quad + \tfrac{\eta_t}{2}\|\nabla f_t(\theta_t)\|^2\end{aligned} \tag{9}$$

Moreover, from the strong convexity, we have $f_t(z_t) \geq f_t(\theta_t) + \nabla f_t(\theta_t)^T(z_t - \theta_t) + \frac{\ell}{2}\|z_t - \theta_t\|^2$, which is equivalent to $\nabla f_t(\theta_t)^T(\theta_t - z_t) \geq f_t(\theta_t) - f_t(z_t) + \frac{\ell}{2}\|z_t - \theta_t\|^2$. Combined with Eq.(9), we have

$$\begin{aligned}f_t(\theta_t) - f_t(z_t) &\leq \tfrac{1}{2\eta_t}\big(\|\theta_t - z_t\|^2 - \|\theta_{t+1} - z_t\|^2\big) \\ &\quad + \tfrac{\eta_t}{2}\|\nabla f_t(\theta_t)\|^2 - \tfrac{\ell}{2}\|z_t - \theta_t\|^2\end{aligned} \tag{10}$$

Then we can lower bound $\|\theta_{t+1} - z_t\|^2$ by

$$\begin{aligned}\|\theta_{t+1} - z_t\|^2 &= \|\theta_{t+1} - z_{t+1}\|^2 + \|z_{t+1} - z_t\|^2 \\ &\quad + 2(\theta_{t+1} - z_{t+1})^\top(z_{t+1} - z_t) \\ &\geq \|\theta_{t+1} - z_{t+1}\|^2 - 4D\|z_{t+1} - z_t\|\end{aligned} \tag{11}$$

Combining (10) and (11) gives

$$\begin{aligned}&f_t(\theta_t) - f_t(z_t) \\ &\leq \tfrac{1}{2\eta_t}\big(\|\theta_t - z_t\|^2 - \|\theta_{t+1} - z_{t+1}\|^2\big) + \tfrac{2D}{\eta_t}\|z_{t+1} - z_t\| \\ &\quad + \tfrac{\eta_t}{2}\|\nabla f_t(\theta_t)\|^2 - \tfrac{\ell}{2}\|z_t - \theta_t\|^2\end{aligned}$$

Summing over t from 1 to T, dropping the term $-\frac{1}{2\eta_T}\|\theta_{T+1} - z_{T+1}\|^2$, setting $z_{T+1} = z_T$, using the inequality $\|\nabla f_t(\theta_t)\|^2 \leq G^2$, and re-arranging gives

$$\begin{aligned}&\sum_{t=1}^{T}\Big(f_t(\theta_t) - f_t(z_t)\Big) \\ &\leq \tfrac{1}{2}(\tfrac{1}{\eta_1} - \ell)\|\theta_1 - z_1\|^2 + \tfrac{1}{2}\sum_{t=1}^{T}(\tfrac{1}{\eta_t} - \tfrac{1}{\eta_{t-1}} - \ell)\|\theta_t - z_t\|^2 \\ &\quad + 2D\sum_{t=1}^{T-1}\tfrac{1}{\eta_t}\|z_{t+1} - z_t\| + \tfrac{G^2}{2}\sum_{t=1}^{T}\eta_t \\ &\leq 2D\ell\tfrac{1}{1-\gamma}V + \tfrac{G^2}{2}\sum_{t=1}^{T}\eta_t\end{aligned}$$

where for the second inequality, we use the following results: $\frac{1}{\eta_1} - \ell = 0$, $\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \ell = \frac{\ell(1-\gamma)(\gamma^{t-1}-1)}{1-\gamma} \leq 0$, $\frac{1}{\eta_t} = \frac{\ell(1-\gamma^t)}{1-\gamma} \leq \frac{\ell}{1-\gamma}$, and the definition of V.

□

Similar to the case of discounted online Newton methods, if a bound on the path length, V, is known, the discount factor can be tuned to achieve low dynamic regret:

Corollary 3. *By setting $\gamma = 1 - \frac{1}{2}\sqrt{\frac{\max\{V, \log^2 T/T\}}{2DT}}$, the following bound can be obtained:*

$$\sum_{t=1}^{T}\Big(f_t(\theta_t) - f_t(z_t)\Big) \leq \max\{O(\log T), O(\sqrt{TV})\}.$$

This result is tighter than the $O(\sqrt{T(1+V)})$ bound obtained by (Zhang, Lu, and Zhou 2018) on convex functions, but not directly comparable to the $O(V^*)$ bounds obtained in (Mokhtari et al. 2016) for smooth, strongly convex functions.

Similar to the Corollary 2 on discounted online Newton methods, Corollary 3 requires knowing V. In the next section, we will see how a meta-algorithm can be used to obtain the same bounds without knowing V.

Meta-algorithm

In previous sections, we discussed the results on dynamic regret for both α-exp-concave and ℓ-strongly convex objectives. The tightest regret bounds were obtained by choosing a discount factor that depends on V, a bound on the path length. In this section, we solve this issue by running multiple algorithms in parallel with different discount factors.

For online convex optimization, a similar meta-algorithm has been used by (Zhang, Lu, and Zhou 2018) to search over step sizes. However, the method of (Zhang, Lu, and Zhou 2018) cannot be used directly in either the α-exp-concave or ℓ-strongly convex case due to the added $O(\sqrt{T})$ regret from running multiple algorithms. In order to remove this factor, we exploit the exp-concavity in the experts algorithm, as in Chapter 3 in (Cesa-Bianchi and Lugosi 2006).

In this section, we will show that by using appropriate parameters and analysis designed specifically for our cases, the meta-algorithm can be used to solve our issues.

Algorithm 2 Meta-Algorithm

```
Given step size $\lambda$, and a set $\mathcal{H}$ containing discount factors for each algorithm.
Activate a set of algorithms $\{A^\gamma | \gamma \in \mathcal{H}\}$ by calling Algorithm 1 (exp-concave case) or the update in Eq.(8) (strongly convex case) for each parameter $\gamma \in \mathcal{H}$.
Sort $\gamma$ in descending order $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_N$, and set $w_1^{\gamma_i} = \frac{C}{i(i+1)}$ with $C = 1 + 1/|\mathcal{H}|$.
for t=1,...,T do
    Obtain $\theta_t^\gamma$ from each algorithm $A^\gamma$.
    Play $\theta_t = \sum_{\gamma \in \mathcal{H}} w_t^\gamma \theta_t^\gamma$, and incur loss $f_t(\theta_t^\gamma)$ for each $\theta_t^\gamma$.
    Update $w_t^\gamma$ by
        $$w_{t+1}^\gamma = \frac{w_t^\gamma \exp(-\lambda f_t(\theta_t^\gamma))}{\sum_{\mu \in \mathcal{H}} w_t^\mu \exp(-\lambda f_t(\theta_t^\mu))}.$$
    Send back the gradient $\nabla f_t(\theta_t^\gamma)$ for each algorithm $A^\gamma$.
end for
```

Exp-concave case

Before showing the regret result, we first show that the cumulative loss of the meta-algorithm is comparable to all $A^\gamma \in \mathcal{H}$:

Lemma 3. *If f_t is α-exp-concave and $\lambda = \alpha$, the cumulative loss difference of Algorithm 2 for any $\gamma \in \mathcal{H}$ is bounded as:*

$$\sum_{t=1}^{T} (f_t(\theta_t) - f_t(\theta_t^\gamma)) \leq \frac{1}{\alpha} \log \frac{1}{w_1^\gamma}$$

This result shows how $O(\sqrt{T})$ regret incurred by running an experts algorithm is reduced in the α-exp-concave case. The result is similar to Proposition 3.1 of (Cesa-Bianchi and Lugosi 2006).

Based on the above lemma, if we can show that there exists an algorithm A^γ, which can bound the regret $\sum_{t=1}^{T}(f_t(\theta_t^\gamma) - f_t(z_t)) \leq \max\{O(\log T), O(\sqrt{TV})\}$, then we can combine these two results and show that the regret holds for $\theta_t, t = 1, \ldots, T$ as well:

Theorem 7. *For any comparator sequence $z_1, \ldots, z_T \in \mathcal{S}$, setting $\mathcal{H} = \left\{\gamma_i = 1 - \eta_i \middle| i = 1, \ldots, N\right\}$ with $T \geq 2$ where $\eta_i = \frac{1}{2}\frac{\log T}{T\sqrt{2D}}2^{i-1}$, $N = \lceil \frac{1}{2}\log_2(\frac{2DT^2}{\log^2 T})\rceil + 1$, and $\lambda = \alpha$ leads to the result:*

$$\sum_{t=1}^{T} (f_t(\theta_t) - f_t(z_t)) \leq O(\max\{\log T, \sqrt{TV}\})$$

As described previously, the proof's main idea is to show that we could both find an algorithm A^γ bounding the regret $\sum_{t=1}^{T}(f_t(\theta_t^\gamma) - f_t(z_t)) \leq \max\{O(\log T), O(\sqrt{TV})\}$ and cover the V with $O(\log T)$ different γ choices. Please see the arxiv's version for the formal proof.

In practice, we include the additional case when $\gamma = 1$ to make the overall algorithm explicitly balance the static regret. Also, the free parameter ϵ used in Algorithm 1 is important for the actual performance. If it is too small, the update will be easily effected by the gradient to have high generalization error. In practice, it can be set to be equal to $1/(\rho^2 D^2)$ or $1/(\rho^2 D^2 N)$ with $\rho = \frac{1}{2}\min\{\frac{1}{4GD}, \alpha\}$ like in (Hazan 2016).

Strongly convex case

For the strongly convex problem, since the parameter γ used in Corollary 3 is the same as the one in Corollary 2, it seems likely that the meta-algorithm should work with the same setup in as Theorem 7. The only parameter that needs to be changed is λ, which was set above to α, the parameter of α-exp-concavity.

To proceed, we first show that the ℓ-strongly convex function with bounded gradient (e.g.,$\|\nabla f_t\| \leq G$) is also ℓ/G^2-exp-concave. Previous works also pointed out this, but their statement only works when f_t is second-order differentiable, while our result is true when f_t is first-order differentiable.

Lemma 4. *For the ℓ-strongly convex function f_t with $\|\nabla f_t\| \leq G$, it is also α-exp-concave with $\alpha = \ell/G^2$.*

Lemma 4 indicates that running Algorithm 2 with strongly convex function leads to the same result as in Lemma 3. Thus, using the similar idea as discussed in the case of α-exp-concavity and Algorithm 2, the theorem below can be obtained:

Theorem 8. *For any comparator sequence $z_1, \ldots, z_T \in \mathcal{S}$, setting $\mathcal{H} = \left\{\gamma_i = 1 - \eta_i \middle| i = 1, \ldots, N\right\}$ with $T \geq 2$*

where $\eta_i = \frac{1}{2}\frac{\log T}{T\sqrt{2D}}2^{i-1}$, $N = \lceil \frac{1}{2}\log_2(\frac{2DT^2}{\log^2 T})\rceil + 1$, *and* $\lambda = \ell/G^2$ *leads to the result:*

$$\sum_{t=1}^{T}(f_t(\theta_t) - f_t(z_t)) \leq O(\max\{\log T, \sqrt{TV}\})$$

As discussed in the previous subsection, in practice, we also include the case when $\gamma = 1$ to make the overall algorithm explicitly balance the static regret and set ϵ accordingly as in the exp-concave case.

A Lower bound

In the previous subsections, we demonstrate how to achieve the dynamic regret $\max\{O(\log T), O(\sqrt{TV})\}$ for both the exp-concave and strongly convex problems without knowing V. In this subsection, we will give a lower bound, which approaches the upper bound for large and small V.

Proposition 1. *For losses of the form* $f_t(\theta) = (\theta - \epsilon_t)^2$, *for all* $\gamma_0 \in (0, 1)$ *and all* $V = T^{\frac{2+\gamma_0}{4-\gamma_0}}$, *there is a comparison sequence* z_1^T *such that* $\sum_{t=2}^{T} \|z_t - z_{t-1}\| \leq V$ *and*

$$\mathcal{R}_d \geq \max\{O(\log T), O\big((VT)^{\frac{\gamma_0}{2}}\big)\}.$$

The above result has the following indications: 1. For $V = o(T)$ but approaching to T, it is impossible to achieve better bound of $\mathcal{R}_d \geq O\Big((VT)^{\frac{\alpha_0}{2}}\Big)$ with $\alpha_0 < 1$. 2. For other ranges of V like $V = O(\sqrt{T})$, its lower bound is not established and still an open question.

Conclusion

In this paper, we propose a discounted online Newton algorithm that generalizes recursive least squares with forgetting factors and existing online Newton methods. We prove a dynamic regret bound $\max\{O(\log T), O(\sqrt{TV})\}$ which provides a rigorous analysis of forgetting factor algorithms. In the special case of simple quadratic functions, we demonstrate that the discounted Newton method reduces to a gradient descent algorithm with a particular step size rule. We show how this step size rule can be generalized to apply to strongly convex functions, giving a substantially more computationally efficient algorithm than the discounted online Newton method, while recovering the dynamic regret guarantees. The strongest regret guarantees depend on knowledge of the path length, V. We show how to use a meta-algorithm that optimizes over discount factors to obtain the same regret guarantees without knowledge of V as well as a lower bound which matches the obtained upper bound for certain range of V. Finally, when the functions are smooth we show how this new gradient descent method enables a static regret of $\mathcal{R}_s \leq O(T^{1-\beta})$ and $\mathcal{R}_d^* \leq O(T^{\beta}(1 + V^*))$, where $\beta \in (0, 1)$ is a user-specified trade-off parameter.

References

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

Blum, A.; Kumar, V.; Rudra, A.; and Wu, F. 2004. Online learning in online auctions. *Theoretical Computer Science* 324(2-3):137–146.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; and Singer, Y. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7(Mar):551–585.

Garivier, A., and Moulines, E. 2011. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, 174–188. Springer.

Guo, L.; Ljung, L.; and Priouret, P. 1993. Performance analysis of the forgetting factor rls algorithm. *International journal of adaptive control and signal processing* 7(6):525–537.

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

Hazan, E., and Seshadhri, C. 2009. Efficient learning algorithms for changing environments. In *Proceedings of the 26th annual international conference on machine learning*, 393–400. ACM.

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

Mokhtari, A.; Shahrampour, S.; Jadbabaie, A.; and Ribeiro, A. 2016. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 7195–7201. IEEE.

Russac, Y.; Vernade, C.; and Cappé, O. 2019. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems*, 12017–12026.

Sayed, A. H. 2011. *Adaptive filters*. John Wiley & Sons.

Yang, T.; Zhang, L.; Jin, R.; and Yi, J. 2016. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, 449–457.

Yuan, J., and Lamperski, A. 2018. Online convex optimization for cumulative constraints. In *Advances in Neural Information Processing Systems*, 6137–6146.

Yuan, J., and Lamperski, A. 2019. Online adaptive principal component analysis and its extensions. In *International Conference on Machine Learning*, 7213–7221.

Zhang, L.; Lu, S.; and Zhou, Z.-H. 2018. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems*, 1323–1333.

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

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 928–936.