# Online Learning with Adversarial Delays

Kent Quanrud* and Daniel Khashabi†

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{quanrud2,khashab2}@illinois.edu

## Abstract

We study the performance of standard online learning algorithms when the feedback is delayed by an adversary. We show that `online-gradient-descent` [1] and `follow-the-perturbed-leader` [2] achieve regret $O(\sqrt{D})$ in the delayed setting, where $D$ is the sum of delays of each round's feedback. This bound collapses to an optimal $O(\sqrt{T})$ bound in the usual setting of no delays (where $D = T$). Our main contribution is to show that standard algorithms for online learning already have simple regret bounds in the most general setting of delayed feedback, making adjustments to the analysis and not to the algorithms themselves. Our results help affirm and clarify the success of recent algorithms in optimization and machine learning that operate in a delayed feedback model.

## 1 Introduction

Consider the following simple game. Let $K$ be a bounded set, such as the unit $\ell_1$ ball or a collection of $n$ experts. Each round $t$, we pick a point $x_t \in K$. An adversary then gives us a cost function $f_t$, and we incur the *loss* $\ell_t = f_t(x_t)$. After $T$ rounds, our *total loss* is the sum $L_T = \sum_{t=1}^T \ell_t$, which we want to minimize.

We cannot hope to beat the adversary, so to speak, when the adversary picks the cost function *after* we select our point. There is margin for optimism, however, if rather than evaluate our total loss in absolute terms, we compare our strategy to the best fixed point in hindsight. The *regret* of a strategy $x_1, \ldots, x_T \in K$ is the additive difference $R(T) = \sum_{t=1}^T f_t(x_t) - \arg\min_{x \in K} \sum_{t=1}^T f_t(x)$.

Surprisingly, one can obtain positive results in terms of regret. Kalai and Vempala showed that a simple and randomized follow-the-leader type algorithm achieves $R(T) = O(\sqrt{T})$ in expectation for linear cost functions [2] (here, the big-$O$ notation assumes that the diameter of $K$ and the $f_t$'s are bounded by constants). If $K$ is convex, then even if the cost vectors are more generally convex cost functions (where we incur losses of the form $\ell_t = f_t(x_t)$, with $f_t$ a convex function), Zinkevich showed that gradient descent achieves regret $R(T) = O(\sqrt{T})$ [1]. There is a large body of theoretical literature about this setting, called *online learning* (see for example the surveys by Blum [3], Shalev-Shwartz [4], and Hazan [5]).

Online learning is general enough to be applied to a diverse family of problems. For example, Kalai and Vempala's algorithm can be applied to online combinatorial problems such as shortest paths [6], decision trees [7], and data structures [8, 2]. In addition to basic machine learning problems with convex loss functions, Zinkevich considers applications to industrial optimization, where the

*`http://illinois.edu/~quanrud2`. Supported in part by NSF grants CCF-1217462, CCF-1319376, CCF-1421231, CCF-1526799.

†`http://illinois.edu/~khashab2`. Supported in part by a grant from Google.

value of goods is not known until after the goods are produced. Other examples of applications of online learning include universal portfolios in finance [9] and online topic-ranking for multi-labeled documents [10].

The standard setting assumes that the cost vector $f_t$ (or more generally, the feedback) is given to and processed by the player before making the next decision in round $t+1$. Philosophically, this is not how decisions are made in real life: we rush through many different things at the same time with no pause for careful consideration, and we may not realize our mistakes for a while. Unsurprisingly, the assumption of immediate feedback is too restrictive for many real applications. In online advertising, online learning algorithms try to predict and serve ads that optimize for clicks [11]. The algorithm learns by observing whether or not an ad is clicked, but in production systems, a massive number of ads are served between the moment an ad is displayed to a user and the moment the user has decided to either click or ignore that ad. In military applications, online learning algorithms are used by radio jammers to identify efficient jamming strategies [12]. After a jammer attempts to disrupt a packet between a transmitter and a receiver, it does not know if the jamming attempt succeeded until an acknowledgement packet is sent by the receiver. In cloud computing, online learning helps devise efficient resource allocation strategies, such as finding the right mix of cheaper (and inconsistent) spot instances and more reliable (and expensive) on-demand instances when renting computers for batch jobs [13]. The learning algorithm does not know how well an allocation strategy worked for a batch job until the batch job has ended, by which time many more batch jobs have already been launched. In finance, online learning algorithms managing portfolios are subject to information and transaction delays from the market, and financial firms invest heavily to minimize these delays.

One strategy to handle delayed feedback is to pool independent copies of a fixed learning algorithm, each of which acts as an undelayed learner over a subsequence of the rounds. Each round is delegated to a single instance from the pool of learners, and the learner is required to wait for and process its feedback before rejoining the pool. If there are no learners available, a new copy is instantiated and added to the pool. The size of the pool is proportional to the maximum number of outstanding delays at any point of decision, and the overall regret is bounded by the sum of regrets of the individual learners. This approach is analyzed for constant delays by Weinberger and Ordentlich [14], and a more sophisticated analysis is given by Joulani *et al.* [15]. If $\alpha$ is the expected maximum number of outstanding feedbacks, then Joulani *et al.* obtain a regret bound on the order of $O(\sqrt{\alpha T})$ (in expectation) for the setting considered here. The blackbox nature of this approach begets simultaneous bounds for other settings such as partial information and stochastic rewards. Although maintaining copies of learners in proportion to the delay may be prohibitively resource intensive, Joulani *et al.* provide a more efficient variant for the stochastic bandit problem, a setting not considered here.

Another line of research is dedicated to scaling gradient descent type algorithms to distributed settings, where asynchronous processors naturally introduce delays in the learning framework. A classic reference in this area is the book of Bertsekas and Tsitskilis [16]. If the data is very sparse, so that input instances and their gradients are somewhat orthogonal, then intuitively we can apply gradients out of order without significant interference across rounds. This idea is explored by Recht *et al.* [17], who analyze and test parallel algorithm on a restricted class of strongly convex loss functions, and by Duchi *et al.* [18] and McMahan and Streeter [19], who design and analyze distributed variants of adaptive gradient descent [20]. Perhaps the most closely related work in this area is by Langford *et al.*, who study the `online-gradient-descent` algorithm of Zinkevich when the delays are bounded by a constant number of rounds [21]. Research in this area has largely moved on from the simplistic models considered here; see [22, 23, 24] for more recent developments.

The impact of delayed feedback in learning algorithms is also explored by Riabko [25] under the framework of "weak teachers".

For the sake of concreteness, we establish the following notation for the delayed setting. For each round $t$, let $d_t \in \mathbb{Z}^+$ be a non-negative integer *delay*. The feedback from round $t$ is delivered at the end of round $t + d_t - 1$, and can be used in round $t + d_t$. In the standard setting with no delays, $d_t = 1$ for all $t$. For each round $t$, let $\mathcal{F}_t = \{u \in [T] : u + d_u - 1 = t\}$ be the set of rounds whose feedback appears at the end of round $t$. We let $D = \sum_{t=1}^{T} d_t$ denote the sum of all delays; in the standard setting with no delays, we have $D = T$.

In this paper, we investigate the implications of delayed feedback when the delays are *adversarial* (i.e., *arbitrary*), with no assumptions or restrictions made on the adversary. Rather than design new algorithms that may generate a more involved analysis, we study the performance of the classical algorithms `online-gradient-descent` and `follow-the-perturbed-leader`, essentially unmodified, when the feedback is delayed. In the delayed setting, we prove that both algorithms have a simple regret bound of $O(\sqrt{D})$. These bounds collapse to match the well-known $O(\sqrt{T})$ regret bounds if there are no delays (i.e., where $D = T$).

**Paper organization** In Section 2, we analyze the `online-gradient-descent` algorithm in the delayed setting, giving upper bounds on the regret as a function of the sum of delays $D$. In Section 3, we analyze the `follow-the-perturbed-leader` in the delayed setting and derive a regret bound in terms of $D$. Due to space constraints, extensions to `online-mirror-descent` and `follow-the-lazy-leader` are deferred to the appendix. We conclude and propose future directions in Section 4.

## 2 Delayed gradient descent

**Convex optimization** In online convex optimization, the input domain $K$ is convex, and each cost function $f_t$ is convex. For this setting, Zinkevich proposed a simple online algorithm, called `online-gradient-descent`, designed as follows [1]. The first point, $x_1$, is picked in $K$ arbitrarily. After picking the $t$th point $x_t$, `online-gradient-descent` computes the gradient $\nabla f_t|_{x_t}$ of the loss function at $x_t$, and chooses $x_{t+1} = \pi_K(x_t - \eta \nabla f_t|_{x_t})$ in the subsequent round, for some parameter $\eta \in \mathbb{R}_{>0}$. Here, $\pi_K$ is the projection that maps a point $x'$ to its nearest point in $K$ (discussed further below). Zinkevich showed that, assuming the Euclidean diameter of $K$ and the Euclidean lengths of all gradients $\nabla f_t|_x$ are bounded by constants, `online-gradient-descent` has an optimal regret bound of $O(\sqrt{T})$.

**Delayed gradient descent** In the delayed setting, the loss function $f_t$ is not necessarily given by the adversary before we pick the next point $x_{t+1}$ (or even at all). The natural generalization of `online-gradient-descent` to this setting is to process the convex loss functions and apply their gradients the moment they are delivered. That is, we update

$$x'_{t+1} = x_t - \eta \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s},$$

for some fixed parameter $\eta$, and then project $x_{t+1} = \pi_K(x'_{t+1})$ back into $K$ to choose our $(t+1)$th point. In the setting of Zinkevich, we have $\mathcal{F}_t = \{t\}$ for each $t$, and this algorithm is exactly `online-gradient-descent`. Note that a gradient $\nabla f_s|_{x_s}$ does not need to be timestamped by the round $s$ from which it originates, which is required by the pooling strategies of Weinberger and Ordentlich [14] and Joulani *et al.* [15] in order to return the feedback to the appropriate learner.

**Theorem 2.1.** *Let $K$ be a convex set with diameter 1, let $f_1, \ldots, f_T$ be convex functions over $K$ with $\|\nabla f_t|_x\|_2 \leq L$ for all $x \in K$ and $t \in [T]$, and let $\eta \in \mathbb{R}$ be a fixed parameter. In the presence of adversarial delays,* `online-gradient-descent` *selects points $x_1, \ldots, x_T \in K$ such that for all $y \in K$,*

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(y) = O\left(\frac{1}{\eta} + \eta L^2 (T + D)\right),$$

*where $D$ denotes the sum of delays over all rounds $t \in [T]$.*

For $\eta = 1/L\sqrt{T+D}$, Theorem 2.1 implies a regret bound of $O(L\sqrt{D+T}) = O(L\sqrt{D})$. This choice of $\eta$ requires prior knowledge of the final sum $D$. When this sum is not known, one can calculate $D$ on the fly: if there are $\delta$ outstanding (undelivered) cost functions at a round $t$, then $D$ increases by exactly $\delta$. Obviously, $\delta \leq T$ and $T \leq D$, so $D$ at most doubles. We can therefore employ the "doubling trick" of Auer *et al.* [26] to dynamically adjust $\eta$ as $D$ grows.

In the undelayed setting analyzed by Zinkevich, we have $D = T$, and the regret bound of Theorem 2.1 matches that obtained by Zinkevich. If each delay $d_t$ is bounded by some fixed value $\tau$, Theorem 2.1 implies a regret bound of $O(L\sqrt{\tau T})$ that matches that of Langford *et al.* [21]. In both of these special cases, the regret bound is known to be tight.

Before proving Theorem 2.1, we review basic definitions and facts on convexity. A function $f : K \to \mathbb{R}$ is *convex* if

$$f((1-\alpha)x + \alpha y) \leq (1-\alpha)f(x) + \alpha f(y) \qquad \forall x, y \in K, \alpha \in [0,1].$$

If $f$ is differentiable, then $f$ is convex iff

$$f(x) + \nabla f|_x \cdot (y - x) \leq f(y) \qquad \forall x, y \in K. \tag{1}$$

For $f$ convex but not necessarily differentiable, a *subgradient* of $f$ at $x$ is any vector that can replace $\nabla f|_x$ in equation (1). The (possible empty) set of gradients of $f$ at $x$ is denoted by $\partial f(x)$.

The gradient descent may occasionally update along a gradient that takes us out of the constrained domain $K$. If $K$ is convex, then we can simply project the point back into $K$.

**Lemma 2.2.** *Let $K$ be a closed convex set in a normed linear space $X$ and $x \in X$ a point, and let $x' \in K$ be the closest point in $K$ to $x$. Then, for any point $y \in K$,*

$$\|x - y\|_2 \leq \|x' - y\|_2.$$

We let $\pi_K$ denote the map taking a point $x$ to its closest point in the convex set $K$.

*Proof of Theorem 2.1.* Let $y = \arg\min_{x \in K}(f_1(x) + \cdots + f_T(x))$ be the best point in hindsight at the end of all $T$ rounds. For $t \in [T]$, by convexity of $f_t$, we have,

$$f_t(y) \geq f_t(x_t) + \nabla f_t|_{x_t} \cdot (y - x_t).$$

Fix $t \in [T]$, and consider the distance between $x_{t+1}$ and $y$. By Lemma 2.2, we know that $\|x_{t+1} - y\|_2 \leq \left\|x'_{t+1} - y\right\|_2$, where $x'_{t+1} = x_t - \eta \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s}$.

We split the sum of gradients applied in a single round and consider them one by one. For each $s \in \mathcal{F}_t$, let $\mathcal{F}_{t,s} = \{r \in \mathcal{F}_t : r < s\}$, and let $x_{t,s} = x_t - \eta \sum_{r \in \mathcal{F}_{t,s}} \nabla f_r|_{x_r}$. Suppose $\mathcal{F}_t$ is nonempty, and fix $s' = \max \mathcal{F}_t$ to be the last index in $\mathcal{F}_t$. By Lemma 2.2, we have,

$$\begin{aligned}
\|x_{t+1} - y\|_2^2 \leq \left\|x'_{t+1} - y\right\|_2^2 &= \left\|x_{t,s'} - \eta \nabla f_{s'}|_{x_{s'}} - y\right\|_2^2 \\
&= \|x_{t,s'} - y\|_2^2 - 2\eta\big(\nabla f_{s'}|_{x_{s'}} \cdot (x_{t,s'} - y)\big) + \eta^2 \left\|\nabla f_{s'}|_{x_{s'}}\right\|_2^2.
\end{aligned}$$

Repeatedly unrolling the first term in this fashion gives

$$\|x_{t+1} - y\|_2^2 \leq \|x_t - y\|_2^2 - 2\eta \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s} \cdot (x_{t,s} - y) + \eta^2 \sum_{s \in \mathcal{F}_t} \|\nabla f_s|_{x_s}\|_2^2.$$

For each $s \in \mathcal{F}_t$, by convexity of $f$, we have,

$$\begin{aligned}
-\nabla f_s|_{x_s} \cdot (x_{t,s} - y) = \nabla f_s|_{x_s} \cdot (y - x_{t,s}) &= \nabla f_s|_{x_s} \cdot (y - x_s) + \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}) \\
&\leq f_s(y) - f_s(x_s) + \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}).
\end{aligned}$$

By assumption, we also have $\|\nabla f_s|_{x_s}\|_2 \leq L$ for each $s \in \mathcal{F}_t$. With respect to the distance between $x_{t+1}$ and $y$, this gives,

$$\|x_{t+1} - y\|_2^2 \leq \|x_t - y\|_2^2 + 2\eta \sum_{s \in \mathcal{F}_t} (f_s(y) - f_s(x_s) + \nabla f_s|_{x_s} \cdot (x_s - x_{t,s})) + \eta^2 \cdot |\mathcal{F}_t| \cdot L^2.$$

Solving this inequality for the regret terms $\sum_{s \in \mathcal{F}_t} f_s(x_s) - f_s(y)$ and taking the sum of inequalities over all rounds $t \in [T]$, we have,

$$\begin{aligned}
\sum_{t=1}^{T}(f_t(x_t) - f_t(y)) &= \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} f_s(x_s) - f_s(y) \\
&\leq \frac{1}{2\eta} \cdot \sum_{t=1}^{T} \left( \|x_t - y\|_2^2 - \|x_{t+1} - y\|_2^2 + 2\eta \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}) + \eta^2 \cdot |\mathcal{F}_t| \cdot L^2 \right) \\
&= \frac{1}{2\eta} \left( \sum_{t=1}^{T} \|x_t - y\|_2^2 - \|x_{t+1} - y\|_2^2 \right) + \frac{\eta}{2} T L^2 + \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}) \\
&\leq \frac{1}{2\eta} + \frac{\eta}{2} T L^2 + \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}).
\end{aligned} \tag{2}$$

The first two terms are familiar from the standard analysis of `online-gradient-descent`. It remains to analyze the last sum, which we call the *delay term*.

Each summand $\nabla f_s|_{x_s} \cdot (x_s - x_{t,s})$ in the delay term contributes loss proportional to the distance between the point $x_s$ when the gradient $\nabla f_s|_{x_s}$ is generated and the point $x_{t,s}$ when the gradient is applied. This distance is created by the other gradients that are applied in between, and the number of such in-between gradients are intimately tied to the total delay, as follows. By Cauchy-Schwartz, the delay term is bounded above by

$$\sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}) \leq \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \|\nabla f_s|_{x_s}\|_2 \|x_s - x_{t,s}\|_2 \leq L \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \|x_s - x_{t,s}\|_2. \quad (3)$$

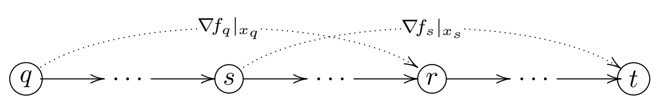
Consider a single term $\|x_s - x_{t,s}\|_2$ for fixed $t \in [T]$ and $s \in \mathcal{F}_t$. Intuitively, the difference $x_{t,s} - x_s$ is roughly the sum of gradients received between round $s$ and when we apply the gradient from round $s$ in round $t$. More precisely, by applying the triangle inequality and Lemma 2.2, we have,

$$\|x_{t,s} - x_s\|_2 \leq \|x_{t,s} - x_t\|_2 + \|x_t - x_s\|_2 \leq \|x_{t,s} - x_t\|_2 + \|x'_t - x_s\|_2.$$

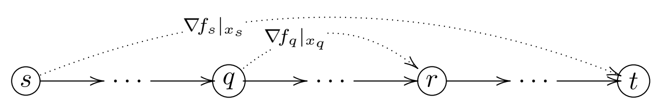
For the same reason, we have $\|x'_t - x_s\|_2 \leq \|x'_t - x_{t-1}\|_2 + \left\|x'_{t-1} - x_s\right\|_2$, and unrolling in this fashion, we have,

$$\begin{aligned} \|x_{t,s} - x_s\|_2 &\leq \|x_{t,s} - x_t\|_2 + \sum_{r=s}^{t-1} \left\|x'_{r+1} - x_r\right\|_2 \leq \eta \sum_{p \in \mathcal{F}_{t,s}} \left\|\nabla f_p|_{x_p}\right\|_2 + \eta \sum_{r=s}^{t-1} \sum_{q \in \mathcal{F}_r} \left\|\nabla f_q|_{x_q}\right\|_2 \\ &\leq \eta \cdot L \cdot \left( |\mathcal{F}_{t,s}| + \sum_{r=s}^{t-1} |\mathcal{F}_r| \right). \end{aligned} \quad (4)$$

After substituting equation (4) into equation (3), it remains to bound the sum $\sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} (|\mathcal{F}_{t,s}| + \sum_{r=s}^{t-1} |\mathcal{F}_r|)$. Consider a single term $|\mathcal{F}_{t,s}| + \sum_{r=s}^{t-1} |\mathcal{F}_r|$ in the sum. This quantity counts, for a gradient $\nabla f_s|_{x_s}$ from round $s$ delivered just before round $t \geq s$, the number of other gradients that are applied while $\nabla f_s|_{x_s}$ is withheld. Fix two rounds $s$ and $t$, and consider an intermediate round $r \in \{s, \ldots, t\}$. If $r < t$ then fix $q \in \mathcal{F}_r$, and if $r = t$ then fix $q \in \mathcal{F}_{t,s}$. The feedback from round $q$ is applied in a round $r$ between round $s$ and round $t$. We divide our analysis into two scenarios. In one case, $q \leq s$, and the gradient from round $q$ appears only after $s$, as in the following diagram.




In the other case, $q > s$, as in the following diagram.




For each round $u$, let $d_u$ denote the number of rounds the gradient feedback is delayed (so $u \in \mathcal{F}_{u+d_u}$). There are at most $d_s$ instances of the latter case, since $q$ must lie in $s+1, \ldots, t$. The first case can be charged to $d_q$. To bound the first case, observe that for fixed $q$, the number of indices $s$ such that $q < s \leq d_q + q \leq d_s + s$ is at most $d_q$. That is, all instances of the second case for a fixed $q$ can be charged to $d_q$. Between the two cases, we have $\sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} (|\mathcal{F}_{t,s}| + \sum_{r=s}^{t-1} |\mathcal{F}_r|) \leq 2 \sum_{t=1}^{T} d_t$, and the delay term is bounded by

$$\sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \nabla f_s|_{x_s} \cdot (x_s - x_{t,s}) \leq 2\eta \cdot L^2 \sum_{t=1}^{T} d_t.$$

With respect to the overall regret, this gives,

$$\sum_{t=1}^{T} (f(x_t) - f(y)) \leq \frac{1}{2\eta} + \eta \cdot L^2 \left( \frac{T}{2} + 2 \sum_{t=1}^{T} d_t \right) = O\left( \frac{1}{\eta} + \eta L^2 D \right),$$

as desired. ■

*Remark* 2.3. The delay term $\sum_{t=1}^{T}\sum_{s\in\mathcal{F}_t}\nabla f_s|_{x_s}\cdot(x_s - x_{t,s})$ is a natural point of entry for a sharper analysis based on strong sparseness assumptions. The distance $x_s - x_{t,s}$ is measured by its projection against the gradient $\nabla f_s|_{x_s}$, and the preceding proof assumes the worst case and bounds the dot product with the Cauchy-Schwartz inequality. If, for example, we assume that gradients are pairwise orthogonal and analyze `online-gradient-descent` in the unconstrained setting, then the dot product $\nabla f_s|_{x_s}\cdot(x_s - x_{t,s})$ is 0 and the delay term vanishes altogether.

## 3 Delaying the Perturbed Leader

**Discrete online linear optimization** In discrete online linear optimization, the input domain $K \subset \mathbb{R}^n$ is a (possibly discrete) set with bounded diameter, and each cost function $f_t$ is of the form $f_t(x) = c_t \cdot x$ for a bounded-length cost vector $c_t$. The previous algorithm `online-gradient-descent` does not apply here because $K$ is not convex.

A natural algorithm for this problem is `follow-the-leader`. Each round $t$, let $y_t = \arg\min_{x\in K} x\cdot(c_1 + \cdots + c_t)$ be the optimum choice over the first $t$ cost vectors. The algorithm picking $y_t$ in round $t$ is called `be-the-leader`, and can be shown to have zero regret. Of course, `be-the-leader` is infeasible since the cost vector $c_t$ is revealed *after* picking $y_t$. `follow-the-leader` tries the next best thing, picking $y_{t-1}$ in round $t$. Unfortunately, this strategy can have linear regret, largely because it is a deterministic algorithm that can be manipulated by an adversary.

Kalai and Vempala [2] gave a simple and elegant correction called `follow-the-perturbed-leader`. Let $\epsilon > 0$ be a parameter to be fixed later, and let $Q_\epsilon = [0, 1/\epsilon]^n$ be the cube of length $1/\epsilon$. Each round $t$, `follow-the-perturbed-leader` randomly picks a vector $c_0 \in Q_\epsilon$ by the uniform distribution, and then selects $x_t = \arg\min_{x\in K} x\cdot(c_0 + \cdots + c_{t-1})$ to optimize over the previous costs plus the random perturbation $c_0$. With the diameter of $K$ and the lengths $\|c_t\|$ of each cost vector held constant, Kalai and Vempala showed that `follow-the-perturbed-leader` has regret $O(\sqrt{T})$ in expectation.

**Following the delayed and perturbed leader** More generally, `follow-the-perturbed-leader` optimizes over all information available to the algorithm, plus some additional noise to smoothen the worst-case analysis. If the cost vectors are delayed, we naturally interpret `follow-the-perturbed-leader` to optimize over all cost vectors $c_t$ delivered in time for round $t$ when picking its point $x_t$. That is, the $t$th leader becomes the best choice with respect to all cost vectors delivered in the first $t$ rounds:

$$y_t^{\mathrm{d}} = \arg\min_{x\in K}\sum_{s=1}^{t}\sum_{r\in\mathcal{F}_s} c_r\cdot x$$

(we use the superscript $\mathrm{d}$ to emphasize the delayed setting). The $t$th perturbed leader optimizes over all cost vectors delivered through the first $t$ rounds in addition to the random perturbation $c_0 \in Q_\epsilon$:

$$\tilde{y}_t^{\mathrm{d}} = \arg\min_{x\in K}\left(c_0\cdot x + \sum_{s=1}^{t}\sum_{r\in\mathcal{F}_s} c_r\cdot x\right).$$

In the delayed setting, `follow-the-perturbed-leader` chooses $x_t = \tilde{y}_{t-1}^{\mathrm{d}}$ in round $t$. We claim that `follow-the-perturbed-leader` has a direct and simple regret bound in terms of the sum of delays $D$, that collapses to Kalai and Vempala's $O(\sqrt{T})$ regret bound in the undelayed setting.

**Theorem 3.1.** *Let $K \subseteq \mathbb{R}^n$ be a set with $L_1$-diameter $\leq 1$, $c_1, \ldots, c_T \in \mathbb{R}^n$ with $\|c_t\|_1 \leq 1$ for all $t$, and $\eta > 0$. In the presence of adversarial delays,* `follow-the-perturbed-leader` *picks points $x_1, \ldots, x_T \in K$ such that for all $y \in K$,*

$$\sum_{t=1}^{T}\mathbf{E}[c_t\cdot x_t] \leq \sum_{t=1}^{T} c_t\cdot y + O\big(\epsilon^{-1} + \epsilon D\big).$$

For $\epsilon = 1/\sqrt{D}$, Theorem 3.1 implies a regret bound of $O(\sqrt{D})$. When $D$ is not known *a priori*, the doubling trick can be used to adjust $\epsilon$ dynamically (see the discussion following Theorem 2.1).

To analyze `follow-the-perturbed-leader` in the presence of delays, we introduce the notion of a prophet, who is a sort of omniscient leader who sees the feedback immediately. Formally, the $t$th *prophet* is the best point with respect to all the cost vectors over the first $t$ rounds:

$$z_t = \arg\min_{x \in K} (c_1 + \cdots + c_t) \cdot x.$$

The $t$th *perturbed prophet* is the best point with respect to all the cost vectors over the first $t$ rounds, in addition to a perturbation $c_0 \in Q_\epsilon$:

$$\tilde{z}_t = \arg\min_{x \in K} (c_0 + c_1 + \cdots + c_t) \cdot x. \tag{5}$$

The prophets and perturbed prophets behave exactly as the leaders and perturbed leaders in the setting of Kalai and Vempala with no delays. In particular, we can apply the regret bound of Kalai and Vempala to the (infeasible) strategy of following the perturbed prophet.

**Lemma 3.2** ([2]). *Let $K \subseteq \mathbb{R}^n$ be a set with $L_1$-diameter $\leq 1$, let $c_1, \ldots, c_T \in \mathbb{R}^n$ be cost vectors bounded by $\|c_t\|_1 \leq 1$ for all $t$, and let $\epsilon > 0$. If $\tilde{z}_1, \ldots, \tilde{z}_{T-1} \in K$ are chosen per equation (5), then $\sum_{t=1}^T \mathbf{E}[c_t \cdot \tilde{z}_{i-1}] \leq \sum_{t=1}^T c_t \cdot y + O(\epsilon^{-1} + \epsilon T)$. for all $y \in K$.*

The analysis by Kalai and Vempala observes that when there are no delays, two consecutive perturbed leaders $\tilde{y}_t$ and $\tilde{y}_{t+1}$ are distributed similarly over the random noise [2, Lemma 3.2]. Instead, we will show that $\tilde{y}_t^{\mathrm{d}}$ and $\tilde{z}_t$ are distributed in proportion to delays. We first require a technical lemma that is implicit in [2].

**Lemma 3.3.** *Let $K$ be a set with $L_1$-diameter $\leq 1$, and let $u, v \in \mathbb{R}^n$ be vectors. Let $y, z \in \mathbb{R}^n$ be random vectors defined by $y = \arg\min_{y \in K}(q+u) \cdot y$ and $z = \arg\min_{z \in K}(q+v) \cdot z$, where $q$ is chosen uniformly at random from $Q = \prod_{i=1}^n [0, r]$, for some fixed length $r > 0$. Then, for any vector $c$,*

$$\mathbf{E}[c \cdot z] - \mathbf{E}[c \cdot y] \leq \frac{\|v-u\|_1 \|c\|_\infty}{r}.$$

*Proof.* Let $Q' = v + Q$ and $Q'' = u + Q$, and write $y = \arg\min_{y \in K} q'' \cdot y$ and $z = \arg\min_{z \in K} q' \cdot z$, where $q' \in Q'$ and $q'' \in Q''$ are chosen uniformly at random. Then

$$\mathbf{E}[c \cdot z] - \mathbf{E}[c \cdot y] = \mathbf{E}_{q'' \in Q''}[c \cdot z] - \mathbf{E}_{q' \in Q'}[c \cdot y].$$

Subtracting $\mathbf{P}[q' \in Q' \cap Q'']\mathbf{E}_{q' \in Q' \cap Q''}[c \cdot z]$ from both terms on the right, we have

$$\begin{aligned}&\mathbf{E}_{q'' \in Q''}[c \cdot z] - \mathbf{E}_{q' \in Q'}[c \cdot y] \\ &\quad = \mathbf{P}[q'' \in Q'' \setminus Q'] \cdot \mathbf{E}_{q'' \in Q'' \setminus Q'}[c \cdot z] - \mathbf{P}[q' \in Q' \setminus Q''] \cdot \mathbf{E}_{q' \in Q' \setminus Q''}[c \cdot y]\end{aligned}$$

By symmetry, $\mathbf{P}[q'' \in Q'' \setminus Q'] = \mathbf{P}[q' \in Q' \setminus Q'']$, and we have,

$$\mathbf{E}[c \cdot z] - \mathbf{E}[c \cdot y] \leq (\mathbf{P}[q'' \in Q'' \setminus Q'])\mathbf{E}_{q'' \in Q'' \setminus Q', q' \in Q' \setminus Q''}[c \cdot (z-y)].$$

By assumption, $K$ has $L_1$-diameter $\leq 1$, so $\|y-z\|_1 \leq 1$, and by Hölder's inequality, we have,

$$\mathbf{E}[c \cdot z] - \mathbf{E}[c \cdot y] \leq \mathbf{P}[q'' \in Q'' \setminus Q'] \|c\|_\infty.$$

It remains to bound $\mathbf{P}[q'' \in Q'' \setminus Q'] = \mathbf{P}[q' \in Q' \setminus Q'']$. If $\|v-u\|_1 \leq r$, we have,

$$\operatorname{vol}(Q' \cap Q'') = \prod_{i=1}^n (r - |v_i - u_i|) = \operatorname{vol}(Q') \prod_{i=1}^n \left(1 - \frac{|(v_i - u_i)|}{r}\right) \geq \operatorname{vol}(Q')\left(1 - \frac{\|v-u\|_1}{r}\right).$$

Otherwise, if $\|u-v\|_1 > r$, then $\operatorname{vol}(Q' \cap Q'') = 0 \geq \operatorname{vol}(Q')(1 - \|v-u\|_1/r)$. In either case, we have,

$$\mathbf{P}[q' \in Q' \setminus Q''] = \frac{\operatorname{vol}(Q' \cap Q'')}{\operatorname{vol}(Q')} \leq 1 - \frac{\operatorname{vol}(Q' \cap Q'')}{\operatorname{vol}(Q')} \leq \frac{\|v-u\|_1}{r},$$

and the claim follows. ∎

Lemma 3.3 could also have been proven geometrically in similar fashion to Kalai and Vempala.

**Lemma 3.4.** $\sum_{t=1}^{T} \mathbf{E}[c_t \cdot \tilde{z}_{t-1}] - \mathbf{E}\left[c_t \cdot \tilde{y}^{\mathrm{d}}_{t-1}\right] \leq \epsilon D$*, where $D$ is the sum of delays of all cost vectors.*

*Proof.* Let $u_t = \sum_{s=1}^{t} c_t$ be the sum of all costs through the first $t$ rounds, and $v_t = \sum_{s:s+d_s \leq t} c_t$ be the sum of cost vectors actually delivered through the first $t$ rounds. Then the perturbed prophet $\tilde{z}_{t-1}$ optimizes over $c_0 + u_{t-1}$ and $\tilde{y}^{\mathrm{d}}_{t-1}$ optimizes over $c_0 + v_{t-1}$. By Lemma 3.3, for each $t$, we have

$$\mathbf{E}_{c_0 \sim Q_\epsilon}[c_t \cdot \tilde{z}_{t-1}] - \mathbf{E}_{c_0 \sim Q_\epsilon}\left[c_t \cdot \tilde{y}^{\mathrm{d}}_{t-1}\right] \leq \epsilon \cdot \|u_{t-1} - v_{t-1}\|_1 \|c_t\|_\infty \leq \epsilon \cdot |\{s < t : s + d_s \geq t\}|$$

Summed over all $T$ rounds, we have,

$$\sum_{t=1}^{T} \mathbf{E}_{c_0}[c_t \cdot \tilde{z}_t] - \mathbf{E}_{c_0}\left[c_t \cdot \tilde{y}^{\mathrm{d}}_t\right] \leq \epsilon \sum_{t=1}^{T} |\{s < t : s + d_s \geq t\}|.$$

The sum $\sum_{t=1}^{T} |\{s < t : s + d_s \geq t\}|$ charges each cost vector $c_s$ once for every round it is delayed, and therefore equals $D$. Thus, $\sum_{t=1}^{T} \mathbf{E}_{c_0}[c_t \cdot \tilde{z}_t] - \mathbf{E}_{c_0}\left[c_t \cdot \tilde{y}^{\mathrm{d}}_t\right] \leq \epsilon D$, as desired. ■

Now we complete the proof of Theorem 3.1.

*Proof of Theorem 3.1.* By Lemma 3.4 and Lemma 3.2, we have,

$$\sum_{t=1}^{T} \mathbf{E}\left[c_t \cdot \tilde{y}^{\mathrm{d}}_{t-1}\right] \leq \sum_{t=1}^{T} \mathbf{E}[c_t \cdot \tilde{z}_{t-1}] + \epsilon D \leq \operatorname*{arg\,min}_{x \in K} \sum_{t=1}^{T} \mathbf{E}[c_t \cdot x] + O(\epsilon^{-1} + \epsilon D),$$

as desired. ■

# 4 Conclusion

We prove $O(\sqrt{D})$ regret bounds for `online-gradient-descent` and `follow-the-perturbed-leader` in the delayed setting, directly extending the $O(\sqrt{T})$ regret bounds known in the undelayed setting. More importantly, by deriving a simple bound as a function of the delays, without any restriction on the delays, we establish a simple and intuitive model for measuring delayed learning. This work suggests natural relationships between the regret bounds of online learning algorithms and delays in the feedback.

Beyond analyzing existing algorithms, we hope that optimizing over the regret as a function of $D$ may inspire different (and hopefully simple) algorithms that readily model real world applications and scale nicely to distributed environments.

**Acknowledgements** We thank Avrim Blum for introducing us to the area of online learning and helping us with several valuable discussions. We thank the reviewers for their careful and insightful reviews: finding errors, referencing relevant works, and suggesting a connection to mirror descent.

# References

[1] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. 20th Int. Conf. Mach. Learning* (ICML), pages 928–936, 2003.

[2] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Sys. Sci.*, 71:291–307, 2005. Extended abstract in Proc. 16th Ann. Conf. Comp. Learning Theory *(COLT)*, 2003.

[3] A. Blum. On-line algorithms in machine learning. In A. Fiat and G. Woeginger, editors, *Online algorithms*, volume 1442 of *LNCS*, chapter 14, pages 306–325. Springer Berlin Heidelberg, 1998.

[4] S. Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, 2011.

[5] E. Hazan. Introduction to online convex optimization. Internet draft available at `http://ocobook.cs.princeton.edu`, 2015.

[6] E. Takimoto and M. Warmuth. Path kernels and multiplicative updates. *J. Mach. Learn. Research*, 4:773–818, 2003.

[7] D. Helmbold and R. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Mach. Learn. J.*, 27(1):61–68, 1997.

[8] A. Blum, S. Chawla, and A. Kalai. Static optimality and dynamic search optimality in lists and trees. *Algorithmica*, 36(3):249–260, 2003.

[9] T. M. Cover. Universal portfolios. *Math. Finance*, 1(1):1–29, 1991.

[10] K. Crammer and Y. Singer. A family of additive online algorithms for category ranking. *J. Mach. Learn. Research*, 3:1025–1058, 2003.

[11] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Quiñonero Candela. Practical lessons from predicting clicks on ads at facebook. In *Proc. 20th ACM Conf. Knowl. Disc. and Data Mining* (KDD), pages 1–9. ACM, 2014.

[12] S. Amuru and R. M. Buehrer. Optimal jamming using delayed learning. In *2014 IEEE Military Comm. Conf.* (MILCOM), pages 1528–1533. IEEE, 2014.

[13] I. Menache, O. Shamir, and N. Jain. On-demand, spot, or both: Dynamic resource allocation for executing batch jobs in the cloud. In *11th Int. Conf. on Autonomic Comput.* (ICAC), 2014.

[14] M.J. Weinberger and E. Ordentlich. On delayed prediction of individual sequences. *IEEE Trans. Inf. Theory*, 48(7):1959–1976, 2002.

[15] P. Joulani, A. György, and C. Szepesvári. Online learning under delayed feedback. In *Proc. 30th Int. Conf. Mach. Learning* (ICML), volume 28, 2013.

[16] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[17] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: a lock-free approach to parallelizing stochastic gradient descent. In *Adv. Neural Info. Proc. Sys. 24* (NIPS), pages 693–701, 2011.

[18] J. Duchi, M.I. Jordan, and B. McMahan. Estimation, optimization, and parallelism when data is sparse. In *Adv. Neural Info. Proc. Sys. 26* (NIPS), pages 2832–2840, 2013.

[19] H.B. McMahan and M. Streeter. Delay-tolerant algorithms for asynchronous distributed online learning. In *Adv. Neural Info. Proc. Sys. 27* (NIPS), pages 2915–2923, 2014.

[20] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Research*, 12:2121–2159, July 2011.

[21] J. Langford, A. J. Smola, and M. Zinkevich. Slow learners are fast. In *Adv. Neural Info. Proc. Sys. 22* (NIPS), pages 2331–2339, 2009.

[22] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar. An asynchronous parallel stochastic coordiante descent algorithm. *J. Mach. Learn. Research*, 16:285–322, 2015.

[23] J. C. Duchi, T. Chaturapruek, and C. Ré. Asynchronous stochastic convex optimization. *CoRR*, abs/1508.00882, 2015. To appear in Adv. Neural Info. Proc. Sys. 28 *(NIPS)*, 2015.

[24] S. J. Wright. Coordinate descent algorithms. *Math. Prog.*, 151(3–34), 2015.

[25] D. Riabko. *On the flexibility of theoretical models for pattern recognition*. PhD thesis, University of London, April 2005.

[26] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *J. Assoc. Comput. Mach.*, 44(3):426–485, 1997.

[27] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[28] A.S. Nemirovski and D.B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, 1983.

[29] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31:161–175, 2003.

[30] Y. Singer and S. Shalev-Shwartz. A primal-dual perspective of onine learning algoritmhs. *Mach. Learn.*, 69:115–142, 2007.

[31] S. Shalev-Shwartz. *Online Learning: Theory, algorithms, and applications*. PhD thesis, The Hebrew University of Jerusalem, July 2007.

[32] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Proc. 21st Ann. Conf. Comp. Learning Theory* (COLT), pages 57–68, 2008.

[33] A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization. Available online at `http://www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf`, Fall Semester 2013.

[34] R. T. Rockafeller. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, 1970.

[35] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1993.

[36] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1993.

[37] C. Zălinescu. *Convex analysis in general vector spaces*. World Scientific, 2002.

[38] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Springer, 2004.

[39] A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM J. Optim.*, 12(1):79–108, 2001.

[40] D.D. Sleator and R.E. Tarjan. Self-adjusting binary search trees. *J. Assoc. Comput. Mach.*, 32(3):652–686, 1985.

[41] N. Shavit and D. Touitou. Software transactional memory. *Dist. Comput.*, 10(2):99–116, 1997.

# A Mirror descent with delayed subgradients

**Geometries in $\mathbb{R}^n$ beyond $\|\cdot\|_2$** The algorithm `online-gradient-descent`, presented in Section 2, is designed for convex loss functions $f_t$ whose subgradients $\nabla f_t|_x$ have bounded *Euclidean* length $\|\nabla f_t|_x\|_2$. While $\|\cdot\|_2$ is a geometrically intuitive norm to analyze, there are settings where other norms on $\mathbb{R}^n$ are more natural.

For example, in expert selection, we identify each dimension $i \in [n]$ with one of $n$ experts. Each round $t \in [T]$, we select an expert $i \in [n]$ and then receive a cost vector $c_t \in [-1, 1]^n$, where the $i$th coordinate $c_{t,i} \in [-1, 1]$ measures the loss incurred by expert $i$ for this round. More precisely, we choose a distribution of experts defined by a point $x_t$ in the probability simplex $K = \left\{x \in [0, 1]^n : \|x\|_1 \overset{\text{def}}{=} \sum_{i=1}^n |x_i| = 1\right\}$. The $i$th coordinate $x_{t,i}$ corresponds to the probability of selecting expert $i$. The quantity $c_t \cdot x_t$ is the *expected* loss of the randomized strategy $x_t$. Note that the adversary selects $c_t$ with full knowledge of the randomized strategy $x_t$, but not the random bits that fix the choice of the $i$th expert.

A cost vector $c_t \in [-1, 1]^n$ can have Euclidean norm $\|c_t\|_2 = \sqrt{n}$, but its maximum norm is bounded by $\|c_t\|_\infty \overset{\text{def}}{=} \max\{|c_{t,1}|, \dots, |c_{t,n}|\} \leq 1$. Because it operates in the Euclidean norm, `online-gradient-descent` obtains a regret bound on the order of $O(\sqrt{nT})$ in expert selection, and one would prefer a regret bound with respect to the $L_\infty$-bound of 1 rather than the Euclidean bound of $\sqrt{n}$. Indeed, a different algorithm that selects experts in proportion to the exponential of the costs (among others) improves the regret bounds from $O(\sqrt{nT})$ to $O(\sqrt{\ln(n)T})$ [27].

In this section, we analyze an algorithm called `online-mirror-descent`, introduced by Nemirovski and Yudin for (offline) convex minimization [28], that generalizes both `online-gradient-descent` and randomized expert selection by exponential weights‡. The algorithm has a diverse body of analyses and interpretations [29, 30, 31, 32, 4, 33, 5], and our presentation is particularly influenced by the survey [4] and the lecture notes [33].

**Additional results from convex analysis** Before discussing `online-mirror-descent` in detail, we briefly review additional definitions and properties from convex analysis. These properties are all well known. Our overview is not comprehensive and in particular we limit ourselves to $\mathbb{R}^n$ for ease of exposition. We refer the reader to [34, 35, 36, 37, 38] for further background.

Let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function. The *epigraph* of $\varphi$ is the set $\operatorname{epi}\varphi = \{(x, \mu) \in \mathbb{R}^n \times \mathbb{R} : \mu \geq \varphi(x)\}$ in the higher-dimensional space $\mathbb{R}^{n+1}$. The function $\varphi$ is *closed* if $\operatorname{epi}\varphi$ is a closed subset of $\mathbb{R}^n$. The closure of a function $\varphi$ is the closed function whose epigraph is the closure of $\operatorname{epi}\varphi$. If the function $\varphi$ is closed, then the closure of $\varphi$ is $\varphi$. The *domain* of $\varphi$ is the

‡Technically, the form of `online-mirror-descent` presented here generalizes a lazy version of `online-gradient-descent`, that applies a gradient $\nabla f_t|_{x_t}$ to the unprojected point $x'_t$ rather than to the projected point $x_t = \pi_K(x'_t)$ to obtain the next unprojected point $x_t$. This variant of `online-gradient-descent` is also analyzed in [1].

convex set of points $\operatorname{dom}\varphi \overset{\text{def}}{=} \{x \in \mathbb{R}^n : \varphi(x) < \infty\}$ where $\varphi$ is finite. The function $\varphi$ is *proper* if $\operatorname{dom}\varphi \neq \emptyset$.

Let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function. The *Fenchel conjugate* of $\varphi$ is the function $\bar{\varphi} : \mathbb{R}^n \to \mathbb{R}$ defined by $\bar{\varphi}(\bar{x}) = \sup_{x \in \mathbb{R}^n} \langle x, \bar{x}\rangle - \varphi(x)$. The conjugate function $\bar{\varphi}$ is a closed and convex function, and proper if and only if $\varphi$ is [34, Theorem 2.2]. The conjugate $\bar{\bar{\varphi}}$ of the conjugate function $\bar{\varphi}$ is the closure of the original function $\varphi$, and equals $\varphi$ if $\varphi$ is closed. The subgradients of $\varphi$ and $\bar{\varphi}$ are related by the following.

**Lemma A.1** ([34, Theorem 23.5])**.** *Let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed and proper convex function, $x \in \mathbb{R}^n$, and $\bar{x} \in \mathbb{R}^n$. The following conditions are equivalent.*

*(i)* $\bar{x} \in \partial\varphi(x)$.
*(ii)* *The function $z \mapsto \langle z, \bar{x}\rangle - \varphi(z)$ attains its supremum in $z$ at $z = x$.*
*(iii)* $\varphi(x) + \bar{\varphi}(\bar{x}) \leq \langle x, \bar{x}\rangle$.
*(iv)* $\varphi(x) + \bar{\varphi}(\bar{x}) = \langle x, \bar{x}\rangle$.
*(v)* $x \in \partial\bar{\varphi}(\bar{x})$.
*(vi)* *The function $\bar{z} \mapsto \langle x, \bar{z}\rangle - \bar{\varphi}(\bar{z})$ attains its supremum at $\bar{z} = \bar{x}$.*

Consider any fixed norm $\|\cdot\|$ on $\mathbb{R}^n$. The *dual norm* of $\|\cdot\|$ is the norm $\|\cdot\|_\star$ on $\mathbb{R}^n$ defined by $\|y\|_\star = \sup_{\|x\|=1}|\langle x, y\rangle|$. A norm $\|\cdot\|$ and its dual norm $\|\cdot\|_\star$ satisfy a generalized form of the Cauchy-Schwartz inequality, $\langle x, y\rangle \leq \|x\|\|y\|_\star$. For example, the dual norm of the Euclidean norm $\|\cdot\|_2$ is again $\|\cdot\|_2$. The dual norm of the $L_\infty$-norm $\|\cdot\|_\infty$ is the $L_1$-norm $\|\cdot\|_1$. For $p > 1$, the dual norm of the $L_p$-norm $\|x\|_p \overset{\text{def}}{=} \left(\sum_{i=1}^n |x|^p\right)^{1/p}$ is the $L_q$-norm $\|\cdot\|_q$, where $q$ satisfies $1/p + 1/q = 1$.

Let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function. For a constant $\rho > 0$, we say $f$ is *smooth with modulus* $\rho$ with respect to a norm $\|\cdot\|$ if it is differentiable and has $\rho$-Lipschitz-continuous gradients. That is, for any two points $x, y \in \mathbb{R}^n$, we have $\|\nabla\varphi|_x - \nabla\varphi|_y\|_\star \leq \rho\|x - y\|$. There are many equivalent conditions for a convex function to be smooth; see, for example, [38, Theorem 2.1.5] or [37, Section 3.5].

Let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function and $\sigma > 0$ a constant. The function $\varphi$ is *strongly convex with modulus* $\sigma$ with respect to a norm $\|\cdot\|$ if the function $\psi(x) \overset{\text{def}}{=} \varphi(x) - (\sigma/2)\|x\|^2$ is also convex. As with smoothness, there are many equivalent properties for a function to be strongly convex; see, for example, [38, Section 2.1.3] or [37, Section 3.5]. In particular, for any two points $x, y \in \mathbb{R}^n$ and subgradient $\nabla\varphi|_x \in \partial\varphi(x)$, we have $\varphi(y) \geq \varphi(x) + \langle y - x, \nabla\varphi|_x\rangle + (\sigma/2)\|y - x\|^2$.

Smoothness and strong convexity are dual properties in the following sense.

**Lemma A.2** ([37, Section 3.5], [4, Lemma 2.19])**.** *Let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed and convex function. Then $\varphi$ is strongly convex with modulus $\sigma$ with respect to a norm $\|\cdot\|$ if and only if $\bar{\varphi}$ is smooth with modulus $\sigma^{-1}$ with respect to the dual norm $\|\cdot\|_\star$.*

**Mirror descent** Let $K$ be a closed convex set in $\mathbb{R}^n$. Fix a norm $\|\cdot\|$ in $\mathbb{R}^n$, and let $\varphi$ be a closed and strongly convex function on $\mathbb{R}^n$. The strongly convex function $\varphi$ is called the *regularizer*, and is required to have subgradients in the closed input domain $K$. Without loss of generality, we assume that $\varphi(x) = \infty$ over all nonfeasible points $x \notin K$.

Each round $t \in [T]$, `online-mirror-descent` has both a *primal point* $x_t \in K$ and a *dual point* $\bar{x}_t \in \mathbb{R}^n$. We choose the first dual point to be 0. Recall that by Lemma A.2, since $\varphi$ is strongly convex, its Fenchel conjugate $\bar{\varphi}$ is smooth and in particular differentiable everywhere. Given the $t$th dual point $\bar{x}_t$, the primal point $x_t$ is the derivative of the conjugate $\bar{\varphi}$ at $\bar{x}_t$, $x_t \overset{\text{def}}{=} \nabla\bar{\varphi}|_{\bar{x}_t}$. Given a subgradient $\nabla f_t|_{x_t} \in \partial f_t(x_t)$, we update the dual point against the gradient by the formula $\bar{x}_{t+1} \overset{\text{def}}{=} \bar{x}_t - \eta\nabla f_t|_{x_t}$. The overall process looks like,

$$\ldots, \quad x_t \overset{\text{def}}{=} \nabla\bar{\varphi}|_{\bar{x}_t}, \quad \bar{x}_{t+1} \overset{\text{def}}{=} \bar{x}_t - \eta\nabla f_t|_{x_t}, \quad x_{t+1} \overset{\text{def}}{=} \nabla\bar{\varphi}|_{\bar{x}_{t+1}}, \quad \ldots$$

Intuitively, all the information (in the form of gradients) exists in the dual space (with norm $\|\cdot\|_\star$), and is "mirrored" back into the primal space (with the original norm $\|\cdot\|$) via the smooth conjugate function $\bar{\varphi}$.

If we inline the role of the dual point and apply Lemma A.1, the next point $x_{t+1}$ is chosen as a function of $x_t$ as

$$x_{t+1} = \nabla\bar{\varphi}|_{\bar{x}_{t+1}} = \nabla\bar{\varphi}|_{\bar{x}_t - \eta\nabla f_t|_{x_t}} = \arg\max_{x\in K}\{\langle x, \bar{x}_t - \eta\nabla f_t|_{x_t}\rangle - \varphi(x)\}.$$

If we expand this last term further, then the algorithm can be written equivalently as

$$\begin{aligned} x_{t+1} &= \arg\max_{x\in K}\{\langle x, \bar{x}_t\rangle - \eta\,\langle x, \nabla f_t|_{x_t}\rangle - \varphi(x)\} \\ &= \arg\max_{x\in K}\{\varphi(x_t) + \langle x - x_t, \bar{x}_t\rangle - \eta\,\langle x, \nabla f_t|_{x_t}\rangle - \varphi(x)\} \\ &= \arg\max_{x\in K}\{-\eta\,\langle x, \nabla f_t|_{x_t}\rangle - D_\varphi(x|x_t, \bar{x}_t)\} \\ &= \arg\min_{x\in K}\{\langle x, \eta\nabla f_t|_{x_t}\rangle + D_\varphi(x|x_t, \bar{x}_t)\}, \end{aligned} \tag{6}$$

where $D_\varphi(x|x_t, \bar{x}_t) \overset{\text{def}}{=} \varphi(x) - \varphi(x_t) - \langle x - x_t, \bar{x}_t\rangle$ is the Bregman divergence of $\varphi$ from $x_t$ to $x$ with the choice of subgradient $\bar{x}_t \in \partial\varphi(x_t)$. The Bregman divergance $D_\varphi(x|x_t, \bar{x}_t)$ measures the amount of error between the actual value of $\varphi$ at $x$ and its first-order approximation from $x_t$ with respect to the subgradient $\bar{x}_t \in \partial\varphi(x_t)$.[§] By convexity of $\varphi$, $D_\varphi(\cdot|\cdot)$ is always nonnegative. Moreover, because $\varphi$ is strongly convex, $D_\varphi(x|x_t, \bar{x}_t)$ grows quadratically in the distance between $x$ and $x_t$. Despite these and other nice properties, $D_\varphi(x|x_t, \bar{x}_t)$ satisfies neither the triangle inequality nor symmetry. The final formulation of mirror descent in equation (6) chooses $x_{t+1}$ greedily against the direction of the gradient, but the greediness is tempered by the Bregman divergance $D_\varphi(\cdot|x_t, \bar{x}_t)$.

The Bregman divergence satisfies the following identity that we will use. For $x, y, z \in \mathbb{R}^n$, and subgradients $\bar{x} \in \partial\varphi(x)$ and $\bar{y} \in \partial\varphi(y)$, we have

$$\langle z - y, \bar{y} - \bar{x}\rangle = D_\varphi(z|x, \bar{x}) - D_\varphi(z|y, \bar{y}) - D_\varphi(y|x, \bar{x}) \tag{7}$$

(see, for example, [33, Fact 5.3.3]).

*Example* A.3. For a closed convex set $K$, consider the function

$$\varphi(x) \overset{\text{def}}{=} \begin{cases} \frac{1}{2}\|x\|_2^2 & \text{if } x \in K. \\ +\infty & \text{otherwise.} \end{cases}$$

The function $\varphi$ is strongly convex (with modulus 1) with respect to the Euclidean norm $\|\cdot\|_2$. The derivative of the conjugate $\bar{\varphi}$ takes the dual point $\bar{x} \in \mathbb{R}^n$ to its closest point in $K$. Thus `online-mirror-descent`, with this choice of norm and regularizer, resembles a lazy version of the `online-gradient-descent` algorithm considered in Section 2.

*Example* A.4 ([39, 29]). Let $K$ be the probability simplex in $\mathbb{R}^n$, and consider the negative entropy function

$$\varphi(x) \overset{\text{def}}{=} \begin{cases} \sum_{i=1}^n x_i \ln x_i & \text{if } x \in K, \\ +\infty & \text{otherwise} \end{cases}$$

where $0 \ln 0 \overset{\text{def}}{=} 0$. The negative entropy function $\varphi$ is strongly convex (with modulus 1) with respect to the $L_1$-norm, and its Frenchel conjugate is the log-partition function $\bar{\varphi}(\bar{x}) \overset{\text{def}}{=} \ln(\sum_{i=1}^n \exp(\bar{x}_i))$. The derivative of $\bar{\varphi}$ is defined coordinate-wise by $(\nabla\bar{\varphi}|_{\bar{x}})_i \overset{\text{def}}{=} \exp(\bar{x}_i)/(\sum_{i'=1}^n \exp(\bar{x}_{i'}))$. Note that $\nabla\bar{\varphi}|_{\bar{x}}$ is always a point in the probability simplex. In the context of expert selection, the `online-mirror-descent` algorithm with this choice of norm and regularizer randomly selects an expert $i$ in round $t$ in proportion to the exponential weight.

**Mirror descent in the delayed setting** In the delayed setting, the subgradient of the loss function $f_t$ is not necessarily given by the adversary before we update the dual and pick the next point $x_{t+1}$ (or even at all). The natural generalization of `online-mirror-descent` to this setting is to process the convex loss functions and apply their subgradients to the dual point the moment they are delivered. That is, we update

$$\bar{x}_{t+1} = \bar{x}_t - \eta \sum_{s\in\mathcal{F}_t} \nabla f_s|_{x_s}$$

[§]We make the choice of subgradient explicit because $\varphi$ is not necessarily differentiable.

for some fixed parameter $\eta$, and then project $x_{t+1} = \nabla\bar{\varphi}|_{\bar{x}_{t+1}}$. In the undelayed setting, we have $\mathcal{F}_t = \{t\}$ for each $t$, and this algorithm is exactly `online-mirror-descent`.

**Theorem A.5.** *Let $K \subseteq \mathbb{R}^n$ be a closed convex set. For a fixed norm $\|\cdot\|$ on $\mathbb{R}^n$, let $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed and strongly convex function with modulus $\sigma$ and with $\operatorname{dom}\varphi = K$. Let $f_1, \ldots, f_T$ be convex functions with subgradients satisfying $\|\nabla f_t|_x\|_\star \leq L$ for all $x \in K$ and $t \in T$. In the presence of adversarial delays,* `online-mirror-descent` *selects points $x_1, \ldots, x_T \in K$ such that for all $y \in K$,*

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(y) = O\left(\frac{1}{\eta}(\varphi(y) - \varphi(x_1)) + \eta\frac{L^2(T+D)}{\sigma}\right).$$

In the gradient descent setup of example A.3, the term $\varphi(y) - \varphi(x_1)$ is bounded above by the square of the diameter $\Delta \overset{\text{def}}{=} \sup_{x,y \in K}\|x - y\|_2$ of $K$. For $\eta = \Theta(\Delta\sqrt{D}/L)$, we again achieve a regret bound on the order of $O(\Delta L\sqrt{D})$, matching Theorem 2.1 (there we treated $\Delta$ as a constant). In the expert selection setting of example A.4, $\varphi(y) - \varphi(x_1)$ is bounded above by $O(\ln(n))$, and for $\eta = \Theta(\sqrt{\ln(n)D})$, we achieve a regret bound on the order of $O(\sqrt{\ln(n)D})$.

*Proof of Theorem A.5.* Let $y \in K$ be any fixed point in $K$. By convexity, we first bound the regret by

$$\sum_{t=1}^{T}(f_t(x_t) - f_t(y)) \leq \sum_{t=1}^{T} \langle x_t - y, \nabla f_t|_{x_t}\rangle . \tag{8}$$

The standard (undelayed) analysis for mirror descent proceeds to analyze each summand $\langle x_t - y, \nabla f_t|_{x_t}\rangle$ with the knowledge that $\nabla f_t|_{x_t}$ is applied to (the dual of) $x_t$. This is not true in the delayed setting, and we first align each gradient $\nabla f_s|_{x_s} \in \mathcal{F}_t$ alongside the primal point to which it is effectively applied.

As in the proof for `online-gradient-descent`, we split the sum of subgradients $\sum_{s\in\mathcal{F}_t} \nabla f_s|_{x_s}$ applied in a single round $t \in [T]$ and apply them one by one in increasing order of their originating round $s$. For each $s \in \mathcal{F}_t$, let $\mathcal{F}_{t,s} \overset{\text{def}}{=} \{r \in \mathcal{F}_t : r < s\}$ be the indices of other gradients that are applied in round $t$ before $\nabla f_s|_{x_s}$. Let $\bar{x}_s^- \overset{\text{def}}{=} \bar{x}_t - \eta\sum_{r\in\mathcal{F}_{t,s}} \nabla f_r|_{x_r}$ be the intermediate dual point to which we apply $\nabla f_s|_{x_s}$, and let $\bar{x}_s^+ \overset{\text{def}}{=} \bar{x}_s^- - \eta\nabla f_s|_{x_s}$ be the intermediate dual point after applying $\nabla f_s|_{x_s}$. Let $x_s^- \overset{\text{def}}{=} \nabla\bar{\varphi}|_{\bar{x}_s^-}$ and $x_s^+ \overset{\text{def}}{=} \nabla\bar{\varphi}|_{\bar{x}_s^+}$ be their projected primal points in $K$.

Rearranging the sum in equation (8) and organizing the gradients by point of delivery, we have

$$\begin{aligned}\sum_{t=1}^{T} \langle x_t - y, \nabla f_t|_{x_t}\rangle &= \sum_{t=1}^{T}\sum_{s\in\mathcal{F}_t} \langle x_s - y, \nabla f_s|_{x_s}\rangle \\ &= \sum_{t=1}^{T}\sum_{s\in\mathcal{F}_t} \left\langle x_s^- - y, \nabla f_s|_{x_s}\right\rangle + \sum_{t=1}^{T}\sum_{s\in\mathcal{F}_t} \left\langle x_s - x_s^-, \nabla f_s|_{x_s}\right\rangle . \end{aligned} \tag{9}$$

The first sum is familiar from the standard analysis of undelayed mirror descent and will be analyzed by similar techniques. The second term generalizes the delay term in equation (2) from the earlier proof for `online-gradient-descent`.

To analyze the first sum, fix $t \in [T]$ and $s \in \mathcal{F}_t$, and expand the summand

$$\left\langle x_s^- - y, \nabla f_s|_{x_s}\right\rangle = \left\langle x_s^+ - y, \nabla f_s|_{x_s}\right\rangle + \left\langle x_s^- - x_s^+, \nabla f_s|_{x_s}\right\rangle .$$

The subsequent primal point $x_s^+$ minimizes the function $\psi_s(x) \overset{\text{def}}{=} \langle x, \eta\nabla f_s|_{x_s} - \bar{x}_s^-\rangle + \varphi(x)$ over $x \in K$. Since $\psi_s$ is convex, $K$ is convex, and $y \in K$, the first-order conditions at $x_s^+$ imply that $\langle y - x_s^+, \delta\rangle \geq 0$ for any subgradient $\delta$ of $\psi_s$ at $x_s^+$. In particular, at the subgradient $\eta\nabla f_s|_{x_s} - \bar{x}_s^- + \bar{x}_s^+ \in \partial\psi_s(x_s^+)$, we have,

$$\left\langle y - x_s^+, \eta\nabla f_s|_{x_s} - \bar{x}_s^- + \bar{x}_s^+\right\rangle \geq 0;$$

rearranging, this implies that

$$\left\langle x_s^+ - y, \nabla f_s|_{x_s} \right\rangle \leq \frac{1}{\eta} \left\langle y - x_s^+, \bar{x}_s^+ - \bar{x}_s^- \right\rangle.$$

Furthermore, by the aforementioned identity in equation (7), we have

$$\left\langle y - x_s^+, \bar{x}_s^+ - \bar{x}_s^- \right\rangle = D_\varphi\big(y|x_s^-, \bar{x}_s^-\big) - D_\varphi\big(y|x_s^+, \bar{x}_s^+\big) - D_\varphi\big(x_s^+|x_s^-, \bar{x}_s^-\big).$$

Thus,

$$\begin{aligned} \left\langle x_s^- - y, \nabla f_s|_{x_s} \right\rangle &= \left\langle x_s^+ - y, \nabla f_s|_{x_s} \right\rangle + \left\langle x_s^- - x_s^+, \nabla f_s|_{x_s} \right\rangle \\ &\leq \frac{1}{\eta}\big(D_\varphi\big(y|x_s^-, \bar{x}_s^-\big) - D_\varphi\big(y|x_s^+, \bar{x}_s^+\big) - D_\varphi\big(x_s^+|x_s^-, \bar{x}_s^-\big)\big) + \left\langle x_s^- - x_s^+, \nabla f_s|_{x_s} \right\rangle \\ &= \frac{1}{\eta}\big(D_\varphi\big(y|x_s^-, \bar{x}_s^-\big) - D_\varphi\big(y|x_s^+, \bar{x}_s^+\big)\big) + \frac{1}{\eta}\big(\left\langle x_s^- - x_s^+, \nabla f_s|_{x_s} \right\rangle - D_\varphi\big(x_s^+|x_s^-, \bar{x}_s^-\big)\big). \end{aligned} \tag{10}$$

The first term will later telescope over all rounds $s \in [T]$. Consider the second term $\langle x_s^- - x_s^+, \eta \nabla f_s|_{x_s}\rangle - D_\varphi(x_s^+|x_s^-, \bar{x}_s^-)$. Since $\varphi$ is strongly convex with modulus $\sigma$, we have $D_\varphi(x_s^+|x_s^-, \bar{x}_s^-) \geq \frac{\sigma}{2}\|x_s^+ - x_s^-\|^2$. Thus,

$$\begin{aligned} &\left\langle x_s^- - x_s^+, \eta\nabla f_s|_{x_s} \right\rangle - D_\varphi\big(x_s^+|x_s^-, \bar{x}_s^-\big) \\ &\quad\leq \left\langle x_s^- - x_s^+, \eta\nabla f_s|_{x_s} \right\rangle - \frac{\sigma}{2}\left\|x_s^+ - x_s^-\right\|^2 && \text{by strong convexity of } \varphi, \\ &\quad\leq \left\|x_s^+ - x_s^-\right\| \|\eta\nabla f_s|_{x_s}\|_\star - \frac{\sigma}{2}\left\|x_s^+ - x_s^-\right\|^2 && \text{by Cauchy-Schwartz,} \\ &\quad\leq \frac{1}{2\sigma}\|\eta\nabla f_s|_{x_s}\|_\star^2 && \text{by the identity } 2ab - a^2 \leq b^2, \\ &\quad= O\left(\frac{\eta^2 L^2}{\sigma}\right) && \text{by assumption.} \end{aligned} \tag{11}$$

Plugging equation (11) into equation (10), we have,

$$\left\langle x_s^- - y, \nabla f_s|_{x_s} \right\rangle \leq \frac{1}{\eta}\big(D_\varphi\big(y|x_s^-, \bar{x}_s^-\big) - D_\varphi\big(y|x_s^+, \bar{x}_s^+\big)\big) + O\left(\frac{\eta L^2}{\sigma}\right).$$

Summed over all $s \in \mathcal{F}_t$ and $t \in [T]$ and telescoping, we have,

$$\begin{aligned} \sum_{t=1}^T \sum_{s \in \mathcal{F}_t} \left\langle x_s^- - y, \nabla f_s|_{x_s} \right\rangle &= \sum_{t=1}^T \sum_{s \in \mathcal{F}_t} \left(\frac{1}{\eta}\big(D_\varphi\big(y|x_s^-, \bar{x}_s^-\big) - D_\varphi\big(y|x_s^+, \bar{x}_s^+\big)\big) + O\left(\frac{\eta L^2}{\sigma}\right)\right) \\ &= \frac{1}{\eta}(D_\varphi(y|x_1, \bar{x}_1) - D_\varphi(y|x_{T+1}, \bar{x}_{T+1})) + O\left(\frac{\eta L^2 T}{\sigma}\right). \end{aligned} \tag{12}$$

arriving at the usual bound for (undelayed) mirror descent.

It remains to analyze the delay term $\sum_{t=1}^T \sum_{s \in \mathcal{F}_t} \langle x_s - x_s^-, \nabla f_s|_{x_s}\rangle$. As in the case of `online-gradient-descent`, each summand $\langle x_s - x_s^-, \nabla f_s|_{x_s}\rangle$ reflects the distance between the point $x_s$ when $\nabla f_s|_{x_s}$ is generated, and the point $x_s^-$ when $\nabla f_s|_{x_s}$ is applied. By Cauchy-Schwartz, the delay term is bounded above by

$$\sum_{t=1}^T \sum_{s \in \mathcal{F}_t} \left\langle x_s - x_s^-, \nabla f_s|_{x_s} \right\rangle \leq \sum_{t=1}^T \sum_{s \in \mathcal{F}_t} \left\|x_s - x_s^-\right\| \|\nabla f_s|_{x_s}\|_\star \leq L \sum_{t=1}^T \sum_{s \in \mathcal{F}_t} \left\|x_s - x_s^-\right\|.$$

Fix $s$ and consider the distance $\|x_s - x_s^-\|$. Applying the triangle inequality many times over, we first write

$$\left\|x_s - x_s^-\right\| \leq \sum_{r=s}^{t-1} \|x_r - x_{r+1}\| + \|x_t - x_{t,s}\|$$

Fix $r \in \{s, s+1, \ldots, t-1\}$ and consider a single term $\|x_r - x_{r+1}\|$. This is the distance traversed by the primal point from applying the gradients $\sum_{q \in \mathcal{F}_r} \nabla f_q|_{x_q}$ to $x_r$. In terms of derivatives of the dual regularizer, we have $\|x_r - x_{r+1}\| = \left\|\nabla\bar{\varphi}|_{\bar{x}_r} - \nabla\bar{\varphi}|_{\bar{x}_{r+1}}\right\|$, where $\bar{x}_{r+1} = \bar{x}_r - \eta \sum_{q \in \mathcal{F}_r} \nabla f_q|_{x_q}$. As the Fenchel conjugate of a strongly convex function with modulus $\sigma$, $\bar{\varphi}$ is smooth with modulus $\sigma^{-1}$, and in particular,

$$\left\|\nabla\bar{\varphi}|_{\bar{x}_r} - \nabla\bar{\varphi}|_{\bar{x}_{r+1}}\right\| = O\left(\frac{1}{\sigma}\|\bar{x}_r - \bar{x}_{r+1}\|_\star\right) = O\left(\frac{1}{\sigma}\left\|\eta \sum_{q \in \mathcal{F}_r} \nabla f_q|_{x_q}\right\|_\star\right).$$

Similarly, we have $\|x_t - x_{s,t}\| \le O\left(\sigma^{-1}\left\|\eta \sum_{p \in \mathcal{F}_{t,s}} \nabla f_p|_{x_p}\right\|_\star\right)$. Thus,

$$\begin{aligned}
\left\|x_s - x_s^-\right\| &= O\left(\sum_{r=s}^{t-1} \frac{1}{\sigma}\left\|\eta \sum_{q \in \mathcal{F}_r} \nabla f_q|_{x_q}\right\|_\star + \frac{1}{\sigma}\left\|\eta \sum_{p \in \mathcal{F}_{t,s}} \nabla f_p|_{x_p}\right\|_\star\right) \\
&= O\left(\frac{\eta}{\sigma}\left(\sum_{r=s}^{t-1} \sum_{q \in \mathcal{F}_r} \left\|\nabla f_q|_{x_q}\right\|_\star + \sum_{p \in \mathcal{F}_{t,s}} \left\|\nabla f_p|_{x_p}\right\|_\star\right)\right) \\
&= O\left(\frac{\eta L}{\sigma}\left(\sum_{r=s}^{t-1} |\mathcal{F}_r| + |\mathcal{F}_{t,s}|\right)\right).
\end{aligned}$$

With respect to the delay term, we have

$$\sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \left\langle x_s - x_s^-, \nabla f_s|_{x_s}\right\rangle \le L \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_s} \left\|x_s^- - x_s\right\| = O\left(\frac{\eta L^2}{\sigma} \sum_{t=1}^{T} \sum_{s \in \mathcal{F}_s} \left(\sum_{r=s}^{t-1} |\mathcal{F}_r| + |\mathcal{F}_{t,s}|\right)\right).$$

This sum was encountered earlier in the proof for `online-gradient-descent`, and the same argument shows that the delay term is bounded by $\sum_{t=1}^{T} \sum_{s \in \mathcal{F}_t} \langle x_s - x_s^-, \nabla f_s|_{x_s}\rangle = O(\eta L^2 D/\sigma)$. Plugging this inequality along with equation (12) into equation (8), we bound the regret by

$$\begin{aligned}
R(T) &\le \frac{1}{\eta}(D_\varphi(y|x_1, \bar{x}_1) - D_\varphi(y|x_{T+1}, \bar{x}_{T+1})) + \eta \frac{TL^2}{2\sigma} + O\left(\eta \frac{DL^2}{\sigma}\right) \\
&= O\left(\frac{1}{\eta}(\varphi(y) - \varphi(x_1)) + \eta \frac{L^2(T+D)}{\sigma}\right),
\end{aligned}$$

as desired. ■

# B Late and lazy leaders

A basic application of online combinatorial optimization is adaptive data structures. For example, in the *tree update problem*, one maintains a binary search tree over a collection of $n$ items, and serves an unknown sequence of access queries. If the cost of accessing an item is proportional to the depth of the item in the tree, then one prefers to have frequently placed items near the root, and less frequent items at the leaves. One classical data structure for this problem is the splay tree, by Sleater and Tarjan, who also introduced the tree update problem [40]. Kalai and Vempala observe that this problem can partially be solved by `follow-the-perturbed-leader`. We simply track the sum of access costs for each item, and each round we rebuild an optimal binary search tree with respect to those costs (plus a perturbation) by dynamic programming in $O(n^2)$ time. Compared to the cost of the best fixed tree in hindsight, this algorithm has an additive loss of $O(n\sqrt{nT})$ to serve $T$ requests. This bound overlooks the fact that rebuilding a binary search tree between every two queries is absurdly inefficient.

To model this and similar problems efficiently, one introduces the notion of a *switching cost*, which is a fixed value that is charged every time we change our decision between subsequent rounds. For this setting, Kalai and Vempala gave a similar algorithm called `follow-the-lazy-`

leader, which is nearly equivalent to `follow-the-perturbed-leader`, but changes decisions very infrequently. Happily, `follow-the-lazy-leader` is just as simple as `follow-the-perturbed-leader`.

Initially, the algorithm picks a point $p \in Q_\epsilon = \left[0, \epsilon^{-1}\right]^n$ uniformly at random. This defines a grid, $G_\epsilon = \left\{p + \epsilon^{-1} z : z \in \mathbb{Z}^n\right\}$, of width $\epsilon^{-1}$ and shifted randomly. Rather than follow the perturbed leader, we follow the leader and round it up to its nearest grid point. Formally, we define the $t$th *lazy leader* to be the unique grid point $\overline{y}_t \in G_\epsilon$ of the square $y_t + Q_0$. Because the grid is randomly shifted by the same distribution as the perturbation in `follow-the-perturbed-leader`, the perturbed leader and lazy leader are distributed identically and have identical expected costs [2, Lemma 1.2]. This observation immediately implies that `follow-the-lazy-leader` obtains regret $O(\sqrt{T})$[¶].

Extending `follow-the-lazy-leader` to the delayed setting has natural applications. For example, in the tree update problem, it is preferable to be able to serve many queries simultaneously, and to keep serving requests from an outdated search tree while building the next tree in parallel. One simple implementation to satisfy these requirement is to keep a pointer to the tree in *software transactional memory* (STM) [41]. When `follow-the-lazy-leader` shifts grid points, we build the new search tree in the background and swap the pointer in the STM once the new tree is completed. Here, the delays arise from serving requests from an outdated tree before the tree is replaced by a new one.

**Theorem B.1.** *Let $K \subseteq \mathbb{R}^n$ be a set with $L_1$-diameter $\leq 1$, let $c_1, \ldots, c_T \in \mathbb{R}^n_{\geq 0}$ be non-negative vectors with $\|c_t\|_1 \leq 1$ for all $t$, and let $\epsilon > 0$. In the presence of adversarial delays,* `follow-the-lazy-leader` *picks points $x_1, \ldots, x_T \in K$ such that two consecutive points $x_i$ and $x_{i+1}$ differ at most $\epsilon T$ times, and for all $y \in K$,*

$$\sum_{t=1}^{T} \mathbf{E}[c_t \cdot x_t] \leq \sum_{t=1}^{T} c_t \cdot y + O\big(\epsilon^{-1} + \epsilon D\big).$$

*Proof.* In the delayed setting, the $t$th delayed leader $\overline{y}^{\mathrm{d}}_t$ is the best point with respect to the grid point corresponding to the sum of all feedback through the first $t$ rounds. More formally, let $g : \mathbb{R}^n \to G_\epsilon$ be the mapping each point to the first grid point greater than or equal in each dimension to the point. Formally, we have

$$\overline{y}^{\mathrm{d}}_t = \arg\min_{x \in K} x \cdot g\left(\sum_{s=1}^{t} \sum_{r \in \mathcal{F}_s} c_r\right)$$

for each round $t$ (as before, we use the superscript $\mathrm{d}$ to emphasize the delayed setting). Following the lazy leader is the strategy that chooses the $t$th point $x_t$ as $x_t = \overline{y}^{\mathrm{d}}_{t-1}$ for each round $t$. Observe that since each cost vector $c_i$ is nonnegative, and their sum has $L_1$-norm $\|c_1 + \cdots + c_T\|_1 \leq T$, $\overline{y}^{\mathrm{d}}_t$ changes grid points at most $\epsilon T$ times.

The same proof as [2, Lemma 1.2] shows that each $\overline{y}^{\mathrm{d}}_t$ is distributed identically as $\tilde{y}^{\mathrm{d}}_t$. By linearity of expectation, $\mathbf{E}\left[c_t \cdot \overline{y}^{\mathrm{d}}_{t-1}\right] = \mathbf{E}\left[c_t \cdot \tilde{y}^{\mathrm{d}}_{t-1}\right]$ for all $t \in [T]$. Summing this equality over all rounds $t \in [T]$ and applying the regret bound for `follow-the-perturbed-leader` (Theorem 3.1) gives the bounds we seek. ■

[¶]Kalai and Vempala's regret bound for `follow-the-lazy-leader` has the caveat that the adversary is oblivious to the random shift $p \in Q_\epsilon$, which applies here as well.