Dynamic and Adaptive Implicit Online Learning

Anonymous Author
Anonymous Institution

Abstract

This work focuses on the setting of dynamic regret in the context of online learning. Building on recent advances in the analysis of Implicit updates in online learning, we propose an adaptation of the Implicit version of Online Mirror Descent to the dynamic setting, when the full loss is observed at the end of each round. Our proposed algorithm is adaptive to different measures of variability of the environment at the same time, namely the temporal variability of the loss functions and the path length of the sequence of comparators. In particular, we improve the known regret bounds with respect to the temporal variability of the losses. Furthermore, our analysis reveals that our results are tight and cannot be improved without further assumptions. Next, we show applications of our algorithm, in particular to the case of composite loss functions which could be relevant in practical scenarios. Finally, we show how to combine existing algorithms to obtain a new algorithm adaptive to different sequences of comparators simultaneously.

1 Introduction

Online learning is a powerful tool in modeling many practical scenarios. Furthermore, in recent years it has led to advancements in various areas of machine learning in general, both practically and theoretically. The usual goal in online learning is to minimize the *static regret* against a fixed comparator. Formally, given a convex set $\mathcal{V} \subseteq \mathbb{R}^d$, a time horizon T and a sequence of cost functions $\ell_1, \dots, \ell_t$, the goal is to design algorithms such that for any comparator model $\boldsymbol{u} \in \mathcal{V}$ the following quantity is minimized

$$R_T(\boldsymbol{u}) \triangleq \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}),$$

where $\boldsymbol{x}_t$ is the output of the algorithm at time t.

An algorithm is called *no-regret* when the quantity $R_T(\boldsymbol{u})/T \to 0$ as T goes to infinity. Therefore, the goal in online learning is to design algorithms whose regret can be provably upper bounded by a quantity which grows sublinearly in T.

While the static regret is often a well studied objective and many algorithms have a sublinear regret upper bound, sometimes competing with the best comparator is not meaningful. Indeed, there are situations where the environment is not stationary. In this case, rather than comparing the performance of an algorithm against a single benchmark, it is preferable to compete against a "moving" target, i.e., a sequence of different comparators. To model these scenarios, stronger notions of regret and measure of the dynamic environment are used.

Hence, we define the *general dynamic regret* [Zinkevich, 2003; Hall and Willett, 2013] against the sequence $\boldsymbol{u}_{1:T} \triangleq (\boldsymbol{u}_1, \dots, \boldsymbol{u}_T)$ as

$$R_T(\boldsymbol{u}_{1:T}) \triangleq \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}_t) \ . \tag{1}$$

It can be shown that it is impossible to achieve sublinear dynamic regret in the worst-case. However, if one puts some restrictions on the sequence $\boldsymbol{u}_{1:T}$ and makes some regularities assumptions, then Eq. (1) can be sublinear in T. There are various measures which can be used to model the regularity of the environment. A natural measure of non-stationarity introduced in Zinkevich [2003] is the *path-length*[1] of the sequence $\boldsymbol{u}_{1:T}$,

$$C_T(\boldsymbol{u}_{1:T}) \triangleq \sum_{t=2}^{T} \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\| \ . \tag{2}$$

[1] One could also consider other versions of path-length, such as its squared version [Yang et al., 2016].

Preliminary work. Under review by AISTATS 2021. Do not distribute.

Another measure of non-stationarity is given by the *temporal variability* of the loss functions [Besbes et al., 2015]. Formally, let $\ell_{1:T}$ be the shorthand for $(\ell_1, \dots, \ell_T)$. The temporal variability of a sequence $\ell_{1:T}$ is defined as follows

$$V_T(\ell_{1:T}) \triangleq \sum_{t=2}^{T} \max_{\boldsymbol{x} \in \mathcal{V}} |\ell_t(\boldsymbol{x}) - \ell_{t-1}(\boldsymbol{x})| \ . \tag{3}$$

In the remaining we will use the shorthands C_T for $C_T(\boldsymbol{u}_{1:T})$ and V_T for $V_T(\ell_{1:T})$ when the context is clear. Another important definition is the one of *restricted dynamic regret* [Besbes et al., 2015; Jadbabaie et al., 2015; Yang et al., 2016]. In the restricted setting, the sequence of comparators is given by the local minimizers of the loss functions, i.e., $\boldsymbol{u}_{1:T} = (\boldsymbol{u}_1^*, \dots, \boldsymbol{u}_T^*)$, where $\boldsymbol{u}_t^* = \operatorname{argmin}_{\boldsymbol{x} \in \mathcal{V}} \ell_t(\boldsymbol{x})$.

In this work, we focus on online learning in the dynamic setting, considering the *full-information* feedback, where in every round the loss function is revealed. We show a number of results on upper and lower bounds. First, while a simple strategy can achieve an upper bound of $\mathcal{O}(V_T)$ on the restricted dynamic regret, to the best of our knowledge a lower bound for this setting is still not available. Our first contribution in Section 4 is indeed to provide a lower bound for the setting of restricted dynamic regret of $\Omega(V_T)$. Next, we analyse the case of general dynamic regret, as defined in Eq. (1). We provide a lower bound in terms of temporal variability of $\Omega(V_T)$, which shows that the general case is not harder than the restricted case. In Section 5, using recent advances in the analysis of implicit updates in online learning we design an algorithm which is adaptive to both C_T and V_T. Most developments in online learning have been driven by the use of the two paradigms of *Online Mirror Descent* (OMD) and *Follow The Regularized Leader* (FTRL) (see the surveys Shalev-Shwartz [2012]; Orabona [2019]). Both of them usually achieve the same regret bounds thanks to the standard linearization trick: given the convexity of the loss functions one can exploit the fact that $\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle$, where $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$ is a subgradient of the loss function. One can therefore shift its goal to minimize this new objective over time. Importantly, using gradients gives rise to efficient algorithms in practice. On the other hand, one can choose to not use gradients in the optimization process but the loss function instead. This is the approach taken by implicit algorithms. Using an adaptation of OMD to the implicit case, we will show how to achieve the optimal bound for the dynamic regret when the path-length of the class of comparators is fixed in advance. Finally, when the complexity of the class of comparators is not fixed in advance in terms of path-length (i.e., C_T is not fixed beforehand), in Section 6 we show how one can combine existing algorithms and get the optimal bound of $\mathcal{O}(\min\{V_T, \sqrt{T(1+C_T)}\})$ for any possible value of C_T.

2 Related work

In this section we are going to review the two lines of work most related to ours: algorithms designed for non-stationary environments and implicit updates in online learning. In the following, we recap existing results and highlight both similarities and differences compared to this paper.

Dynamic Regret. The notion of dynamic regret was first introduced in the seminal work of Zinkevich [2003], which first proved that *Online Gradient Descent* incurs a regret bound of $\mathcal{O}(\sqrt{T}(1 + C_T))$. This result was later extended by Hall and Willett [2013] considering a modified (and possibly richer) definition of path-length. A lower bound involving the path-length of $\Omega(\sqrt{T(1+C_T)})$ is shown in Zhang et al. [2018a], which also provide an algorithm which matches it. However, when considering the restricted case a more favourable bound is achievable, since Yang et al. [2016] showed a lower bound of $\Omega(C_T)$, with a simple strategy achieving a matching upper bound. In contrast, in this paper we show that when considering the temporal variability in the *full-information* feedback, that is when we receive the loss function rather than its gradient only, the setting does not become harder when shifting from the restricted to general case. Besbes et al. [2015] provided an analysis of restarted gradient descent in the setting of stochastic optimization with noisy gradients which incurs $\mathcal{O}(T^{2/3}(V_T' + 1)^{1/3})$, where V_T' is an upper bound on V_T known in advance. Jadbabaie et al. [2015] gave an algorithm achieving a restricted dynamic regret of[2] $\tilde{\mathcal{O}}(\sqrt{G_T} + \min(\sqrt{(G_T+1)C_T}, ((G_T + 1)T)^{1/3}(V_T+1)^{2/3}))$, where $G_T = \sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{x}_t) - \boldsymbol{p}_t\|_\star^2$ and $\boldsymbol{p}_1, \dots, \boldsymbol{p}_T$ is a predictable sequence computable at the start of round t. Importantly, this bound is obtained without prior knowledge of D_T, C_T and V_T but under the assumption that all of them can be *observable*. If one limits the algorithm to not use predictable sequences, then the bound reduces to $\tilde{\mathcal{O}}(\sqrt{T} + \min\{\sqrt{T(1+C_T)}, T^{1/3}(V_T+1)^{2/3})\}$. In Section 5, we design an algorithm similar in spirit to the one from Jadbabaie et al. [2015], which incurs an improved regret bound of $\min\{\sqrt{T(1+C_T)}, V_T\}$ when C_T is fixed in advance.

A parallel line of work on non-stationary environments involves the study of the weakly and strongly-adaptive

[2] The $\tilde{\mathcal{O}}$ notation hides poly-logarithmic terms.

regret [Hazan and Seshadhri, 2007; Daniely et al., 2015], which aims to minimize the static regret over any possible (sub)interval over the time horizon T. Importantly, it has been shown that strongly-adaptive regret bounds imply dynamic regret bounds. Very recently, Cutkosky [2020] provided a strongly-adaptive algorithm that achieves the optimal dynamic regret bound in terms of path-length, for any sequence of comparators. Importantly, the algorithm is *parameter-free*, meaning that it does not have to know the values of C_T in advance. On the other hand, in Zhang et al. [2018b] it is shown that dynamic regret of the strongly adaptive algorithm given in Jun et al. [2017] is order of $\tilde{\mathcal{O}}(\max\{\sqrt{T}, T^{2/3}(V_T+1)^{1/3}\})$. We suppose one can probably apply the same analysis given in Zhang et al. [2018b] to the algorithm from Cutkosky [2020], in order to get a bound of the same order, which is however adaptive to the gradients of the loss functions. To summarize, using existing strongly-adaptive algorithms, one could achieve a dynamic regret bound against any sequence of comparators of $\tilde{\mathcal{O}}(\min\{\sqrt{T(1+C_T)}, \max\{\tilde{\mathcal{O}}(\max\{\sqrt{T}, T^{2/3}(V_T+1)^{1/3}\})\}\})$, for any sequence $\boldsymbol{u}_{1:T}$ without knowing its path-length in advance. However, given our lower bound in V_T, this bound is suboptimal. In Section 6, we show instead how one can achieve an improved bound of $\mathcal{O}(\min\{\sqrt{T(1+C_T)}, V_T\}$ combining existing algorithms. Finally, better bounds can be achieved making additional assumption. For example, improved rates for the dynamic regret were shown in Mokhtari et al. [2016] in the case of strongly-convex functions and in Zhang et al. [2017] when it is possible to query the function multiple times per round.

Implicit Algorithms. Implicit algorithms are known in the optimization literature as proximal methods [Parikh and Boyd, 2014] and can be traced back to the work of Moreau [1965]. In the online learning community, they have been introduce in Kivinen and Warmuth [1997]. While deeply effective in practice, it proved difficult to show the superiority of these algorithms in theory in terms of regret bounds [McMahan, 2010; Kulis and Bartlett, 2010]. However, in a recent work Campolongo and Orabona [2020] showed that implicit updates can outperform their linearized counterparts when the temporal variability is low, at least in the static setting. The dynamic regret for proximal algorithms in the online setting has been also studied in the case of strongly convex losses in Dixit et al. [2019] and for composite losses in Ajalloeian et al. [2020]. Both these works assume different notions of feedback from ours. Finally, it is worth mentioning recent work on smoothed online convex optimization which uses implicit algorithms such as Chen et al. [2018]; Argue et al. [2020]. However, in this setting the loss function is observed before committing to a choice $\boldsymbol{x}_t$ and therefore results are not comparable.

Algorithm 1: Greedy optimizer

Require: Non-empty closed convex set $\mathcal{V} \subset X \subset \mathbb{R}^d$, $\boldsymbol{x}_1 \in \mathcal{V}$.
1: **for** $t = 1, \dots, T$ **do**
2: Output $\boldsymbol{x}_t \in \mathcal{V}$
3: Receive $\ell_t : \mathbb{R}^d \to \mathbb{R}$ and pay $\ell_t(\boldsymbol{x}_t)$
4: Update $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathcal{V}} \ell_t(\boldsymbol{x})$
5: **end for**

3 Definitions

For a function $f : \mathbb{R}^d \to (-\infty, +\infty]$, we define a *subgradient* of f in $\boldsymbol{x} \in \mathbb{R}^d$ as a vector $\boldsymbol{g} \in \mathbb{R}^d$ that satisfies $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x}\rangle$, $\forall \boldsymbol{y} \in \mathbb{R}^d$. We denote the set of subgradients of f in $\boldsymbol{x}$ by $\partial f(\boldsymbol{x})$. We denote the *dual norm* of $\|\cdot\|$ by $\|\cdot\|_\star$. A proper function $f : \mathbb{R}^d \to (-\infty, +\infty]$ is μ-*strongly convex* over a convex set $V \subseteq \operatorname{int}\operatorname{dom} f$ w.r.t. $\|\cdot\|$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in V$ and $\boldsymbol{g} \in \partial f(\boldsymbol{x})$, we have $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$. Let $\psi : X \to \mathbb{R}$ be strictly convex and continuously differentiable on $\operatorname{int} X$. The *Bregman Divergence* w.r.t. ψ is $B_\psi : X \times \operatorname{int} X \to \mathbb{R}_+$ defined as $B_\psi(\boldsymbol{x}, \boldsymbol{y}) = \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \nabla\psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle$. We assume that ψ is strongly convex w.r.t. a norm $\|\cdot\|$ in $\operatorname{int} X$. We also assume w.l.o.g. the strong convexity constant to be 1, which implies

$$B_\psi(\boldsymbol{x}, \boldsymbol{y}) \geq \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \quad \forall \boldsymbol{x} \in X, \boldsymbol{y} \in \operatorname{int} X \ . \tag{4}$$

4 A closer look to the Temporal Variability

In this section, we turn our attention to the temporal variability of the loss functions in the full information setting. In particular, we are going to prove lower bounds in the dynamic scenario which highlight that the regret has to grow as $\Omega(V_T)$. Apparently, this result seems to be in contrast with the result given in Besbes et al. [2015], which reports a lower bound of $\Omega(V_T^{1/3}T^{2/3})$. However, it should be noted that in Besbes et al. [2015] the feedback is different and not directly comparable to our setting. Indeed, it is assumed only access to noisy functions and gradients and therefore their Theorem 2 is not applicable.

In the restricted setting with full information feedback, as noted in Jadbabaie et al. [2015] a simple strategy to achieve a bound in terms of temporal variability V_T is depicted in Algorithm 1. Basically, in each round it plays the minimizer of the observed loss function in the previous round. For any $\boldsymbol{u}_t \in \mathcal{V}$ this leads to the following

$$
\begin{aligned}
&\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}_t)) \\
&\qquad = \sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_{t+1}) - \ell_t(\boldsymbol{u}_t)) \\
&\qquad \leq \sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1})) \\
&\qquad = \ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + \sum_{t=2}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_{t-1}(\boldsymbol{x}_t)) \\
&\qquad \leq \ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T \\
&\qquad = \max(V_T, \mathcal{O}(1)) \ .
\end{aligned}
$$

On the other hand, in Yang et al. [2016] it is shown that the same strategy achieves an upper bound of $\mathcal{O}(\max(C_T, 1))$ when one takes into account the path-length. While the result regarding the path-length is tight, one might wonder if the same could be said about the temporal variability. Next, we show that this is indeed the case.

Theorem 4.1. *Let $\mathcal{V} = [-1, 1]$, and C be a positive constant independent of T. Then, for any algorithm $\mathcal{A}$ on $\mathcal{V}$, and any $\sigma \in (1/\sqrt{T}, 1)$, there exists a sequence of loss functions $\ell_1, \dots, \ell_T$ with temporal variability less than or equal to $2\sigma T$ such that*

$$R(\boldsymbol{u}_{1:T}) \geq C V_T^{\gamma} \ , \tag{5}$$

for any $\gamma \in (0, 1)$.

The proof of this result is an adaptation of Proposition 1 from Yang et al. [2016]. We report it in Appendix A. Note that in Besbes et al. [2015] it is proven that in order to have sublinear dynamic regret in the restricted setting, one must assume that $V_T = o(T)$. Eq. (5) implies that it is impossible to achieve a dynamic regret bound better that $\mathcal{O}(V_T^{\gamma})$, with $\gamma < 1$.

Based on Theorem 4.1 and the result from Yang et al. [2016], it follows that the best policy to adopt in the restricted setting with full-information is the greedy algorithm given at the beginning of this section, which at round t plays the minimizer of the previous loss function. However, the restricted dynamic regret could be a poor benchmark, as an upper bound for it could be very loose for another sequence of comparators whose path length is smaller. Furthermore, as explained in Zhang et al. [2018a] it could be meaningless in the problem of statistical machine learning, where the loss functions ℓ_t are sampled independently from the same distribution and minimizing the restricted dynamic regret could lead to overfitting. Therefore, we will shift our attention to the general dynamic regret in the following. In this case, a lower bound in terms of path-length has been provided in Zhang et al. [2018a]. However, a lower bound in terms of temporal variability is still missing. In the next theorem, we provide a lower bound on the general dynamic regret in terms of V_T.

Theorem 4.2. *For any deterministic online algorithm $\mathcal{A}$ and any $\tau \in [0, TD]$, there exists a sequence of comparators $\boldsymbol{u}_1, \dots, \boldsymbol{u}_T$ and a sequence of loss function $\ell_1, \dots, \ell_T$ such that $V_T(\ell_1, \dots, \ell_T) \leq \tau$*

$$R_T(\boldsymbol{u}_1, \dots, \boldsymbol{u}_T) \geq V_T \ . \tag{6}$$

Proof sketch. We remind that a lower bound in terms of temporal variability for the static regret on constrained domains has been proved in Campolongo and Orabona [2020], which states that for every $\tau \geq 0$, there exists a sequence of loss functions such that V_T is equal to τ and the regret satisfies $R_T(\boldsymbol{u}) \geq \tau$. Here the idea is to divide the time horizon into intervals where the sequence of functions stays almost fixed and apply the lower bound for the static setting to each subinterval. We defer to Appendix A for a formal proof.

Based on the last result, to the best of our knowledge there are no algorithms which achieve the optimal dynamic regret bound of $\mathcal{O}(\min\{V_T, \sqrt{T(1 + C_T)}\})$ in the full information setting. Indeed, algorithms designed for dynamic regret such as Jadbabaie et al. [2015] and Besbes et al. [2015] have a dependency of $V_T^{1/3}$. The same holds true for strongly adaptive algorithms as well, as shown in Zhang et al. [2018b]. This is not really surprising: a bound of $\mathcal{O}(V_T)$ would imply constant regret in the case the loss functions are fixed, i.e. $\ell_t = \ell$ for all t. In this case, using an online-to-batch conversion [Cesa-Bianchi et al., 2004] would result in a convergence rate of $\mathcal{O}(1/T)$. However, this would be in contrast with the lower bound by Nesterov [2013] on non-smooth batch black-box optimization. Unfortunately, all the algorithms mentioned above make use of gradients and therefore are subject to the lower bound of $\mathcal{O}(1/\sqrt{T})$. Therefore, in the dynamic setting using only gradients the lower bound given in Besbes et al. [2015] would probably continue to hold even if the feedback structure is changed (i.e. the noise assumptions are removed). However, not all is lost: in the next section, we illustrate how one can achieve a bound in V_T using an algorithm which makes full use of the loss function (and not just its gradient).

5 Implicit updates in dynamic environments

In a recent work, Campolongo and Orabona [2020] showed how a modified version of OMD with implicit updates can achieve a regret bound in the static setting which is order of $\mathcal{O}(\min(V_T, \sqrt{T}))$. OMD with

Algorithm 2: Dynamic IOMD

Require: Non-empty closed convex set $\mathcal{V} \subset X \subset \mathbb{R}^d$, $\psi : X \to \mathbb{R}$, $\boldsymbol{x}_1 \in \mathcal{V}$, γ such that $B_\psi(\boldsymbol{x}, \boldsymbol{z}) - B_\psi(\boldsymbol{y}, \boldsymbol{z}) \le \gamma \|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{V}$, non increasing sequence $(\eta_t)_{t=1}^T$.

for $t = 1, \dots, T$ **do**
  Output $\boldsymbol{x}_t \in \mathcal{V}$
  Receive $\ell_t : \mathbb{R}^d \to \mathbb{R}$ and pay $\ell_t(\boldsymbol{x}_t)$
  Update
  $\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{V}} \ \ell_t(\boldsymbol{x}) + B_\psi(\boldsymbol{x}, \boldsymbol{x}_t)/\eta_t$
end for

implicit updates is depicted in Algorithm 2. The only difference with its linearized counterpart is in the update rule, which uses directly the loss rather than its (sub)gradient in $\boldsymbol{x}_t$:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x} \in \mathcal{V}} \ \ell_t(\boldsymbol{x}) + B_\psi(\boldsymbol{x}, \boldsymbol{x}_t)/\eta_t$$

In this section, we show how this algorithm automatically satisfies a bound on the dynamic regret. In order to do that, we require a Lipschitz continuity condition on the Bregman divergence. Using this assumption, we can get the bound for the dynamic regret shown in the next theorem.

Theorem 5.1. *Let $\mathcal{V} \subset X \subset \mathbb{R}^d$ be non-empty closed convex sets, $\psi : X \to \mathbb{R}$, and $\boldsymbol{x}_1 \in \mathcal{V}$. Assume there exists $\gamma \in \mathbb{R}$ such that $B_\psi(\boldsymbol{x}, \boldsymbol{z}) - B_\psi(\boldsymbol{y}, \boldsymbol{z}) \le \gamma \|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{V}$. Define $D^2 \triangleq \max_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}} B_\psi(\boldsymbol{x}, \boldsymbol{y})$. Let $(\eta_t)_{t=1}^T$ be a non-increasing sequence. Then, the regret of Algorithm 2 against any sequence $\boldsymbol{u}_{1:T}$ with $\boldsymbol{u}_t \in \mathcal{V}$ for all t is bounded as follows*

$$R_T(\boldsymbol{u}_{1:T}) \le \frac{D^2}{\eta_T} + \gamma \sum_{t=2}^T \frac{\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|}{\eta_t} + \sum_{t=1}^T \delta_t, \quad (7)$$

where $\delta_t = \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)/\eta_t$.

Proof. Let $\boldsymbol{g}'_t \in \partial \ell_t(\boldsymbol{x}_{t+1})$. From the update rule of Algorithm 2 we have that

$$\begin{aligned}
&\eta_t(\ell_t(\boldsymbol{x}_{t+1}) - \ell_t(\boldsymbol{u}_t)) \\
&\quad \le \langle \eta_t \boldsymbol{g}'_t, \boldsymbol{x}_{t+1} - \boldsymbol{u}_t \rangle \\
&\quad \le \langle \nabla\psi(\boldsymbol{x}_t) - \nabla\psi(\boldsymbol{x}_{t+1}), \boldsymbol{x}_{t+1} - \boldsymbol{u}_t \rangle \\
&\quad = B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t), \quad (8)
\end{aligned}$$

where the first inequality follows from the convexity of the loss functions, while the second from the first-order optimality condition.

Now, we consider the first two terms of the r.h.s. of Eq. (8). Using the Lipschitz continuity condition on the Bregman divergence and the fact that η_t is non-increasing over time, we get

$$\begin{aligned}
&\sum_{t=1}^T \frac{1}{\eta_t}(B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_{t+1})) \\
&\quad \le \frac{D^2}{\eta_1} + \sum_{t=2}^T \left(\frac{B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}_{t-1}, \boldsymbol{x}_t)}{\eta_{t-1}} \right) \\
&\quad = \frac{D^2}{\eta_1} + \sum_{t=2}^T \left(\frac{B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}_{t-1}, \boldsymbol{x}_t)}{\eta_t} \right. \\
&\qquad \left. + \frac{B_\psi(\boldsymbol{u}_{t-1}, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}_{t-1}, \boldsymbol{x}_t)}{\eta_{t-1}} \right) \\
&\quad \le \frac{D^2}{\eta_1} + \gamma \sum_{t=2}^T \frac{\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|}{\eta_t} \\
&\qquad + \sum_{t=2}^T B_\psi(\boldsymbol{u}_{t-1}, \boldsymbol{x}_t) \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) \\
&\quad \le \frac{D^2}{\eta_1} + D^2 \left(\frac{1}{\eta_T} - \frac{1}{\eta_1} \right) + \gamma \sum_{t=2}^T \frac{\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|}{\eta_t} \\
&\quad = \frac{D^2}{\eta_T} + \gamma \sum_{t=2}^T \frac{\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|}{\eta_t} \ .
\end{aligned}$$

Adding $\ell_t(\boldsymbol{x}_t)$ on both sides of Eq. (8) and summing over time yields the regret bound in Eq. (7). $\square$

Notice that the Lipschitz continuity assumption is not a strong requirement. For example in the case of learning with expert advice, we have that $\gamma = \mathcal{O}(\ln T)$ if we use a "clipped" simplex (details in Appendix B). In general, it is often not required to modify the domain of interest. Indeed, when the function ψ is Lipschitz on $\mathcal{V}$, the Lipschitz condition on the Bregman divergence is automatically satisfied.

Using a fixed learning rate and assuming to fix in advance an upper bound to the path length of the sequence of comparators, it is straightforward to show a regret bound on the dynamic regret.

Corollary 5.1.1. *Let $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$ and assume $\|\boldsymbol{g}_t\|_\star \le L$. Let $C \ge 0$ be a fixed positive constant, then using $\eta_t = \eta = \frac{1}{L}\sqrt{\frac{2(D^2 + \gamma C)}{T}}$, Algorithm 2 gives the following regret bound against all sequence $\boldsymbol{u}_{1:T}$ whose path length C_T is upper bounded by C*

$$R_T(\boldsymbol{u}_{1:T}) \le L\sqrt{(D^2 + \gamma C)T} \ . \quad (9)$$

Proof. Using a constant learning rate $\eta_t = \eta$ for all t, from Eq. (7) we get

$$R_T(\boldsymbol{u}_{1:T}) \le \frac{D^2}{\eta} + \gamma \sum_{t=2}^T \frac{\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|}{\eta} + \sum_{t=1}^T \delta_t$$

$$\leq \frac{D^2}{\eta} + \gamma \sum_{t=2}^{T} \frac{\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|}{\eta} + \sum_{t=1}^{T} \frac{\eta}{2} \|\boldsymbol{g}_t\|_\star^2$$
$$\leq \frac{D^2}{\eta} + \frac{\gamma}{\eta} C + \eta \frac{L^2 T}{2},$$

where in the second inequality we used Theorem 5.3 from Campolongo and Orabona [2020], while the last inequality follows from the assumption that $\|\boldsymbol{g}_t\|_\star \leq L$. Therefore, if C_T is fixed beforehand, using the given learning rate we get the bound given in Eq. (9). □

From the above result, we can see that the upper bound matches the lower bound given in Zhang et al. [2018a] for all sequence whose path-length C_T equals C. However, for the case of restricted regret this result is suboptimal in light of the lower bound given in Theorem 4.1 for the restricted dynamic regret. Nonetheless, making additional assumptions such as the smoothness of the losses, it is not difficult to make the bound tight, as shown in Theorem 3 of Yang et al. [2016].

On the other hand, while Corollary 5.1.1 shows a regret bound in terms of the path-length C_T, it is still not clear how the algorithm can adapt to the temporal variability. Fortunately, Campolongo and Orabona [2020] showed that a carefully chosen adaptive learning rate gives a bound in terms of V_T. So, the next theorem shows that their algorithm *AdaImplicit* achieves the dynamic regret bound we are interested in.

Theorem 5.2. *Under the assumptions of Theorem 5.1, for any $u \in \mathcal{V}$, running Algorithm 2 with $1/\eta_t = \lambda_t = \frac{1}{\beta^2}\sum_{i=1}^{t} \delta_t$ guarantees*

$$R_T(\boldsymbol{u}_{1:T}) \leq \frac{D^2 + \beta^2 + \gamma C_T}{\beta^2} \min(B_1, B_2),$$

where $B_1 = \ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T$ and $B_2 = \sqrt{(2D^2 + \beta^2)\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\star^2}$

Proof. We can rewrite the bound in Eq. (7) as follows

$$R_T(\boldsymbol{u}_{1:T}) \leq \lambda_T D^2 + \gamma \sum_{t=2}^{T} \lambda_t \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\| + \beta^2 \lambda_{T+1}$$
$$\leq (D^2 + \beta^2)\lambda_{T+1} + \gamma \lambda_{T+1} \sum_{t=2}^{T} \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|$$
$$= (D^2 + \beta^2 + \gamma C_T)\lambda_{T+1},$$

where in the second inequality we have used the fact that $(\lambda_t)_{t=1}^T$ is an increasing sequence.

From the choice of λ_t, we have that

$$\beta^2 \lambda_{T+1} \leq \ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T, \tag{10}$$

where $V_T = \sum_{t=2}^{T} \max_{\boldsymbol{x} \in V} \ell_t(\boldsymbol{x}) - \ell_{t-1}(\boldsymbol{x})$. On the other hand, using Lemma 6.1 from Campolongo and Orabona [2020] we get

$$\beta^2 \lambda_{T+1} \leq \sqrt{(2D^2 + \beta^2) \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\star^2}\,. \tag{11}$$

Therefore, putting together Eqs. (10) and (11) we get the stated bound. □

From the bound in Theorem 5.2, in case one wants to compete against any sequence of comparators whose path length C_T is fixed beforehand, then she can use this value in the tuning of λ_{T+1} in the same spirit of Corollary 5.1.1. Therefore, setting $\beta^2 = (D^2 + \gamma C_T)$, we get

Corollary 5.2.1. *Let $C \geq 0$ be a positive constant. Then, running Algorithm 2 with $\beta^2 = (D^2 + \gamma C)$, the dynamic regret against any sequence $\boldsymbol{u}_{1:T}$ whose path length is less or equal than C is upper bounded as follows*

$$R_T(\boldsymbol{u}_{1:T}) \leq \min \Big\{ 2(\ell_1(\boldsymbol{x}_1) - \ell_t(\boldsymbol{x}_{T+1}) + V_T), \; 2\sqrt{(3D^2 + \gamma C)\textstyle\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\star^2} \Big\}\,.$$

The above results therefore gives us an algorithm whose regret bound on the general dynamic regret is order of $\mathcal{O}(V_T, \sqrt{T(1 + C_T)})$ when we fix the path-length of the sequences of comparators, therefore matching the lower bounds for both the path-length and temporal variability. Next, we are going to show some applications of this algorithm.

5.1 Applications

Competing with strategies. The result given in the previous paragraph was limited to all the fixed sequence of comparators whose path-length is fixed beforehand. Following Jadbabaie et al. [2015], the approach given above can be generalized to any sequence of comparators $(\boldsymbol{u}_{1:T})$ whose path length C_T can be calculated on the fly. Formally, we can denote by Π a set of strategies. Each $\pi = (\pi_1, \ldots, \pi_T) \in \Pi$ is a sequence of mappings such that $\pi_t : \mathcal{F}^{t-1} \to \mathcal{V}$, where $\mathcal{F}^{t-1}$ denotes the history of the game up to time $t-1$. For example Π could be the set of strategies where the sequence of comparators is allowed to switch only k times, i.e. $\Pi = \{\boldsymbol{u}_{1:T}, \boldsymbol{u}_t \in \mathcal{V} : \sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{u}_t \neq \boldsymbol{u}_{t-1}\} \leq k-1\}$. In this case the path-length C_T might not be known a priori, but we can adapt Algorithm 2 to this setting using a doubling trick. We depict this strategy in Algorithm 3. Next, we provide a theorem which gives a regret bound, whose proof can be found in Appendix D.

Algorithm 3: Dynamic AdaImplicit

Require: Non-empty closed convex set $V \subset X \subset \mathbb{R}^d$, $\psi : X \to \mathbb{R}$, $\boldsymbol{x}_1 \in V$, γ such that $B_\psi(\boldsymbol{x}, \boldsymbol{z}) - B_\psi(\boldsymbol{y}, \boldsymbol{z}) \leq \gamma \|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in V$, $\beta_0^2 > 0$

```
i ← 0, λ_1^0 ← 0, Q_0 ← √2 D, C_0 ← 0
for t = 1, ..., T do
    Output x_t ∈ V
    Receive ℓ_t : R^d → R and pay ℓ_t(x_t)
    Update C_i ← C_i + ||u*_t − u*_{t−1}||, where
        u*_t := argmin_{x∈V} ℓ_t(x)
    if C_i > Q_i then
        i ← i + 1
        Q_i ← √2 D 2^i, λ^i_{t+1} ← 0, C_i ← 0,
        β_i^2 ← D^2 + γ Q_i
        Update x_{t+1} ← x_t
    else
        Update
        x_{t+1} ← arg min_{x∈V} ℓ_t(x) + λ^i_t B_ψ(x, x_t)
        Set δ_t ← ℓ_t(x_t) − ℓ_t(x_{t+1}) − λ^i_t B_ψ(x_{t+1}, x_t)
        Update λ^i_{t+1} ← λ^i_t + (1/β_i^2) δ_t
    end if
end for
```

Theorem 5.3. *Let $\mathcal{V} \subset \mathbb{R}^d$ be a non-empty closed convex set. Fix a class of strategies Π, where each strategy $\pi \in \Pi$ is such that $\pi = (\pi_1, \dots, \pi_T)$ and $\pi_t : \mathcal{F}^{t-1} \to \mathcal{V}$. Assume Algorithm 3 is run for N epochs. Then, under the assumptions of Theorem 5.1 the regret against any strategy $\pi \in \Pi$ is bounded as*

$$R_T(\pi) \leq (2 + c) \min \Bigg((\ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T), \sqrt{\left(3D^2 \left(\log_2 \tfrac{C_T}{\sqrt{2}D} + 1\right) + \gamma C_T\right) \textstyle\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2} \Bigg),$$

where $c \triangleq \frac{\sqrt{2}}{D + \gamma\sqrt{2}}$ and $C_T = \sum_{t=2}^{T-1} \|\pi_t(\ell_{1:t-1}) - \pi_{t-1}(\ell_{1:t-2})\|$.

Composite Losses. Next, we are going to show an application of our algorithm which might be relevant in practical scenarios. Assume that the losses received are composed by two parts: one convex changing over time and the other one fixed and known to the algorithm. This losses are called *composite* [Duchi et al., 2010]. This setting was also studied in the implicit case in Song et al. [2018] for the static regret. For example, we might have $\ell_t(\boldsymbol{x}) = \tilde{\ell}_t(\boldsymbol{x}) + \beta \|\boldsymbol{x}\|_1$. In this case, considering a bounded domain $\mathcal{V}$ the update rule for Algorithm 2 will be as follows

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x} \in V} \ \tilde{\ell}_t(\boldsymbol{x}) + \beta \|\boldsymbol{x}\|_1 + B_\psi(\boldsymbol{x}, \boldsymbol{x}_t), \qquad (12)$$

which will promote sparsity in our model. Note that even if in practice often there are no closed form solutions to the minimization problem in Eq. (12), approximate solutions can be found reasonably fast (see for example Song et al. [2018]). We show in Appendix C that *AdaImplicit* [Campolongo and Orabona, 2020] designed for the static regret already satisfies a regret bound order of $\mathcal{O}(\min(V_T, \sqrt{T}))$ (improving over the existing result of Song et al. [2018]). Following the template of the previous analysis, it is easy to extend the above case to the dynamic scenario. In particular, using Algorithm 3 we can give the following theorem, whose proof is reported in Appendix C.

Theorem 5.4. *Let $\ell_t(\boldsymbol{x}) = \tilde{\ell}_t(\boldsymbol{x}) + \beta r(\boldsymbol{x})$ and $\boldsymbol{g}_t \in \partial \tilde{\ell}_t(\boldsymbol{x}_t)$. Let $\tilde{\delta}_t = \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)$. Given a constant $C \geq 0$, define $\lambda_t \triangleq \frac{1}{\alpha^2} \sum_{i=1}^{t-1} \tilde{\delta}_i$, where $\alpha^2 = D^2 + \gamma C$. Then, under the assumptions of Theorem 5.1 the regret of Algorithm 2 run with $1/\eta_t = \lambda_t$ against any sequence of comparators $\boldsymbol{u}_{1:T}$ whose path-length is given by $C_T = C$ is bounded as*

$$R_T(\boldsymbol{u}_{1:T}) \leq \min \Big\{ 2(\ell_1(\boldsymbol{x}_1) - \ell_{T+1}(\boldsymbol{x}_{T+1}) + V_T), 2\sqrt{(3D^2 + \gamma C_T) \textstyle\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2} + \beta(r(\boldsymbol{x}_1) - r(\boldsymbol{x}_{T+1})) \Big\}$$

where $\tilde{\boldsymbol{g}}_t \in \partial \tilde{\ell}_t(\boldsymbol{x}_t)$.

All the results up to this point are given under the assumption that the class of strategies we want to compete against is fixed before the start of the game, i.e., C_T is known beforehand or can be computed through observable quantities. This can be limiting in practice. On the other hand, we would like to note that online learning algorithms can be used in the offline setting as well. At least in the case of static regret, one can still guarantee convergence rates using standard arguments for online-to-batch conversions [Cesa-Bianchi et al., 2004]. Therefore, in the case of offline optimization, one should be able to select the optimal β in Corollary 5.2.1 by trying different values, in order to guarantee the regret bound given in Corollary 5.2.1

Nevertheless, in a truly realistic online setting, knowing the right C_T beforehand might be impossible. Therefore, in the next section we are going to provide an algorithm which can adapt to the values of C_T for any possible sequence of comparators but at the same time guarantees a bound in V_T.

6 Adapting to different path-lengths

Our approach to achieve the optimal bound for any given sequence of comparators is to combine different existing algorithms. In particular, using either the parameter-free algorithm given in Cutkosky [2020] or the one from Zhang et al. [2018a] (which requires the knowledge of T) we can achieve a bound of $\mathcal{O}(\sqrt{T(1 + C_T)})$ for all possible sequences. On the

Algorithm 4: Anytime $(\mathcal{A}, \mathcal{B})$-PROD

Require: Algorithms $\mathcal{A}, \mathcal{B}$, $\eta_1 = 1/2$, $w_{1,\mathcal{A}} = w_{1,\mathcal{B}} = 1/2$.

for $t = 1, \dots, T$ **do**
  Let $s_t = \frac{\eta_t w_{t,\mathcal{A}}}{\eta_t w_{t,\mathcal{A}} + w_{t,\mathcal{B}}/2}$
  Get $\boldsymbol{a}_t$ from $\mathcal{A}$ and $\boldsymbol{b}_t$ from $\mathcal{B}$ and predict $\boldsymbol{x}_t = s_t \boldsymbol{a}_t + (1 - s_t)\boldsymbol{b}_t$
  Receive $\ell_t : \mathbb{R}^d \to [0, 1]$ and pay $\ell_t(\boldsymbol{x}_t)$
  Feed ℓ_t to $\mathcal{A}$ and $\mathcal{B}$
  Set $\eta_{t+1} = \sqrt{(1 + \sum_{i=1}^t (\ell_i(\boldsymbol{b}_i) - \ell_i(\boldsymbol{a}_i))^2)^{-1}}$
  Compute $\delta_t = \ell_t(\boldsymbol{b}_t) - \ell_t(\boldsymbol{a}_t)$ and set $w_{t+1,\mathcal{A}} = w_{t,\mathcal{A}}(1 + \eta_t \delta_t)^{\eta_{t+1}/\eta_t}$
end for

other hand, in order to achieve a bound on the temporal variability, we can simply use the greedy algorithm given in the Section 4, which in every step plays the minimizer of the last seen loss function. In order to combine these different algorithms and get the best result, one can think of using an algorithm for learning with expert advice. However, a generic algorithm for this setting is not sufficient, since in the worst-case we would have a regret bound of $\sqrt{T}$, which could be worse than V_T if the latter is low. Therefore, we are going to use a modification of the *Prod* algorithm [Gaillard et al., 2014] proposed in Sani et al. [2014] and depicted in Algorithm 4. This algorithm takes in input two base algorithms $\mathcal{A}$ and $\mathcal{B}$ and guarantees a regret which is (almost) constant against $\mathcal{B}$ and $\mathcal{O}(\sqrt{T} \ln \ln T)$ against $\mathcal{A}$ in the worst case. The idea here is to use the strongly adaptive algorithm from Cutkosky [2020] as algorithm $\mathcal{A}$, and the greedy algorithm in Algorithm 1 as $\mathcal{B}$. In particular, we have the following theorem for the strongly-adaptive algorithm from Cutkosky [2020].

Theorem 6.1 ([Cutkosky, 2020])**.** *There exists a strongly-adaptive which given any sequence of comparators $\boldsymbol{u}_{1:T}$ with path-length C_T achieves the following regret bound*

$$R_T(\boldsymbol{u}_{1:T}) = \tilde{\mathcal{O}}\left[C_T + D + \sqrt{D(C_T + D)\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\star^2}\right] .$$

Further, we have that $C_T \leq \sqrt{DC_T T}$.

Importantly, in order to use Algorithm 4 we need the losses to be bounded in the interval $[0, 1]$. A lower bound to the loss function is often satisfied for many practical cases, for example with non-negative losses. Given that we assume a bounded domain and bounded gradients then the loss is also automatically upper bounded. Therefore, instead of feeding Algorithm 4 with the original losses, we first scale them by the given upper bound. The formal result on the performance of the algorithm is stated in the next theorem.

Theorem 6.2. *Let $\mathcal{V} \subset \mathbb{R}^d$ be a closed convex set. Furthermore, assume its diameter is bounded by D, $\boldsymbol{0} \in \mathcal{V}$ and the losses are non-negative. Let $\boldsymbol{g}_t(\boldsymbol{x}) \in \partial \ell_t(\boldsymbol{x})$ and assume that $\max_{\boldsymbol{x} \in \mathcal{V}, t \in [T]} \|\boldsymbol{g}_t(\boldsymbol{x})\|_\star \leq L$. Then, running Algorithm 4 with the modified losses $\tilde{\ell}_t(\boldsymbol{x}) = \ell_t(\boldsymbol{x})/c$, where $c \geq LD$, the losses ℓ_t are non-negative, $\mathcal{A}$ the strongly-adaptive algorithm from Cutkosky [2020], and $\mathcal{B}$ the greedy algorithm, guarantees*

$$R_T(\boldsymbol{u}_{1:T}) = \tilde{\mathcal{O}}(\min(V_T, \sqrt{TD(C_T + D)} + \sqrt{C})),$$

where C is an upper bound to the loss of algorithm $\mathcal{A}$ and $\tilde{O}$ hides poly-logarithmic terms in T.

The proof of this last theorem is reported in Appendix C.1. In the worst case, assuming a high temporal variability and that the loss of algorithm $\mathcal{A}$ is linear in T, we have that the regret of our combiner algorithm is order of $\tilde{\mathcal{O}}(\sqrt{TD(C_T + D)} + \sqrt{T})$. However, it should be noted that the algorithm from Cutkosky [2020] is adaptive to the sum of the gradients $\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2$ and actually achieves a *second-order* bound and therefore should have a better bound in more favourable scenarios.

7 Conclusion

In this work, we have shown that existing bounds in the dynamic setting with full information feedback can be improved, by establishing lower bounds on the dynamic regret in terms of temporal variability and showing algorithms with matching upper bounds. In particular, we designed an algorithm using implicit updates that can adapt to both the temporal variability and the path-length of the sequence of comparators. Furthermore, when the desired path-length is not fixed in advance and cannot be observed on the fly, we showed how to combine existing algorithms in order to achieve the optimal bound. Despite the appealing regret bound achieved in the latter setting, the resulting algorithm might not be practical in a realistic scenario, since it basically has to run 3 algorithms in parallel. Furthermore, as observed in previous work, all strongly-adaptive algorithms [Cutkosky, 2020; Jun et al., 2017] have a running time of $\mathcal{O}(T \ln T)$ and it is currently an open question if it can be improved. Future research directions therefore could aim at designing faster and more practical algorithms which can adapt to unknown path-lengths, or in alternative prove that this goal cannot be achieved.

References

Amirhossein Ajalloeian, Andrea Simonetto, and Emiliano Dall'Anese. Inexact online proximal-gradient method for time-varying convex optimization. In *2020 American Control Conference (ACC)*, pages 2850–2857. IEEE, 2020.

C. J. Argue, Anupam Gupta, and Guru Guruganesh. Dimension-free bounds for chasing convex functions. In *Conference on Learning Theory, COLT 2020*, volume 125 of *Proceedings of Machine Learning Research*, pages 219–241. PMLR, 2020.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

Nicolò Campolongo and Francesco Orabona. Temporal variability in implicit online learning. *arXiv preprint arXiv:2006.07503*, 2020.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Niangjun Chen, Gautam Goel, and Adam Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *Conference On Learning Theory*, pages 1574–1594, 2018.

Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In *International Conference on Machine Learning*, volume 2, 2020.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411, 2015.

Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. *IEEE Transactions on Signal Processing*, 67(5):1338–1352, 2019.

John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, pages 14–26. Citeseer, 2010.

Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196, 2014.

Eric Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pages 579–587, 2013.

Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007.

Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.

Kwang-Sung Jun, Francesco Orabona, Stephen Wright, and Rebecca Willett. Improved strongly adaptive online learning using coin betting. In *Artificial Intelligence and Statistics*, pages 943–951. PMLR, 2017.

Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, 132(1):1–63, 1997.

Brian Kulis and Peter L Bartlett. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 575–582, 2010.

H Brendan McMahan. A unified view of regularized dual averaging and mirror descent with implicit updates. *arXiv preprint arXiv:1009.3240*, 2010.

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

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, pages 810–818, 2014.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Chaobing Song, Ji Liu, Han Liu, Yong Jiang, and Tong Zhang. Fully implicit online learning. *arXiv preprint arXiv:1809.09350*, 2018.

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 *International Conference on Machine Learning*, pages 449–457, 2016.

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*, pages 732–741, 2017.

Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in neural information processing systems*, pages 1323–1333, 2018a.

Lijun Zhang, Tianbao Yang, Rong Jin, and Zhi-Hua Zhou. Dynamic regret of strongly adaptive methods. In *International Conference on Machine Learning*, pages 5882–5891, 2018b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

A Lower Bounds

Lower bound for restricted dynamic regret

Theorem 4.1 *Let $\mathcal{V} = [-1, 1]$, and C be a positive constant independent of T. Then, for any algorithm $\mathcal{A}$ on $\mathcal{V}$, and any $\sigma \in (1/\sqrt{T}, 1)$, there exists a sequence of loss functions $\ell_1, \dots, \ell_T$ with temporal variability less than or equal to $2\sigma T$ such that*

$$R(\boldsymbol{u}_{1:T}) \geq C V_T^{\gamma},$$

for any $\gamma \in (0, 1)$.

Proof. As done in Yang et al. [2016], we consider a simple 1-d problem and employ the following sequence of loss functions. Define $\ell_t(x_t) = \frac{1}{2}(x_t - \varepsilon_t)^2$, where $\varepsilon_1, \dots, \varepsilon_T$ is a sequence of random variables sampled uniformly at random between the two values $\{-\sigma, \sigma\}$. Note that we have $\mathbb{E}[\varepsilon_t] = 0$ and $\mathrm{Var}(\epsilon_t) = \mathbb{E}[\varepsilon_t^2] = \sigma^2$. Obviously, the optimal choice in every round is $u_t = \varepsilon_t$. Assume $T \geq 1$. Then, the restricted dynamic regret is given by

$$\begin{aligned}\mathbb{E}\left[R_T(u_{1:T})\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(\varepsilon_t)\right] \\ &= \sum_{t=1}^{T} \frac{1}{2}\mathbb{E}[x_t^2] + \frac{1}{2}\mathbb{E}[\varepsilon^2] \geq \frac{\sigma^2}{2} T, \qquad (13)\end{aligned}$$

where the expectation is taken with respect to the randomness in the sequence of loss functions and any algorithm $\mathcal{A}$, while the inequality is due to the fact that x_t is independent from ε_t and $\mathbb{E}[\varepsilon_t] = 0$. Now, note that we can upper bound the temporal variability as follows

$$\begin{aligned}V_T &= \sum_{t=1}^{T-1} \max_{x \in \mathcal{V}} |\ell_t(x) - \ell_{t+1}(x)| \\ &= \sum_{t=1}^{T-1} \max_{x \in \mathcal{V}} \left|\frac{1}{2}(x - \varepsilon_t)^2 - \frac{1}{2}(x - \varepsilon_{t+1})^2\right| \\ &= \sum_{t=1}^{T-1} \max_{x \in \mathcal{V}} |x(\varepsilon_{t+1} - \varepsilon_t)| \\ &\leq \sum_{t=1}^{T-1} |\varepsilon_{t+1} - \varepsilon_t| \\ &\leq 2\sigma T \ . \qquad (14)\end{aligned}$$

Observe that if we set $\sigma = C'/2$ for a positive constant C', then we recover the result in Proposition 1 of Besbes et al. [2015] which says that it is impossible to achieve sublinear dynamic regret unless $V_T = o(T)$.

The rest of the proof follows easily from Yang et al. [2016], but we report it here for completeness. We let $\sigma = T^{-\mu}$, with $\mu = (1 - \gamma)/(2 - \gamma)$ and $\mu \in (0, 1/2)$. Then, from Eq. (13) we have that $R(u_{1:T}) \geq T^{1-2\mu}/2$, while from Eq. (14) we have that $T \geq (V_T/2)^{\frac{1}{1-\mu}}$. Therefore, putting things together we have that $R(u_{1:T}) \geq \frac{1}{2}(V_T/2)^{\frac{1-2\mu}{1-\mu}} = CV_T^{\gamma}$. Note that if $\gamma = 1$ then $\mu = 0$ and the regret must be linear in T. Therefore, we let $\gamma < 1$. $\square$

Lower bound for general dynamic regret

Theorem 4.2. *For any deterministic online algorithm $\mathcal{A}$ and any $\tau \geq 0$, there exists a sequence of comparators $\boldsymbol{u}_1, \dots, \boldsymbol{u}_T$ and a sequence of loss function $\ell_1, \dots, \ell_T$ such that $V_T(\ell_1, \dots, \ell_T) = \tau$ and*

$$R_T(\boldsymbol{u}_1, \dots, \boldsymbol{u}_T) \geq V_T \ .$$

Proof. We consider the following instance. Let $d \geq 2$, $\|\cdot\|$ an arbitrary norm on $\mathbb{R}^d$, and $\mathcal{V} = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq D/2\}$. Let T be any non-negative integer and $\tau = \tau' D/2$ for a certain $\tau' \geq 0$. We define the set of loss functions whose temporal variability is equal to τ by

$$C(\tau') = \left\{\ell_1, \dots, \ell_T : \sum_{t=2}^{T} \max_{\boldsymbol{x} \in \mathcal{V}} |\ell_t - \ell_{t-1}| = \tau' D/2\right\},$$

Then, we define the dynamic regret against the class $C(\tau')$ and as $R_T(\boldsymbol{u}_{1:T}, C(\tau')) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}_t)$. For any sequence $\boldsymbol{u}_{1:T}$ such that $\boldsymbol{u}_t \in \mathcal{V}$, we have that

$$R_T(\boldsymbol{u}_{1:T}, C(\tau')) \geq \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \min_{\boldsymbol{u}_{1:T} : \boldsymbol{u}_t \in \mathcal{V}} \sum_{t=1}^{T} \ell_t(\boldsymbol{u}_t) \ .$$

Without loss of generality, we assume there is an integer M which divides T and define $I = T/M$. We define $C'(\tau')$ as a subset of $C(\tau')$, where the loss functions are fixed to 0 for each round successive to the first one in the interval, while the first one in each interval is defined as follows $\ell_i(\boldsymbol{x}) = L\langle \boldsymbol{g}_i, \boldsymbol{x}\rangle$, with $L = \tau'/M$ and $\|\boldsymbol{g}_i\|_\star = 1$ for all i.

$$\begin{aligned}C'(\tau') = \Big\{&\ell_1, \dots, \ell_T : \sum_{i=1}^{M} \min_{\boldsymbol{x} \in \mathcal{V}} \ell_{(i-1)I+1}(\boldsymbol{x}) = \tau' \frac{D}{2}, \\ &\ell_{(i-1)I+2}, \dots, \ell_{iI} = 0, \forall i = 1, \dots, M\Big\} \ .\end{aligned}$$

Note that each sequence of loss functions in $C'(\tau')$ changes at most M times and the total temporal variability does not exceed $\tau' D/2$.

$$V_T = \sum_{t=2}^{T} \max_{\boldsymbol{x} \in \mathcal{V}} |\ell_t(\boldsymbol{x}) - \ell_{t-1}(\boldsymbol{x})| = \sum_{i=1}^{M} \max_{\boldsymbol{x} \in \mathcal{V}} |L\langle \boldsymbol{g}_i, \boldsymbol{x}\rangle|$$

$$= ML\frac{D}{2} = \tau'\frac{D}{2} \ .$$

Therefore $C'(\tau') \subset C(\tau')$, implying

$$R_T(\boldsymbol{u}_{1:T}, C(\tau')) \geq R_T(\boldsymbol{u}_{1:T}, C'(\tau')) \ . \tag{15}$$

Now we can study the dynamic regret on this instance. In particular, there is always a sequence of $\boldsymbol{g}_i$ such that $\langle \boldsymbol{x}_t, \boldsymbol{g}_i \rangle = 0$ given our assumption that $d \geq 2$. Therefore, we apply the same reasoning for lower bound in the static case given in Theorem 6.3 in Campolongo and Orabona [2020]. Formally, we have

$$\begin{aligned}
R_T(\boldsymbol{u}_{1:T}, C'(\tau')) &= \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \min_{\boldsymbol{u}_{1:T}:\boldsymbol{u}_t \in \mathcal{V}} \sum_{t=1}^T \ell_t(\boldsymbol{u}_t) \\
&= -\sum_{i=1}^M \min_{\boldsymbol{u} \in \mathcal{V}} \sum_{t=(i-1)I+1}^{iI} L\langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \\
&= ML \cdot \frac{D}{2} = \tau' \cdot \frac{D}{2} = \tau,
\end{aligned}$$

which leads to the result in Eq. (6). $\square$

B Expert Advice

In the following we show how to apply Algorithm 2 to the scenario of learning with expert advice. Notice that in this case the loss is linear, therefore the implicit and the standard version of *Mirror Descent* coincide. However, it is known in the literature that *Mirror Descent* with a dynamic learning rate cannot be used in this setting. For this reason, we modify the domain of interest: instead of the usual simplex, one can use a clipped version of it. In particular, the clipping can depend on the round t and one can define the clipped simplex $\mathcal{V}_t$ as follows

$$\mathcal{V}_t \triangleq \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, \, x_i \geq \alpha_t/d \, \forall i = 1, \dots, d\} \ . \tag{16}$$

Next, we show how one can modify the argument used in Theorem 5.1 in order to get the same bound.

Lemma B.1. *Let $\mathcal{V}_t$ be defined as in Eq. (16) and $\alpha_t = 1/t$. Let $\boldsymbol{x}_t, \boldsymbol{u}'_t \in \mathcal{V}_t$ and $\psi(\boldsymbol{x}) = x_i \ln x_i$ be the negative entropy. Define $D_t^2 \triangleq \max_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{V}_t} B_\psi(\boldsymbol{x}, \boldsymbol{y})$. Assume $(\eta_t)_{t=1}^T$ is a non-increasing sequence. Then, we have that*

$$\begin{aligned}
&\sum_{t=1}^T \frac{B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_{t+1})}{\eta_t} \\
&\quad \leq \frac{\ln dT}{\eta_T}\left(1 + 2\sum_{t=2}^T \|\boldsymbol{u}'_{t-1} - \boldsymbol{u}'_t\|_1\right) \ .
\end{aligned} \tag{17}$$

Proof. Observe that the Bregman divergence between two points $\boldsymbol{x}, \boldsymbol{y} \in V_t$ with respect to ψ is the *KL-divergence* between $\boldsymbol{x}$ and $\boldsymbol{y}$.

We can decompose the sum on the left in Eq. (17) as follows

$$\begin{aligned}
&\sum_{t=1}^T \frac{B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_{t+1})}{\eta_t} \\
&\quad \leq \frac{B_\psi(\boldsymbol{u}'_1, \boldsymbol{x}_1)}{\eta_1} + \sum_{t=2}^T \left[\frac{B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}_{t-1'}, \boldsymbol{x}_t)}{\eta_{t-1}}\right] \\
&\quad = \frac{B_\psi(\boldsymbol{u}'_1, \boldsymbol{x}_1)}{\eta_1} + \sum_{t=2}^T \left[\frac{B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}'_{t-1}, \boldsymbol{x}_t)}{\eta_t} + \right. \\
&\qquad \left. \frac{B_\psi(\boldsymbol{u}'_{t-1}, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}_{t-1'}, \boldsymbol{x}_t)}{\eta_{t-1}}\right] \\
&\quad \leq \frac{D_1^2}{\eta_1} + \sum_{t=2}^T D_t^2\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + \\
&\qquad \sum_{t=2}^T \left[\frac{B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t)}{\eta_t} - \frac{B_\psi(\boldsymbol{u}'_{t-1}, \boldsymbol{x}_t)}{\eta_t}\right] \\
&\quad \leq \frac{D_T^2}{\eta_T} + \sum_{t=2}^T \frac{1}{\eta_t}\left(B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}'_{t-1}, \boldsymbol{x}_t)\right) \ .
\end{aligned} \tag{18}$$

We can bound the KL-divergence between any two points $\boldsymbol{x}_t, \boldsymbol{u}_t \in V_t$ as follows. Set $\beta_t = \frac{(d-1)}{d}\alpha_t$ and assume without loss of generality that $\boldsymbol{u}_t = [1 - \beta_t, \frac{\alpha_t}{d}, \dots, \frac{\alpha_t}{d}]$

$$\begin{aligned}
\mathrm{KL}(\boldsymbol{u}_t, \boldsymbol{x}_t) &= \left(1 - \beta_t\right) \ln \frac{1 - \beta_t}{x_{t,1}} + \sum_{i=2}^d \frac{\alpha_t}{d} \ln \frac{\alpha_t}{d \cdot x_{t,i}} \\
&\leq \ln \frac{d}{\alpha_t} + \sum_{i=2}^d \frac{\alpha_t}{d} \ln \frac{\alpha_t}{\alpha_t} \\
&= \ln dt \\
&\leq \ln dT,
\end{aligned}$$

where the first inequality derives from the fact that $1 - \beta_t \leq 1$ and $x_{t,i} \geq \alpha_t/d$.

On the other hand, for the second sum in Eq. (18) we have that

$$\begin{aligned}
&B_\psi(\boldsymbol{u}'_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}'_{t-1}, \boldsymbol{x}_t) \\
&\quad = \psi(\boldsymbol{u}'_t) - \psi(\boldsymbol{u}'_{t-1}) - \langle \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u}'_t - \boldsymbol{u}'_{t-1} \rangle \\
&\quad = -B_\psi(\boldsymbol{u}'_{t-1}, \boldsymbol{u}'_t) + \langle \nabla\psi(\boldsymbol{x}_t) - \nabla\psi(\boldsymbol{u}'_t), \boldsymbol{u}'_{t-1} - \boldsymbol{u}'_t \rangle \\
&\quad \leq \|\nabla\psi(\boldsymbol{x}_t) - \nabla\psi(\boldsymbol{u}'_t)\|_\infty \|\boldsymbol{u}'_{t-1} - \boldsymbol{u}'_t\|_1 \\
&\quad \leq \ln dT \cdot \|\boldsymbol{u}'_{t-1} - \boldsymbol{u}'_t\|_1,
\end{aligned}$$

where the last inequality derives from the fact that $\|\nabla\psi(\boldsymbol{x}_t) - \nabla\psi(\boldsymbol{u}'_t)\|_\infty = \max_{i \in [d]} \ln \frac{x_{t,i}}{u'_{t,i}} \leq \ln \frac{d}{\alpha_t}$. $\square$

C Composite losses

We are now going to derive a regret bound on the case of composite losses for the static regret scenario using the algorithm *AdaImplicit* from Campolongo and Orabona [2020].

Theorem C.1. *Let $\mathcal{V} \subset X \subseteq \mathbb{R}^d$ be a non-empty closed convex set. Let $\ell_t(\boldsymbol{x}) = \tilde{\ell}_t(\boldsymbol{x}) + \beta r(\boldsymbol{x})$, where $r : X \to \mathbb{R}$ is a convex function. Let B_ψ be the Bregman divergence with respect to $\psi : X \to \mathbb{R}$. Assume ψ to be 1-strongly convex w.r.t. $\|\cdot\|$ and let $\boldsymbol{g}_t \in \partial\tilde{\ell}_t(\boldsymbol{x}_t)$ for all $t \in [T]$. Let $\lambda_t = 1/\eta_t$. Then,* AdaImplicit *with $\lambda_1 = 0$ and $\lambda_t = \frac{1}{D^2}\sum_{i=1}^{t-1}(\tilde{\ell}_i(\boldsymbol{x}_i) - \tilde{\ell}_i(\boldsymbol{x}_{i+1}) - B_\psi(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i))$ for $t = 2, \dots, T$ incurs the following regret bound*

$$R_T(\boldsymbol{u}) \le \min\Big\{2(\ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T), \\ 2D\sqrt{3\textstyle\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2 + \beta[r(\boldsymbol{x}_1) - r(\boldsymbol{x}_{T+1})]}\Big\}, \tag{19}$$

Proof. First, let $\boldsymbol{g}'_t \in \partial\tilde{\ell}_t(\boldsymbol{x}_{t+1})$. Note that for any $\boldsymbol{u} \in V$ we have the following

$$\begin{aligned}
\eta_t(\ell_t(\boldsymbol{x}_{t+1}) - \ell_t(\boldsymbol{u})) &\le \eta_t\langle \boldsymbol{g}'_t + \beta\nabla r(\boldsymbol{x}_{t+1}), \boldsymbol{x}_{t+1} - \boldsymbol{u}\rangle \\
&= \langle \eta_t \boldsymbol{g}'_t + \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t) \\
&\quad + \eta_t\beta\nabla r(\boldsymbol{x}_{t+1}), \boldsymbol{x}_{t+1} - \boldsymbol{u}\rangle \\
&\quad - \langle\nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{u}\rangle \\
&\le \langle\nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1}\rangle \\
&= B_\psi(\boldsymbol{u}, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}, \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t),
\end{aligned}$$

where the second inequality derives from the optimality condition of the update rule.

Now, let $\delta_t = \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) - \frac{B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)}{\eta_t}$ and $\tilde{\delta}_t = \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{x}_{t+1}) - \frac{B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)}{\eta_t}$. Therefore, after adding $\ell_t(\boldsymbol{x}_t)$ on both sides, taking $\ell_t(\boldsymbol{x}_{t+1})$, dividing both sides by η_t and summing over time we get

$$\begin{aligned}
&\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \\
&\quad \le \sum_{t=1}^T \frac{B_\psi(\boldsymbol{u}, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}, \boldsymbol{x}_{t+1})}{\eta_t} + \sum_{t=1}^T \delta_t \\
&\quad \le \frac{D^2}{\eta_1} + D^2\sum_{t=2}^T\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \\
&\qquad + \sum_{t=1}^T[\tilde{\delta}_t + \beta(r(\boldsymbol{x}_t) - r(\boldsymbol{x}_{t+1}))] \\
&\quad = \frac{D^2}{\eta_T} + \sum_{t=1}^T \tilde{\delta}_t + \beta[r(\boldsymbol{x}_1) - r(\boldsymbol{x}_{T+1})] \ .
\end{aligned}$$

Using $\lambda_t = 1/\eta_t$, we can upper bound the regret as $R_T(\boldsymbol{u}) \le (D^2 + \beta^2)\lambda_{T+1} + \beta[r(\boldsymbol{x}_1) - r(\boldsymbol{x}_{T+1})]$. The rest of the proof follows easily from Theorem 6.2 in Campolongo and Orabona [2020]. □

Compared to Song et al. [2018], we have a regret bound which is adaptive. Furthermore, their bound does not contain the temporal variability V_T, which could potentially lead to constant regret. Next, we are going to show how one can adapt the previous theorem to the dynamic case.

Composite losses in dynamic environments

Theorem 5.4 *Let $\ell_t(\boldsymbol{x}) = \tilde{\ell}_t(\boldsymbol{x}) + \beta r(\boldsymbol{x})$ and $\boldsymbol{g}_t \in \partial\tilde{\ell}_t(\boldsymbol{x}_t)$. Let $\tilde{\delta}_t = \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)$. Given a constant $C \ge 0$, define $\lambda_t \triangleq \frac{1}{\alpha^2}\sum_{i=1}^{t-1}\tilde{\delta}_i$, where $\alpha^2 = D^2 + \gamma C$. Then, under the assumptions of Theorem 5.1 the regret of Algorithm 2 run with $1/\eta_t = \lambda_t$ against any sequence of comparators $\boldsymbol{u}_{1:T}$ whose path-length is given by $C_T = C$ is bounded as*

$$R_T(\boldsymbol{u}_{1:T}) \le \min\Big\{2(\ell_1(\boldsymbol{x}_1) - \ell_{T+1}(\boldsymbol{x}_{T+1}) + V_T), \\ 2\sqrt{(3D^2 + \gamma C_T)\textstyle\sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\star^2 + \beta(r(\boldsymbol{x}_1) - r(\boldsymbol{x}_{T+1}))}\Big\}$$

where $\tilde{\boldsymbol{g}}_t \in \partial\tilde{\ell}_t(\boldsymbol{x}_t)$.

Proof. To prove the stated bound, we can adapt the proof from Theorem C.1. In particular, from the update rule using Eq. (8) we have that

$$\begin{aligned}
&\eta_t(\ell_t(\boldsymbol{x}_{t+1}) - \ell_t(\boldsymbol{u}_t)) \\
&\quad \le B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t),
\end{aligned}$$

where $\boldsymbol{g}'_t \in \partial\ell_t(\boldsymbol{x}_{t+1})$. Following the proof of Theorem 5.2, summing $\ell_t(\boldsymbol{x}_t)$ on both sides and rearranging terms we get

$$\begin{aligned}
R_T(\boldsymbol{u}_{1:T}) &\le \sum_{t=1}^T \lambda_t(B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}_t, \boldsymbol{x}_{t+1})) \\
&\quad + \sum_{t=1}^T\Big[\tilde{\delta}_t + \beta(r(\boldsymbol{x}_t) - r(\boldsymbol{x}_{t+1}))\Big] \\
&\le 2(D^2 + \gamma C_T + \alpha^2)\lambda_{T+1} + r(\boldsymbol{x}_1) - r(\boldsymbol{x}_{T+1}) \ .
\end{aligned}$$

From the last inequality, substituting the value of α and using again Theorem 6.2 in Campolongo and Orabona [2020] yields the desired result. □

C.1 Combiner

Theorem 6.2 *Let $\mathcal{V} \subset \mathbb{R}^d$ be a closed convex set. Furthermore, assume its diameter is bounded by D, $\mathbf{0} \in \mathcal{V}$ and the losses are non-negative. Let $\boldsymbol{g}_t(\boldsymbol{x}) \in \partial\ell_t(\boldsymbol{x})$ and assume that $\max_{\boldsymbol{x}\in\mathcal{V}, t\in[T]}\|\boldsymbol{g}_t(\boldsymbol{x})\|_\star \le L$.*

Then, running Algorithm 4 with the modified losses $\tilde{\ell}_t(\boldsymbol{x}) = \ell_t(\boldsymbol{x})/c$, where $c \geq LD$, the losses ℓ_t are non-negative, $\mathcal{A}$ the strongly-adaptive algorithm from Cutkosky [2020], and $\mathcal{B}$ the greedy algorithm, guarantees

$$R_T(\boldsymbol{u}_{1:T}) = \tilde{\mathcal{O}}(\min(V_T, \sqrt{TD(C_T + D)} + \sqrt{C})),$$

where C is an upper bound to the loss of algorithm $\mathcal{A}$ and $\tilde{O}$ hides poly-logarithmic terms in T.

Proof of Theorem 6.2. First, note that from our assumptions we have that $\ell_t(\boldsymbol{x}_t) \leq \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{0}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \|\boldsymbol{g}_t\|_\star \|\boldsymbol{x}_t\| \leq LD/2$. Therefore, with $c = LD/2$ we have that $\tilde{\ell}_t(\boldsymbol{x}_t) = \ell_t(\boldsymbol{x}_t)/c \leq 1$. Next, let $\tilde{L}_t = \sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{x}_t)$ denote the loss of the experts algorithm after t time-steps. From Theorem 3 in Gaillard et al. [2014], we have that the loss of the master algorithm compared to algorithm $\mathcal{B}$ is bounded as follows

$$\tilde{L}_T \leq \tilde{L}_{T,\mathcal{B}} + 2\ln 2 + 2\ln\left(1 + \frac{1}{e}\sum_{t=1}^T \frac{\eta_t}{\eta_{t+1}} - 1\right) .$$

Following Gaillard et al. [2014], it can be shown that the last term above is order of $\ln\ln T$. Therefore, given a sequence $\boldsymbol{u}_{1:T}$ adding and subtracting $\sum_{t=1}^T \boldsymbol{u}_t$ and multiplying by c on both sides above we get

$$\begin{aligned} R_T(\boldsymbol{u}_{1:T}) &\leq R_{T,\mathcal{B}}(\boldsymbol{u}_{1:T}) + 2c\ln 2 + 2cK'_T \\ &\leq V_T + \ell_1(\boldsymbol{x}_1) - \ell_{T+1}(\boldsymbol{x}_{T+1}) + 2cK_T \\ &= \tilde{\mathcal{O}}(V_T), \end{aligned}$$

where we have used $K'_T = \ln(1 + \frac{1}{e}\ln(T+1))$ and $K_T = \ln 2 + K'_T$.

On the other hand, for algorithm $\mathcal{A}$, applying Corollary 4 from Gaillard et al. [2014] we have that

$$\begin{aligned} \tilde{L}_T &\leq \tilde{L}_{T,\mathcal{A}} + (2 + K_T)\sqrt{1 + \sum_{t=1}^T (\tilde{\ell}_t(\boldsymbol{b}_t) - \tilde{\ell}_t(\boldsymbol{a}_t))^2} \\ &\quad + K_T + 4 \\ &\leq \tilde{L}_{T,\mathcal{A}} + (2 + K_T)\sqrt{\tilde{L}_{T,\mathcal{B}}} + K_T + 4, \end{aligned} \tag{20}$$

where in the last step we used Corollary 1 from Sani et al. [2014]. Therefore, adding and subtracting $\sum_{t=1}^T \boldsymbol{u}_t$ and multiplying by c on both sides above we get

$$\begin{aligned} R_T(\boldsymbol{u}_{1:T}) &\leq R_{T,\mathcal{A}} + c(2 + K_T)\sqrt{\tilde{L}_{T,\mathcal{B}}} + K_T + 4 \\ &= \tilde{\mathcal{O}}(\sqrt{TD(C_T + D)} + \sqrt{C}) . \end{aligned} \tag{21}$$

Putting together Eq. (20) and Eq. (21) we get the result stated in Theorem 6.2.

□

D Doubling Trick

In this section we illustrate how Algorithm 3 exploit the use of a doubling trick in order to adapt to the *observable* quantity C_T. The idea is to run Algorithm 2 in phases and tune the learning rate λ_t appropriately. At the beginning of each phase i, we start monitoring the path length C_i. Once it reaches a certain threshold, we restart the algorithm doubling the threshold. Formally, we introduce a quantity Q_i for phase i and set the learning rate λ_t of the algorithm as $\lambda_t^i = \frac{1}{\beta_i^2}\sum_{s=1}^{t-1}\delta_s$, with $\beta_i^2 = D^2 + \gamma Q_i$. The resulting algorithm is shown in Algorithm 3. We stress that a doubling trick is necessary in this case. Indeed, in order to have an fully adaptive learning rate, we should be able to tune it as a function of two quantities varying over time, namely the path-length observed and the temporal variability of the losses paid by the algorithm. While both are increasing quantities over time, they should appear both at the numerator and denominator of the learning rate λ_t. However, this would result in a non-monotone sequence of learning rates, thus contradicting the assumptions in Theorem 5.1. Also, we would like to point out that to the best of our knowledge there are no existing methods in the literature which tune the learning rates with non-monotone sequences, at least in the *Mirror Descent* case.

We are now going to analyze the regret bound incurred by Algorithm 3. First, we need the following lemma which bounds the number of epochs the algorithm is restarted.

Lemma D.1. *Let t_i be the first time-step of epoch i, with $t_0 = 1$. Suppose Algorithm 3 is run for a total of $N+1$ epochs. Let $C_i = \sum_{t=t_i}^{t_{i+1}-1}\|\boldsymbol{u}_t^\star - \boldsymbol{u}_{t-1}^\star\|$, with $\|\boldsymbol{u}_1^\star - \boldsymbol{u}_0^\star\| \triangleq 0$. Let $C_T = \sum_{i=0}^N C_i$. Then, we have that N satisfies*

$$N \leq \log_2\left(\frac{C_T}{\sqrt{2}D} + 1\right) . \tag{22}$$

Proof. First, recall that $\sum_{i=0}^{N-1} a^i = \frac{a^N - 1}{a-1}$. Now, as usual with doubling trick note that the sum in the first N epochs of the quantity we are monitoring is at most equal to the final sum over all $N+1$ epochs.

Therefore, we have the following

$$\begin{aligned} \sum_{i=0}^{N-1}\sqrt{2}D2^i &\leq \sqrt{2}D(2^N - 1) \\ &\leq \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\| \\ &= C_T, \end{aligned}$$

where $\|\boldsymbol{u}_{t_0} - \boldsymbol{u}_{t_0-1}\| = \|\boldsymbol{u}_1 - \boldsymbol{u}_0\| \triangleq 0$ by definition. Solving for N yields the desired result. □

Next, we report the proof of the regret bound of Algorithm 3 given in the main paper.

Theorem 5.3 *Let $\mathcal{V} \subset \mathbb{R}^d$ be a non-empty closed convex set. Fix a class of strategies Π, where each strategy $\pi \in \Pi$ is such that $\pi = (\pi_1, \dots, \pi_T)$ and $\pi_t : \mathcal{F}^{t-1} \to \mathcal{V}$. Assume Algorithm 3 is run for N epochs. Then, under the assumptions of Theorem 5.1 the regret against any strategy $\pi \in \Pi$ is bounded as*

$$R_T(\pi) \le (2+c) \min\Bigg((\ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T), \sqrt{\left(3D^2\left(\log_2 \tfrac{C_T}{\sqrt{2}D} + 1\right) + \gamma C_T\right) \textstyle\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2} \Bigg),$$

where $c \triangleq \frac{\sqrt{2}}{D+\gamma\sqrt{2}}$ and $C_T = \sum_{t=2}^{T-1} \|\pi_t(\ell_{1:t-1}) - \pi_{t-1}(\ell_{1:t-2})\|$.

Proof. Let $V_i = \sum_{t=t_i+1}^{t_{i+1}-1} \max_{\boldsymbol{x}\in V} |\ell_t(\boldsymbol{x}) - \ell_{t-1}(\boldsymbol{x})|$. Using the result from Theorem 5.2, assuming the knowledge of C_i during each phase i we have that

$$\begin{aligned} R(\boldsymbol{u}^\star_{1:T}) &= \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} R(\boldsymbol{u}^\star_{t_i:t_{i+1}-1}) \\ &\le \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \frac{D^2 + \gamma C_i + \beta_i^2}{\beta_i^2} \min(B_1^i, B_2^i), \end{aligned}$$

where in the last inequality we used Theorem 5.2 with $B_1^i = \ell_{t_i}(\boldsymbol{x}_1) - \ell_{t_{i+1}-1}(\boldsymbol{x}_{t_{i+1}}) + V_i$ and $B_2 = \sqrt{(2D^2+\beta^2)\sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{g}_t\|_\star^2}$.

Note that

$$\begin{aligned} \frac{D^2 + \gamma C_i + \beta_i^2}{\beta_i^2} &= \frac{D^2 + \gamma C_i + D^2 + \gamma Q_i}{D^2 + \gamma Q_i} \\ &= 2 + \gamma \frac{(C_i - Q_i)}{D^2 + \gamma Q_i} \\ &\le 2 + \gamma \frac{\sqrt{2}D}{D^2 + \gamma\sqrt{2}D2^i}, \end{aligned}$$

where the last inequality derives from the fact that the last term in C_i which causes the algorithm to restart is such that $\|\boldsymbol{x} - \boldsymbol{y}\| \le \sqrt{2}D$, $\forall \boldsymbol{x}, \boldsymbol{y} \in V$.

Therefore, we have

$$\begin{aligned} R(\boldsymbol{u}_{1:T}) &\le \sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \left(2 + \gamma \frac{\sqrt{2}}{D + \gamma 2^{i+\frac{1}{2}}}\right) \min(B_1^i, B_2^i) \\ &\le \sum_{i=0}^N (2+c) \min\Bigg\{ \ell_{t_i}(\boldsymbol{x}_{t_i}) - \ell_{t_{i+1}-1}(\boldsymbol{x}_{t_{i+1}}) + V_i, \\ &\qquad \sqrt{(3D^2 + \gamma\sqrt{2}D2^i)\textstyle\sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{g}_t\|_\star^2} \Bigg\} \\ &\le (2+c) \sum_{i=0}^N \min\Bigg\{ \ell_{t_i}(\boldsymbol{x}_{t_i}) - \ell_{t_{i+1}-1}(\boldsymbol{x}_{t_{i+1}}) + V_i, \\ &\qquad \sqrt{(3D^2 + \gamma C_i)\textstyle\sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{g}_t\|_\star^2} \Bigg\} \\ &\le (2+c) \min\Bigg\{ \underbrace{\sum_{i=0}^N (\ell_{t_i}(\boldsymbol{x}_{t_i}) - \ell_{t_{i+1}-1}(\boldsymbol{x}_{t_{i+1}}) + V_i)}_{(a)}, \\ &\qquad \underbrace{\sum_{i=0}^N \sqrt{(3D^2 + \gamma C_i)\textstyle\sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{g}_t\|_\star^2}}_{(b)} \Bigg\}, \end{aligned}$$

where in the the second inequality we used the definition of c. We now analyze (a) and (b) separately.

For (b), using the Cauchy-Schwartz inequality we have that

$$\begin{aligned} &\sum_{i=0}^N \sqrt{(3D^2 + \gamma C_i) \sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{g}_t\|_\star^2} \\ &\le \sqrt{\sum_{i=0}^N (3D^2 + \gamma C_i)} \cdot \sqrt{\sum_{i=0}^N \sum_{t=t_i}^{t_{i+1}-1} \|\boldsymbol{g}_t\|_\star^2} \\ &= \sqrt{3ND^2 + \gamma C_T} \cdot \sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2} \\ &\le \sqrt{\left(3D^2\left(\log_2 \frac{C_T}{\sqrt{2}D} + 1\right) + \gamma C_T\right) \sum_{t=1}^T \|\boldsymbol{g}_t\|_\star^2} \ . \end{aligned}$$

On the other hand, for (a) we have

$$\begin{aligned} &\sum_{i=0}^N (\ell_{t_i}(\boldsymbol{x}_{t_i}) - \ell_{t_{i+1}-1}(\boldsymbol{x}_{t_{i+1}}) + V_i) \\ &\qquad \le \ell_1(\boldsymbol{x}_1) - \ell_T(\boldsymbol{x}_{T+1}) + V_T \ . \end{aligned}$$ □

To summarize, we have a worst-case regret bound for dynamic comparators as follows

$$R(\boldsymbol{u}_{1:T}) = \tilde{\mathcal{O}}\left(\min\left\{V_T, \sqrt{T(1+C_T)}\right\}\right) \ . \tag{23}$$

In light of this last result, compared to Jadbabaie et al. [2015], our upper bound from Theorem 5.3 strictly improves their result when optimistic predictions are not helpful.