Journal of Machine Learning Research 18 (2017) 1-50 Submitted 10/14; Revised 11/15; Published 8/17

A Survey of Algorithms and Analysis for Adaptive Online Learning

H. Brendan McMahan
mcmahan@google.com
Google, Inc.
651 N 34th St
Seattle, WA 98103 USA

Editor: Shie Mannor

Abstract

We present tools for the analysis of Follow-The-Regularized-Leader (FTRL), Dual Averaging, and Mirror Descent algorithms when the regularizer (equivalently, prox-function or learning rate schedule) is chosen adaptively based on the data. Adaptivity can be used to prove regret bounds that hold on every round, and also allows for data-dependent regret bounds as in AdaGrad-style algorithms (e.g., Online Gradient Descent with adaptive per-coordinate learning rates). We present results from a large number of prior works in a unified manner, using a modular and tight analysis that isolates the key arguments in easily re-usable lemmas. This approach strengthens previously known FTRL analysis techniques to produce bounds as tight as those achieved by potential functions or primal-dual analysis. Further, we prove a general and exact equivalence between adaptive Mirror Descent algorithms and a corresponding FTRL update, which allows us to analyze Mirror Descent algorithms in the same framework. The key to bridging the gap between Dual Averaging and Mirror Descent algorithms lies in an analysis of the FTRL-Proximal algorithm family. Our regret bounds are proved in the most general form, holding for arbitrary norms and non-smooth regularizers with time-varying weight.

Keywords: online learning, online convex optimization, regret analysis, adaptive algorithms, follow-the-regularized-leader, mirror descent, dual averaging

1. Introduction

We consider the problem of online convex optimization over a series of rounds $t \in \{1, 2, \dots\}$. On each round the algorithm selects a point (e.g., a predictor or an action) $x_t \in \mathbb{R}^n$, and then an adversary selects a convex loss function f_t, and the algorithm suffers loss $f_t(x_t)$. The goal is to minimize

$$\text{Regret}_T(x^*, f_t) \equiv \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*), \tag{1}$$

Algorithm 1 General Template for Adaptive FTRL

Parameters: Scheme for selecting convex r_t s.t. $\forall x,\ r_t(x) \geq 0$ for $t = 0, 1, 2, \ldots$
$x_1 \leftarrow \arg\min_{x \in \mathbb{R}^n}\ r_0(x)$
for $t = 1, 2, \ldots$ **do**
  Observe convex loss function $f_t : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$
  Incur loss $f_t(x_t)$
  Choose incremental convex regularizer r_t, possibly based on $f_1, \ldots f_t$
  Update

$$x_{t+1} \leftarrow \arg\min_{x \in \mathbb{R}^n} \sum_{s=1}^{t} f_s(x) + \sum_{s=0}^{t} r_s(x)$$

end for

the difference between the algorithm's loss and the loss of a fixed point x^*, potentially chosen with full knowledge of the sequence of f_t up through round T. When the functions f_t and round T are clear from the context we write $\text{Regret}(x^*)$. The "adversary" choosing the f_t need not be malicious, for example the f_t might be drawn from a distribution. The name "online convex optimization" was introduced by Zinkevich (2003), though the setting was introduced earlier by Gordon (1999). When a particular set of comparators $\mathcal{X}$ is fixed in advance, one is often interested in $\text{Regret}(\mathcal{X}) \equiv \sup_{x^* \in \mathcal{X}} \text{Regret}(x^*)$; since $\mathcal{X}$ is often a norm ball, frequently we bound $\text{Regret}(x^*)$ by a function of $\|x^*\|$.

Online algorithms with good regret bounds (that is, bounds that are sublinear in T) can be used for a wide variety of prediction and learning tasks (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2012). The case of online logistic regression, where one predicts the probability of a binary outcome, is typical. Here, on each round a feature vector $a_t \in \mathbb{R}^n$ arrives, and we make a prediction $p_t = \sigma(a_t \cdot x_t) \in (0, 1)$ using the current model coefficients $x_t \in \mathbb{R}^n$, where $\sigma(z) = 1/(1 + e^{-z})$. The adversary then reveals the true outcome $y_t \in \{0, 1\}$, and we measure loss with the negative log-likelihood, $\ell(p_t, y_t) = -y_t \log p_t - (1 - y_t) \log(1 - p_t)$. We encode this problem as online convex optimization by taking $f_t(x) = \ell(\sigma(a_t \cdot x), y_t)$; these f_t are in fact convex. Linear Support Vector Machines (SVMs), linear regression, and many other learning problems can be encoded in a similar manner; Shalev-Shwartz (2012) and many of the other works cited here contain more details and examples.

We consider the family of Follow-The-Regularized-Leader (FTRL, or FoReL) algorithms as shown in Algorithm 1 (Shalev-Shwartz, 2007; Shalev-Shwartz and Singer, 2007; Rakhlin, 2008; McMahan and Streeter, 2010; McMahan, 2011). Shalev-Shwartz (2012) and Hazan (2015) provide a comprehensive survey of analysis techniques for non-adaptive members of this algorithm family, where the regularizer is fixed for all rounds and chosen with knowledge of the horizon T. In this survey,

we allow the regularizer to change adaptively. Given a sequence of incremental regularization functions $r_0, r_1, r_2, \ldots$, we consider the algorithm that selects

$$\begin{aligned} x_1 &\in \arg\min_{x \in \mathbb{R}^n} r_0(x) \\ x_{t+1} &= \arg\min_{x \in \mathbb{R}^n} f_{1:t}(x) + r_{0:t}(x) \qquad \text{for } t = 1, 2, \ldots, \end{aligned} \tag{2}$$

where we use the compressed summation notation $f_{1:t}(x) = \sum_{s=1}^{t} f_s(x)$ (we also use this notation for sums of scalars or vectors). The argmin in Eq. (2) is over all $\mathbb{R}^n$, but it is often necessary to constrain the selected points x_t to a convex feasible set $\mathcal{X}$. This can be accomplished in our framework by including the indicator function $I_{\mathcal{X}}$ as a term in r_0 ($I_{\mathcal{X}}$ is a convex function defined by $I_{\mathcal{X}}(x) = 0$ for $x \in \mathcal{X}$ and ∞ otherwise); details are given in Section 2.4. The algorithms we consider are adaptive in that each r_t can be chosen based on $f_1, f_2, \ldots, f_t$. For convenience, we define functions h_t by

$$\begin{aligned} h_0(x) &= r_0(x) \\ h_t(x) &= f_t(x) + r_t(x) \qquad \text{for } t = 1, 2, \ldots \end{aligned}$$

so $x_{t+1} = \arg\min_x h_{0:t}(x)$. Generally we will assume the f_t are convex, and the r_t are chosen so that $r_{0:t}$ (or $h_{0:t}$) is strongly convex for all t, e.g., $r_{0:t}(x) = \frac{1}{2\eta_t}\|x\|_2^2$ (Sections 2.3 and 4.2 review important definitions and results from convex analysis).

FTRL algorithms generalize the Follow-The-Leader (FTL) approach (Hannan, 1957; Kalai and Vempala, 2005), which selects $x_{t+1} = \arg\min_x f_{1:t}(x)$. FTL can provide sublinear regret in the case of strongly convex functions (as we will show), but for general convex functions additional regularization is needed.

Adaptive regularization can be used to construct practical algorithms that provide regret bounds that hold on all rounds T, rather than only on a single round T which is chosen in advance. The framework is also particularly suitable for analyzing AdaGrad-style algorithms that adapt their regularization or norms based on the observed data, for example those of McMahan and Streeter (2010) and Duchi et al. (2010a, 2011). This approach leads to regret bounds that depend on the actual observed sequence of functions f_t (usually via $\nabla f_t(x_t)$), rather than purely worst-case bounds. These tighter bounds translate to much better performance in practice, especially for high-dimensional but sparse problem (e.g., bag-of-words feature vectors). Examples of such algorithms are analyzed in Sections 3.4 and 3.5.

We also study Mirror Descent algorithms, for example updates like

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \nabla f_t(x_t) \cdot x + \lambda \|x\|_1 + \frac{1}{2\eta_t}\|x - x_t\|_2^2$$

where η_t is an adaptive non-increasing learning rate. This update generalizes Online Gradient Descent with a non-smooth regularization term; Mirror Descent also encompasses the use of an arbitrary Bregman divergence in place of the $\|\cdot\|_2^2$ penalty

above. We will discuss this family of algorithms at length in Section 6. In fact, Mirror Descent algorithms can be expressed as particular members of the FTRL family, though generally not the most natural ones. In particular, since the state maintained by Mirror Descent is essentially only the current feasible point x_t, we will see that Mirror Descent algorithms are forced to linearize penalties like $\lambda\|x\|_1$ from previous rounds, while the more natural FTRL algorithms can keep these terms in closed form, leading to practical advantages such as producing sparser models when L_1 regularization is used.

While we focus on online algorithms and regret bounds, the development of many of the algorithms considered rests heavily on work in general convex optimization and stochastic optimization. As a few starting points, we refer the reader to Nemirovsky and Yudin (1983) and Nesterov (2004, 2007). Going the other way, the algorithms presented here can be applied to batch optimization problems of the form

$$\arg\min_{x\in\mathbb{R}^n} \; F(x) \qquad \text{where} \qquad F(x) \equiv \sum_{t=1}^{T} f_t(x) \tag{3}$$

by running the online algorithm for one or more passes over the set of f_t and returning a suitable point (usually the last x_t or an average of past x_t). Using online-to-batch conversion techniques (e.g., Cesa-Bianchi et al. (2004), Shalev-Shwartz (2012, Chapter 5)), one can convert the regret bounds given here to convergence bounds for the batch problem. Many state-of-the-art algorithms for batch optimization over very large datasets can be analyzed in this fashion.

Outline In Section 2, we elaborate on the family of algorithms encompassed by the update of Eq. (2). We then state two regret bounds, Theorems 1 and 2, which are flexible enough to cover many known results for general and strongly convex functions; in Section 3 we use them to derive concrete bounds for many standard online algorithms.

In Section 4 we break the analysis of adaptive FTRL algorithms into three main components, which helps to modularize the arguments. In Section 4.1 we prove the *Strong FTRL Lemma* which lets us express the regret through round T as a regularization term on the comparator x^*, namely $r_{0:T}(x^*)$, plus a sum of per-round stability terms. This reduces the problem of bounding regret to that of bounding these per-round terms. In Section 4.2 we review some standard results from convex analysis, and prove lemmas that make bounding the per-round terms relatively straightforward. The general regret bounds are then proved in Section 4.3 as corollaries of these results.

Section 5 considers the special case of a composite objective, where for example $f_t(x) = \ell_t(x) + \Psi(x)$ with ℓ_t is a smooth loss on the t'th training example and Ψ is a possibly non-smooth regularizer (e.g., $\Psi(x) = \|x\|_1$). Finally, Section 6 proves the

Algorithm 2 General Template for Adaptive Linearized FTRL

Parameters: Scheme for selecting convex r_t s.t. $\forall x,\ r_t(x) \geq 0$ for $t = 0, 1, 2, \ldots$
$z \leftarrow \mathbf{0} \in \mathbb{R}^n$ *//Maintains* $g_{1:t}$
$x_1 \leftarrow \arg\min_{x \in \mathbb{R}^n}\ z \cdot x + r_0(x)$
for $t = 1, 2, \ldots$ **do**
  Select x_t, observe loss function f_t, incur loss $f_t(x_t)$
  Compute a subgradient $g_t \in \partial f_t(x_t)$
  Choose incremental convex regularizer r_t, possibly based on $g_1, \ldots, g_t$
  $z \leftarrow z + g_t$
  $x_{t+1} \leftarrow \arg\min_{x \in \mathbb{R}^n}\ z \cdot x + r_{0:t}(x)$ *//Often solved in closed form*
end for

equivalence of an arbitrary adaptive Mirror Descent algorithm and a certain FTRL algorithm, and uses this to prove regret bounds for Mirror Descent.

New Contributions The principal goal of this work is to provide a useful survey of central results in the analysis of adaptive algorithms for online convex optimization; whenever possible we provide precise references to earlier results that we re-prove or strengthen. Achieving this goal in a concise fashion requires some new results, which we summarize here.

The FTRL style of analysis is both modular and intuitive, but in previous work resulted in regret bounds that are not the tightest possible; we remedy this by introducing the Strong FTRL Lemma in Section 4.1. This also relates the FTRL analysis technique to the primal-dual style of analysis.

By analyzing both FTRL-Proximal algorithms (introduced in the next section) and Dual Averaging algorithms in a unified manner, it is much easier to contrast the strengths and weaknesses of each approach. This highlights a technical but important "off-by-one" difference between the two families in the adaptive setting, as well as an important difference when the algorithm is unconstrained (any $x_t \in \mathbb{R}^n$ is feasible).

Perhaps the most significant new contribution is given in Section 6, where we show that Mirror Descent algorithms (including adaptive algorithms for composite objectives) are in fact particular instances of the FTRL-Proximal algorithm schema, and can be analyzed using the general tools developed for the analysis of FTRL.

2. The FTRL Algorithm Family and General Regret Bounds

We begin by considering two important dimensions in the space of FTRL algorithms. First, the algorithm designer has significant flexibility in deciding whether the sum of previous loss functions is optimized exactly as $f_{1:t}(x)$ in Eq. (2), or if the true losses should be replaced by appropriate lower bounds, $\bar{f}_{1:t}(x)$, for computational efficiency.

Second, we consider whether the incremental regularizers r_t are all minimized at a fixed stationary point x_1, or are chosen so they are minimized at the current x_t. After discussing these options, we state general regret bounds.

2.1 Linearization and the Optimization of Lower Bounds

In practice, it may be infeasible to solve the optimization problem of Eq. (2), or even represent it as t becomes sufficiently large. A key point is that we can derive a wide variety of first-order algorithms by linearizing the f_t, and running the algorithm on these linear functions. Algorithm 2 gives the general scheme. For convex f_t, let x_t be defined as above, and let $g_t \in \partial f_t(x_t)$ be a subgradient (e.g., $g_t = \nabla f_t(x_t)$ for differentiable f_t). Convexity implies for any comparator x^*, $f_t(x_t) - f_t(x^*) \leq g_t \cdot (x_t - x^*)$. A key observation of Zinkevich (2003) is that if we let $\bar{f}_t(x) = g_t \cdot x$, then for any algorithm the regret against the functions $\bar{f}_t$ upper bounds the regret against the original f_t:

$$\text{Regret}(x^*, f_t) \leq \text{Regret}(x^*, \bar{f}_t).$$

Note we can construct the functions $\bar{f}_t$ on the fly (after observing x_t and f_t) and then present them to the algorithm.

Thus, rather than solving $x_{t+1} = \arg\min_x f_{1:t}(x) + r_{0:t}(x)$ on each round t, we now solve $x_{t+1} = \arg\min_x g_{1:t} \cdot x + r_{0:t}(x)$. Note that $g_{1:t} \in \mathbb{R}^n$, and we will generally choose the r_t so that $r_{0:t}(x)$ can also be represented in constant space. Thus, we have at least ensured our storage requirements stay constant even as $t \to \infty$. Further, we will usually be able to choose r_t so the optimization with $g_{1:t}$ can be solved in closed form. For example, if we take $r_{0:t}(x) = \frac{1}{2\eta}\|x\|_2^2$ then we can solve $x_{t+1} = \arg\min_x g_{1:t} \cdot x + r_{0:t}(x)$ in closed form, yielding $x_{t+1} = -\eta g_{1:t}$ (that is, this FTRL algorithm is exactly constant learning rate Online Gradient Descent).

However, we will usually state our results in terms of general f_t, since one can always simply take $f_t = \bar{f}_t$ when appropriate. In fact, an important aspect of our analysis is that it does not depend on linearization; our regret bounds hold for the the general update of Eq. (2) as well as applying to linearized variants.

More generally, we can run the algorithm on any $\bar{f}_t$ that satisfy $\bar{f}_t(x_t) - \bar{f}_t(x^*) \geq f_t(x_t) - f_t(x^*)$ for all x^* and have the regret bound achieved for the $\bar{f}$ also apply to the original f. This is generally accomplished by constructing a lower bound $\bar{f}_t$ that is tight at x_t, that is $\bar{f}_t(x) \leq f_t(x)$ for all x and further $\bar{f}_t(x_t) = f_t(x_t)$. A tight linear lower bound is always possible for convex functions, but for example if the f_t are all strongly convex, better algorithms are possible by taking $\bar{f}_t$ to be an appropriate quadratic lower bound.

A more in-depth introduction to the linearization of convex function can be found in Shalev-Shwartz (2012, Sec 2.4). We also note that the idea of replacing the loss function on each round with an appropriate lower bound ("linearization of

convex functions") is distinct from the modeling decision to replace a non-convex loss function (e.g., the zero-one loss for classification) with a convex upper bound (e.g., the hinge loss). This "convexification by surrogate loss" approach is described in detail by (Shalev-Shwartz, 2012, Sec 2.1).

2.2 Regularization in FTRL Algorithms

The term "regularization" can have multiple meanings, and so in this section we clarify the different roles regularization plays in the present work.

We refer to the functions $r_{0:t}$ as regularization functions, with r_t the incremental increase in regularization on round t (we assume $r_t(x) \geq 0$). This is the regularization in the name Follow-The-Regularized-Leader, and these r_t terms should be viewed as part of the algorithm itself—analogous (and in some cases exactly equivalent) to the learning rate schedule in an Online Gradient Descent algorithm, for example. The adaptive choice of these regularizers is the principle topic of the current work. We study two main classes of regularizers:

- In *FTRL-Centered* algorithms, each r_t (and hence $r_{0:t}$) is minimized at a fixed point, $x_1 = \arg\min_x r_0(x)$. An example is Dual Averaging (which also linearizes the losses), where $r_{0:t}$ is called the *prox-function* (Nesterov, 2009).

- In *FTRL-Proximal* algorithms, each incremental regularization function r_t is minimized by x_t, and we call such r_t incremental proximal regularizers.

When we make neither a proximal nor centered assumption on the r_t, we refer to general FTRL algorithms. Theorem 1 (below) allows us to analyze regularization choices that do not fall into either of these two categories, but the Centered and Proximal cases cover the algorithms of practical interest.

There are a number of reasons we might wish to add additional regularization terms to the objective function in the FTRL update. In many cases this is handled immediately by our general theory by grouping the additional regularization terms with either the f_t or the r_t. However, in some cases it will be advantageous to handle this additional regularization more explicitly. We study this situation in detail in Section 5.

2.3 General Regret Bounds

In this section we introduce two general regret bounds that can be used to analyze many different adaptive online algorithms. First, we introduce some additional notation and definitions.

Notation and Definitions An extended-value convex function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ satisfies

$$\psi(\theta x + (1-\theta)y) \leq \theta\psi(x) + (1-\theta)\psi(y),$$

for $\theta \in (0,1)$, and the domain of ψ is the convex set $\operatorname{dom}\psi \equiv \{x : \psi(x) < \infty\}$ (e.g., Boyd and Vandenberghe (2004, Sec. 3.1.2)); ψ is proper if $\exists x \in \mathbb{R}^n$ s.t. $\psi(x) < +\infty$ and $\forall x \in \mathbb{R}^n, \psi(x) > -\infty$. We refer to extended-value proper convex functions as simply "convex functions."

We write $\partial\psi(x)$ for the subdifferential of ψ at x; a subgradient $g \in \partial\psi(x)$ satisfies

$$\forall y \in \mathbb{R}^n,\ \psi(y) \geq \psi(x) + g \cdot (y - x).$$

The subdifferential $\partial\psi(x)$ for a convex ψ is always non-empty for $x \in \operatorname{int}(\operatorname{dom}\psi)$, and typically non-empty for any $x \in \operatorname{dom}\psi$ for the functions ψ considered in this work; $\partial\psi(x)$ is empty for $x \notin \operatorname{dom}\psi$ (Rockafellar, 1970, Thm. 23.2).

Working with extended convex functions lets us encode constraints seamlessly by using $I_{\mathcal{X}}$, the indicator function on a convex set $\mathcal{X} \subseteq \mathbb{R}^n$ given by

$$I_{\mathcal{X}}(x) = \begin{cases} 0 & x \in \mathcal{X} \\ \infty & \text{otherwise} , \end{cases} \tag{4}$$

since $I_{\mathcal{X}}$ is itself an extended convex function. Generally we assume $\mathcal{X}$ is a closed convex set. This approach makes it convenient to write $\arg\min_x$ as shorthand for $\arg\min_{x \in \mathbb{R}^n}$.

A function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is *σ-strongly convex* w.r.t. a norm $\|\cdot\|$ if for all $x, y \in \mathbb{R}^n$,

$$\forall g \in \partial\psi(x),\quad \psi(y) \geq \psi(x) + g \cdot (y - x) + \tfrac{\sigma}{2}\|y - x\|^2. \tag{5}$$

If some ψ only satisfies Eq. (5) for $x, y \in \mathcal{X}$ for a convex set $\mathcal{X}$, then the function $\psi' = \psi + I_{\mathcal{X}}$ satisfies Eq. (5) for all $x, y \in \mathbb{R}^n$, and so is strongly convex by our definition. Thus, we can work with ψ' without any need to explicitly refer to $\mathcal{X}$.

The *convex conjugate* (or Fenchel conjugate) of an arbitrary function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is

$$\psi^\star(g) \equiv \sup_x g \cdot x - \psi(x). \tag{6}$$

For a norm $\|\cdot\|$, the dual norm is given by

$$\|x\|_\star \equiv \sup_{y : \|y\| \leq 1} x \cdot y.$$

It follows from this definition that for any $x, y \in \mathbb{R}^n$, $x \cdot y \leq \|x\|\|y\|_\star$, a generalization of Hölder's inequality. We make heavy use of norms $\|\cdot\|_{(t)}$ that change as a function of the round t; the dual norm of $\|\cdot\|_{(t)}$ is $\|\cdot\|_{(t),\star}$.

Our basic assumptions correspond to the framework of Algorithm 1, which we summarize together with a few technical conditions as follows:

Setting 1 *We consider the algorithm that selects points according to Eq.* (2) *based on convex r_t that satisfy $r_t(x) \geq 0$ for $t \in \{0, 1, 2, \dots\}$, against a sequence of convex*

loss functions $f_t : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. *Further, letting* $h_{0:t} = r_{0:t} + f_{1:t}$ *we assume* $\operatorname{dom} h_{0:t}$ *is non-empty. Recalling* $x_t = \arg\min_x h_{0:t-1}(x)$*, we further assume* $\partial f_t(x_t)$ *is non-empty.*

The minor technical assumptions made here do not rule out any practical applications. We can now introduce the theorems which will be our main focus. The first will typically be applied to FTRL-Centered algorithms such as Dual Averaging:

Theorem 1 General FTRL Bound *Consider Setting 1, and suppose the* r_t *are chosen such that* $h_{0:t} + f_{t+1} = r_{0:t} + f_{1:t+1}$ *is 1-strongly-convex w.r.t. some norm* $\|\cdot\|_{(t)}$*. Then, for any* $x^* \in \mathbb{R}^n$ *and for any* $T > 0$*,*

$$\text{Regret}_T(x^*) \leq r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t-1),\star}^2.$$

Our second theorem handles proximal regularizers:

Theorem 2 FTRL-Proximal Bound *Consider Setting 1, and further suppose the* r_t *are chosen such that* $h_{0:t} = r_{0:t} + f_{1:t}$ *is 1-strongly-convex w.r.t. some norm* $\|\cdot\|_{(t)}$*, and further the* r_t *are proximal, that is* x_t *is a minimizer of* r_t*. Then, choosing any* $g_t \in \partial f_t(x_t)$ *on each round, for any* $x^* \in \mathbb{R}^n$ *and for any* $T > 0$*,*

$$\text{Regret}_T(x^*) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t),\star}^2.$$

We state these bounds in terms of strong convexity conditions on $h_{0:t}$ in order to also cover the case where the f_t are themselves strongly convex. In fact, if each f_t is strongly convex, then we can choose $r_t(x) = 0$ for all t, and Theorems 1 and 2 produce *identical* bounds (and algorithms).[1] When it is not known a priori whether the loss functions f_t are strongly convex, the r_t can be chosen adaptively to add only as much strong convexity as needed, following Bartlett et al. (2007). On the other hand, when the f_t are not strongly convex (e.g., linear), a sufficient condition for both theorems is choosing the r_t such that $r_{0:t}$ is 1-strongly-convex w.r.t. $\|\cdot\|_{(t)}$.

It is worth emphasizing the "off-by-one" difference between Theorems 1 and 2 in this case: we can choose r_t based on g_t, and when using proximal regularizers, this lets us influence the norm we use to measure g_t in the final bound (namely the $\|g_t\|_{(t),\star}^2$ term); this is not possible using Theorem 1, since we have $\|g_t\|_{(t-1),\star}^2$. This makes constructing AdaGrad-style adaptive learning rate algorithms for FTRL-Proximal easier (McMahan and Streeter, 2010), whereas with FTRL-Centered algorithms one must start with slightly more regularization. We will see this in more detail in Section 3.

1. To see this, note in Theorem 1 the norm in $\|g_t\|_{(t-1),\star}$ is determined by the strong convexity of $f_{1:t}$, and in Theorem 2 the norm in $\|g_t\|_{(t),\star}$ is again determined by the strong convexity of $f_{1:t}$.

Theorem 1 leads immediately to a bound for Dual Averaging algorithms (Nesterov, 2009), including the Regularized Dual Averaging (RDA) algorithm of Xiao (2009), and its AdaGrad variant (Duchi et al., 2011) (in fact, this statement is equivalent to Duchi et al. (2011, Prop. 2) when we assume the f_t are not strongly convex). As in these cases, Theorem 1 is usually applied to FTRL-Centered algorithms where x_1 (often the origin) is a global minimizer of $r_{0:t}$ for each t. The theorem does not require this; however, such a condition is usually necessary to bound $r_{0:T-1}(x^*)$ and hence $\text{Regret}(x^*)$ in terms of $\|x^*\|$.

Less general versions of these theorems often assume that each $r_{0:t}$ is α_t-strongly-convex with respect to a fixed norm $\|\cdot\|$. Our results include this as a special case, see Section 3 and Lemma 3 in particular.

Non-Adaptive Algorithms These theorems can also be used to analyze non-adaptive algorithms. If we choose $r_0(x)$ to be a fixed non-adaptive regularizer (perhaps chosen with knowledge of T) that is 1-strongly convex w.r.t. $\|\cdot\|$, and all $r_t(x) = 0$ for $t \geq 1$, then we have $\|x\|_{(t),\star} = \|x\|_\star$ for all t, and so both theorems provide the identical statement

$$\text{Regret}(x^*) \leq r_0(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_\star^2. \tag{7}$$

This matches Shalev-Shwartz (2012, Theorem 2.11), though we improve by a constant factor due to the use of the Strong FTRL Lemma.

2.4 Incorporating a Feasible Set

We have introduced the FTRL update as an unconstrained optimization over $x \in \mathbb{R}^n$. For many learning problems, where x_t is a vector of model parameters, this may be fine, but in other applications we need to enforce constraints. These could correspond to budget constraints, structural constraints like $\|x_t\|_2 \leq R$ or $\|x_t\|_1 \leq R_1$, a constraint that x_t is a flow on a graph, or that x_t is a probability distribution. In all of these cases, this amounts to the constraint that $x_t \in \mathcal{X}$ where $\mathcal{X}$ is a suitable convex feasible set. Further, for FTRL-Proximal algorithms a constraint like $\|x_t\|_2 \leq R$ is generally needed in order to bound $r_{0:T}(x^*)$; see Section 3.3.

Such constraints can be addressed immediately in our setting by adding the additional regularizer $I_{\mathcal{X}}$ to r_0, based on the equivalence

$$\arg\min_{x\in\mathbb{R}^n} f_{1:t}(x) + r_{0:t}(x) + I_{\mathcal{X}}(x) \quad = \quad \arg\min_{x\in\mathcal{X}} f_{1:t}(x) + r_{0:t}(x).$$

Further, if $r_{0:t}$ satisfies the conditions of Theorem 1, then so does $r_{0:t} + I_{\mathcal{X}}$. Similarly, for Theorem 2, adding $I_{\mathcal{X}}$ to r_0 will generally still produce a scheme where r_t has x_t as a minimizer, and so the theorem will still apply. We apply this technique to specific algorithms in Section 3.

Note that while the theorems still apply, the regret bounds change in an important way, since $I_{\mathcal{X}}(x^*)$ now appears in the regret bound: that is, if Theorem 1 on functions $r_0, r_1, \ldots,$ gives a bound $\text{Regret}(x^*) \leq r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t-1),\star}$, then the version constrained to select from $\mathcal{X}$ by adding $I_{\mathcal{X}}$ to r_0 has regret bound

$$\text{Regret}_T(x^*) \leq I_{\mathcal{X}}(x^*) + r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t-1),\star}.$$

This bound is *vacuous* for $x^* \notin \mathcal{X}$, but identical to the unconstrained bound for $x^* \in \mathcal{X}$. This makes sense: one can show that any online algorithm constrained to select $x_t \in \mathcal{X}$ cannot in general hope to have sublinear regret against some $x^* \notin \mathcal{X}$. Thus, if we believe some $x^* \notin \mathcal{X}$ could perform very well, incorporating the constraint $x_t \in \mathcal{X}$ is a significant sacrifice that should only be made if external considerations really require it.

3. Application to Specific Algorithms and Settings

Before proving these theorems, we apply them to a variety of specific algorithms. We will use the following lemma, which collects some facts for the sequence of incremental regularizers r_t. These claims are immediate consequences of the relevant definitions.

Lemma 3 *Consider a sequence of r_t as in Setting 1. Then, since $r_t(x) \geq 0$, we have $r_{0:t}(x) \geq r_{0:t-1}(x)$, and so $r^\star_{0:t}(x) \leq r^\star_{0:t-1}(x)$, where $r^\star_{0:t}$ is the convex-conjugate of $r_{0:t}$. If each r_t is σ_t-strongly convex w.r.t. a norm $\|\cdot\|$ for $\sigma_t \geq 0$, then, $r_{0:t}$ is $\sigma_{0:t}$-strongly convex w.r.t. $\|\cdot\|$, or equivalently, is 1-strongly-convex w.r.t. $\|x\|_{(t)} = \sqrt{\sigma_{0:t}}\|x\|$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{0:t}}}\|x\|$.*

For reasons that will become clear, it is natural to define a learning rate schedule η_t to be the inverse of the cumulative strong convexity,

$$\eta_t = \frac{1}{\sigma_{0:t}}.$$

In fact, in many cases it will be more natural to define the learning rate schedule, and infer the sequence of σ_t,

$$\sigma_t = \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}},$$

with $\sigma_0 = \frac{1}{\eta_0}$.

For simplicity, in this section we assume the loss functions have already been linearized, that is, $f_t(x) = g_t \cdot x$, unless otherwise stated. Figure 1 summarizes most of the FTRL algorithms analyzed in this section.

3.1 Constant Learning Rate Online Gradient Descent

As a warm-up, we first consider a non-adaptive algorithm, unconstrained constant learning rate Online Gradient Descent, which selects $x_1 = 0$ and thereafter

$$x_{t+1} = x_t - \eta g_t, \tag{8}$$

where the parameter $\eta > 0$ is the learning rate. Iterating this update, we see $x_{t+1} = -\eta g_{1:t}$. There is a close connection between Online Gradient Descent and FTRL, which we will use to analyze this algorithm. If we take FTRL with $r_0(x) = \frac{1}{2\eta}\|x\|_2^2$ and $r_t(x) = 0$ for $t \geq 1$, we have the update

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2, \tag{9}$$

which we can solve in closed form to see $x_{t+1} = -\eta g_{1:t}$ as well. Applying either Theorem 1 or 2 (recall they are equivalent when the regularizer is fixed) gives the bound of Eq. (7), in this case

$$\text{Regret}_T(x^*) \leq \frac{1}{2\eta}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^{T} \eta \|g_t\|_2^2, \tag{10}$$

using Lemma 3 for $\|x\|_{(t),\star} = \sqrt{\eta}\|x\|_2$. Suppose we are concerned with x^* where $\|x^*\|_2 \leq R$, the g_t satisfy $\|g_t\|_2 \leq G$, and we want to minimize regret after T' rounds. Then, choosing $\eta = \frac{R}{G\sqrt{T'}}$ minimizes Eq. (10) when $T = T'$, and we have

$$\text{Regret}_T(x^*) \leq \frac{RG}{2}\sqrt{T'} + \frac{RG}{2}\frac{T}{\sqrt{T'}},$$

or $\text{Regret}(x^*) \leq RG\sqrt{T}$ when $T = T'$. However, this bound is only $\mathcal{O}(\sqrt{T})$ when $T = \mathcal{O}(T')$. For $T \ll T'$, or $T \gg T'$ the bound is no longer interesting, and in fact the algorithm will likely perform poorly. This deficiency can be addressed via the "doubling trick", where we double T' and restart the algorithm each time T grows larger than T' (c.f., Shalev-Shwartz (2012, 2.3.1)). However, adaptively choosing the learning rate without restarting will allow us to achieve better bounds than the doubling trick (by a constant factor) with a more practically useful algorithm. We do this in Sections 3.2 and 3.3 below.

Constant Learning Rate Online Gradient Descent with a Feasible Set Above we assumed $\|x^*\|_2 \leq R$, but there is no a priori bound on the magnitude of the x_t selected by the algorithm. Following the approach of Section 2.4, we can incorporate a feasible set by taking $r_0(x) = \frac{1}{2\eta}\|x\|_2^2 + I_{\mathcal{X}}(x)$, so the update becomes

$$x_{t+1} = \arg\min_{x \in \mathbb{R}^n} g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2 + I_{\mathcal{X}}(x) = \arg\min_{x \in \mathcal{X}} g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2. \tag{11}$$

Following Shalev-Shwartz (2012, Sec. 2.6), this update is equivalent to the two-step update where we first solve the unconstrained problem and then project onto the feasible set, namely

$$u_{t+1} = \arg\min_{x \in \mathbb{R}^n} g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2$$
$$x_{t+1} = \Pi_{\mathcal{X}}(u_{t+1}) \qquad \text{where} \qquad \Pi_{\mathcal{X}}(u) \equiv \arg\min_{x \in \mathcal{X}} \|x - u\|_2.$$

Many FTRL algorithms on feasible sets can in this way be interpreted as lazy-projection algorithms, where we find (or maintain) the solution to the unconstrained problem, and then project onto the feasible set when needed.

Theorem 1 can be used to analyze the constrained algorithm of Eq. (11) in exactly the same way we analyzed Eq. (9): adding $I_{\mathcal{X}}$ does not change the strong convexity of the $\|x\|_2^2$ terms in the regularizer, and so the only difference is in the $r_{0:T}(x^*)$ term. Instead of Eq. (10), we have

$$\forall x^* \in \mathcal{X},\ \text{Regret}_T(x^*) \leq \frac{1}{2\eta}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \eta \|g_t\|_2^2,$$

where we have chosen to use the explicit $\forall x^* \in \mathcal{X}$ rather than the equivalent choice of including $I_{\mathcal{X}}(x^*)$ on the right-hand side.

Interestingly, the update of Eq. (11) is no longer equivalent to the standard projected Online Gradient Descent update $x_{t+1} = \Pi_{\mathcal{X}}(x_t - \eta g_t)$; this issue is discussed in the context of more general Mirror Descent updates in Appendix C.2. We will be able to analyze this algorithm using techniques from Section 6.

3.2 Dual Averaging

Dual Averaging is an adaptive FTRL-Centered algorithm with linearized loss functions; the adaptivity allows us to prove regret bounds that are $\mathcal{O}(\sqrt{T})$ for all T. We choose $r_t(x) = \frac{\sigma_t}{2}\|x\|_2^2$ for constants $\sigma_t \geq 0$, so $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \sqrt{\sigma_{0:t}}\|x\|_2$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{0:t}}}\|x\|_2 = \sqrt{\eta_t}\|x\|_2$, using Lemma 3. Plugging into Theorem 1 then gives

$$\forall T,\ \text{Regret}_T(x^*) \leq \frac{1}{2\eta_{T-1}}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \eta_{t-1}\|g_t\|_2^2.$$

Suppose we know $\|g_t\|_2 \leq G$, and we consider x^* where $\|x^*\|_2 \leq R$. Then, with the choice $\eta_t = \frac{R}{\sqrt{2}G\sqrt{t+1}}$, using the inequality $\sum_{t=1}^T \frac{1}{\sqrt{t}} \leq 2\sqrt{T}$, we arrive at

$$\forall T,\ \text{Regret}_T(x^*) \leq \frac{\sqrt{2}}{2}\left(R + \frac{\|x^*\|_2^2}{R}\right) G\sqrt{T}. \tag{12}$$

When in fact $\|x^*\| \leq R$, we have Regret $\leq \sqrt{2}RG\sqrt{T}$, but the bound of Eq. (12) is valid (and meaningful) for arbitrary $x^* \in \mathbb{R}^n$. Observe that on a particular round T, this bound is a factor $\sqrt{2}$ worse than the bound of $RG\sqrt{T}$ shown in Section 3.1 when the learning rate is tuned for exactly round T; this is the (small) price we pay for a bound that holds uniformly for all T.

As in the previous example, Dual Averaging can also be restricted to select from a feasible set $\mathcal{X}$ by including $I_{\mathcal{X}}$ in r_0. Additional non-smooth regularization can also be applied by adding the appropriate terms to r_0 (or any of the r_t); for example, we can add an L_1 and L_2 penalty by adding the terms $\lambda_1\|x\|_1 + \lambda_2\|x\|_2^2$. When in addition the f_t are linearized, this produces the Regularized Dual Averaging algorithm of Xiao (2009). Note that our result of $\sqrt{2}RG\sqrt{T}$ improves on the bound of $2RG\sqrt{T}$ achieved by Xiao (2009, Cor. 2(a)). We consider the case of such additional regularization terms in more detail in Section 5.

3.3 FTRL-Proximal

Suppose $\mathcal{X} \subseteq \{x \mid \|x\|_2 \leq R\}$, and we choose $r_0(x) = I_{\mathcal{X}}(x)$ and for $t > 1$, $r_t(x) = \frac{\sigma_t}{2}\|x - x_t\|_2^2$. It is worth emphasizing that unlike in the previous examples, for FTRL-Proximal the inclusion of the feasible set $\mathcal{X}$ is essential to proving regret bounds. With this constraint we have $r_{0:t}(x^*) \leq \frac{\sigma_{1:t}}{2}(2R)^2$ for any $x^* \in \mathcal{X}$, since each $x_t \in \mathcal{X}$. Without forcing $x_t \in \mathcal{X}$, however, the terms $\|x^* - x_t\|_2^2$ in $r_{0:t}(x^*)$ cannot be usefully bounded.

With these choices, $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \sqrt{\sigma_{1:t}}\|x\|_2$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{1:t}}}\|x\|_2$. Thus, applying Theorem 2, we have

$$\forall x^* \in \mathcal{X}, \quad \text{Regret}(x^*) \leq \frac{1}{2\eta_T}(2R)^2 + \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|^2, \tag{13}$$

where again $\eta_t = \frac{1}{\sigma_{1:t}}$. Choosing $\eta_t = \frac{\sqrt{2}R}{G\sqrt{t}}$ and assuming $\|x^*\| \leq R$ and $\|g_t\|_2 \leq G$,

$$\text{Regret}(x^*) \leq 2\sqrt{2}RG\sqrt{T}. \tag{14}$$

Note that we are a factor of 2 worse than the corresponding bound for Dual Averaging. However, this is essentially an artifact of loosely bounding $\|x^* - x_t\|_2^2$ by $(2R)^2$, whereas for Dual Averaging we can bound $\|x^* - 0\|_2^2$ with R^2. In practice one would hope x_t is closer to x^* than 0, and so it is reasonable to believe that the FTRL-Proximal bound will actually be tighter post-hoc in many cases. Empirical evidence also suggests FTRL-Proximal can work better in practice (McMahan, 2011).

3.4 FTRL-Proximal with Diagonal Matrix Learning Rates

We now consider an AdaGrad FTRL-Proximal algorithm which is adaptive to the observed sequence of gradients g_t, improving on the previous result. For simplicity,

first consider a one-dimensional problem. Let $r_0 = I_{\mathcal{X}}$ with $\mathcal{X} = [-R, R]$, and fix a learning rate schedule for FTRL-Proximal where

$$\eta_t = \frac{\sqrt{2}R}{\sqrt{\sum_{s=1}^{t} g_s^2}}$$

for use in Eq. (13). This gives

$$\text{Regret}(x^*) \leq 2\sqrt{2}R\sqrt{\sum_{t=1}^{T} g_t^2}, \tag{15}$$

where we have used the following lemma, which generalizes $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$:

Lemma 4 *For any non-negative real numbers $a_1, a_2, \ldots, a_n$,*

$$\sum_{i=1}^{n} \frac{a_i}{\sqrt{\sum_{j=1}^{i} a_j}} \leq 2\sqrt{\sum_{i=1}^{n} a_i} \ .$$

For a proof see Auer et al. (2002) or Streeter and McMahan (2010, Lemma 1). The bound of Eq. (15) gives us a fully adaptive version of Eq. (14): not only do we not need to know T in advance, we also do not need to know a bound on the norms of the gradients G. Rather, the bound is fully adaptive and we see, for example, that the bound only depends on rounds t where the gradient is nonzero (as one would hope). We do, however, require that R is chosen in advance; for algorithms that avoid this, see Streeter and McMahan (2012); Orabona (2013); McMahan and Abernethy (2013), and McMahan and Orabona (2014).

To arrive at an AdaGrad-style algorithm for n-dimensions we need only apply the above technique on a per-coordinate basis, namely using learning rate

$$\eta_{t,i} = \frac{\sqrt{2}R_\infty}{\sqrt{\sum_{s=1}^{t} g_{s,i}^2}}$$

for coordinate i, where we assume $\mathcal{X} \subseteq [-R_\infty, R_\infty]^n$. Streeter and McMahan (2010) take the per-coordinate approach directly; the more general approach here allows us to handle arbitrary feasible sets and L_1 or other non-smooth regularization.

We take $r_0 = I_{\mathcal{X}}$, and for $t \geq 1$ define $r_t(x) = \frac{1}{2}\|Q_t^{\frac{1}{2}}(x - x_t)\|_2^2$ where $Q_t = \text{diag}\left(\sigma_{t,i}\right)$, the diagonal matrix with entries $\sigma_{t,i} = \eta_{t,i}^{-1} - \eta_{t-1,i}^{-1}$. This Q_t is positive semi-definite, and for any such Q_t, we have that $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \|(Q_{1:t})^{\frac{1}{2}}x\|_2$, which has dual norm $\|g\|_{(t),\star} = \|(Q_{1:t})^{-\frac{1}{2}}g\|_2$. Then, plugging into Theorem 2 gives

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|(Q_{1:t})^{-\frac{1}{2}}g_t\|_2.$$

which improves on McMahan and Streeter (2010, Theorem 2) by a constant factor.

Essentially, this bound amounts to summing Eq. (15) across all n dimensions; McMahan and Streeter (2010, Cor. 9) show this bound is at least as good (and often better) than that of Eq. (14). Full matrix learning rates can be derived using a matrix generalization of Lemma 4, e.g., Duchi et al. (2011, Lemma 10); however, since this requires $\mathcal{O}(n^2)$ space and potentially $\mathcal{O}(n^2)$ time per round, in practice these algorithms are often less useful than the diagonal varieties.

It is perhaps not immediately clear that the diagonal FTRL-Proximal algorithm is easy and efficient to implement. However, taking the linear approximation to f_t, one can see $h_{1:t}(x) = g_{1:t} \cdot x + r_{1:t}(x)$ is itself just a quadratic which can be represented using two length n vectors, one to maintain the linear terms ($g_{1:t}$ plus adjustment terms) and one to maintain $\sum_{s=1}^{t} g_{s,i}^2$, from which the diagonal entries of $Q_{1:t}$ can be constructed. That is, the update simplifies to

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \; (g_{1:t} - a_{1:t}) \cdot x + \sum_{i=1}^{n} \frac{1}{2\eta_{t,i}} x_i^2 \qquad \text{where} \qquad a_t = \sigma_t x_t.$$

This update can be solved in closed-form on a per-coordinate basis when $\mathcal{X} = [-R_\infty, R_\infty]^n$. For a general feasible set, it is equivalent to a lazy-projection algorithm that first solves for the unconstrained solution and then projects it onto $\mathcal{X}$ using norm $\|(Q_{1:t})^{\frac{1}{2}} \cdot \|$ (see McMahan and Streeter (2010, Eq. 7)). Pseudo-code which also incorporates L_1 and L_2 regularization is given in McMahan et al. (2013).

3.5 AdaGrad Dual Averaging

Similar ideas can be applied to Dual Averaging (where we center each r_t at x_1), but one must use some care due to the "off-by-one" difference in the bounds. For example, for the diagonal algorithm, it is necessary to choose per-coordinate learning rates

$$\eta_t \approx \frac{R}{\sqrt{G^2 + \sum_{s=1}^{t} g_s^2}},$$

where $|g_t| \leq G$. Thus, we arrive at an algorithm that is almost (but not quite) fully adaptive in the gradients, since a modest dependence on the initial guess G of the maximum per-coordinate gradient remains in the bound. This offset appears, for example, as the δI terms added to the learning rate matrix H_t in Figure 1 of Duchi et al. (2011). We will see this issue again in Section 3.7.

3.6 Strongly Convex Functions

Suppose each loss function f_t is 1-strongly-convex w.r.t. a norm $\|\cdot\|$, and let $r_t(x) = 0$ for all t (that is, we use the Follow-The-Leader (FTL) algorithm). Define $\|x\|_{(t)} =$

Non-Adaptive FTRL Algorithms (fixed regularizer r_0, with $r_t(x) = 0$ for $t \geq 1$)

Constant Learning Rate Unprojected Online Gradient Descent

$$\begin{aligned} x_{t+1} &= x_t - \eta g_t \\ &= \arg\min_x g_{1:t} \cdot x_t + \frac{1}{2\eta}\|x\|_2^2 \\ &= -\eta g_{1:t} \end{aligned}$$

Follow-The-Leader where the f_t are 1-strongly-convex w.r.t. $\|\cdot\|$

$$x_{t+1} = \arg\min_x f_{1:t}(x)$$

Online Gradient Descent for strongly-convex functions

$$\begin{aligned} x_{t+1} &= \arg\min_x g_{1:t} \cdot x + \frac{1}{2}\sum_{s=1}^{t}\|x - x_s\|^2 \qquad \text{where } g_t \in \partial f_t(x_t) \\ &= x_t - \eta_t g_t \qquad \text{where } \eta_t = \frac{1}{t} \end{aligned}$$

Adaptive FTRL-Centered Algorithms (r_t chosen adaptively and minimized at x_1)

Unconstrained Dual Averaging (adaptive to t)

$$\begin{aligned} x_{t+1} &= \arg\min_x g_{1:t} \cdot x + \frac{1}{2\eta_t}\|x\|_2^2 \qquad \text{where } \eta_t = \frac{R}{\sqrt{2}G\sqrt{t+1}} \\ &= -\eta_t g_{1:t} \end{aligned}$$

FTRL with the entropic regularizer over the probability simplex Δ (adaptive to g_t)

$$\begin{aligned} x_{t+1} &= \arg\min_{x\in\Delta} g_{1:t} \cdot x + \frac{1}{2\eta_t}\sum_{i=1}^{n} x_i \log x_i \qquad \text{where } \eta_t = \frac{\sqrt{\log n}}{\sqrt{G_\infty^2 + \sum_{s=1}^{t}\|g_s\|_\infty^2}}, \text{ or} \\ x_{t+1,i} &= \frac{\exp(-\eta_t g_{1:t,i})}{\sum_{i=1}^{n}\exp(-\eta_t g_{1:t,i})} \qquad \text{in closed form} \end{aligned}$$

Adaptive FTRL-Proximal Algorithms (r_t chosen adaptively and minimized at x_t)

FTRL-Proximal (adaptive to t) with $\sigma_s = \eta_s^{-1} - \eta_{s-1}^{-1}$

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} g_{1:t} \cdot x + \sum_{s=1}^{t}\frac{\sigma_s}{2}\|x - x_s\|_2^2 \qquad \text{where } \eta_t = \frac{\sqrt{2}R}{G\sqrt{t}}$$

AdaGrad FTRL-Proximal (adaptive to g_t) with $\sigma_{s,i} = \eta_{s,i}^{-1} - \eta_{s-1,i}^{-1}$.

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} g_{1:t} \cdot x + \sum_{s=1}^{t}\frac{1}{2}\left\|\operatorname{diag}\left(\sigma_{s,i}^{\frac{1}{2}}\right)(x - x_s)\right\|_2^2 \qquad \text{where } \eta_{t,i} = \frac{\sqrt{2}R}{\sqrt{\sum_{s=1}^{t} g_{s,i}^2}}$$

Figure 1: Example updates for algorithms in different branches of the FTRL family.

$\sqrt{t}\|x\|$, and observe $h_{0:t}(x)$ is 1-strongly-convex w.r.t. $\|\cdot\|_{(t)}$ (by Lemma 3). Then, applying either Theorem 1 or 2 (recalling they coincide when all $r_t(x) = 0$),

$$\text{Regret}(x^*) \leq \frac{1}{2}\sum_{t=1}^{T}\|g_t\|_{(t),\star}^2 = \frac{1}{2}\sum_{t=1}^{T}\frac{1}{t}\|g_t\|^2 \leq \frac{G^2}{2}(1+\log T),$$

where we have used the inequality $\sum_{t=1}^{T} 1/t \leq 1 + \log T$ and assumed $\|g_t\| \leq G$. This recovers, e.g., Kakade and Shalev-Shwartz (2008, Cor. 1) for the the exact FTL algorithm. This algorithm requires optimizing over $f_{1:t}$ exactly, which may be computationally prohibitive.

For a 1-strongly-convex f_t with $g_t \in \partial f_t(x_t)$ we have by definition

$$f_t(x) \geq \underbrace{f_t(x_t) + g_t \cdot (x - x_t) + \frac{1}{2}\|x - x_t\|^2}_{=\bar{f}_t}.$$

Thus, we can define a $\bar{f}_t$ equal to the right-hand-side of the above inequality, so $\bar{f}_t(x) \leq f_t(x)$ and $\bar{f}_t(x_t) = f_t(x_t)$. The $\bar{f}_t$ are also 1-strongly-convex w.r.t. $\|\cdot\|$, and so running FTL on these functions produces an identical regret bound. Theorem 11 will show that the update $x_{t+1} = \arg\min_x \bar{f}_{1:t}(x)$ is equivalent to the Online Gradient Descent update

$$x_{t+1} = x_t - \frac{1}{t}g_t,$$

showing this update is essentially the Online Gradient Descent algorithm for strongly convex functions given by Hazan et al. (2007).[2]

3.7 Adaptive Dual Averaging with the Entropic Regularizer

We consider problems where the algorithm selects a probability distribution (e.g., in order to sample an action from a discrete set of n choices), that is $x_t \in \Delta_n$ with

$$\Delta_n = \left\{x \;\middle|\; \sum\nolimits_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0\right\}.$$

We assume gradients are bounded so that $\|g_t\|_\infty \leq G_\infty$, which is natural for example if each action has a cost in the range $[-G_\infty, G_\infty]$, so $g_t \cdot x$ gives the expected cost of choosing an action from the distribution x. This is the classic problem of prediction from expert advice (Vovk, 1990; Littlestone and Warmuth, 1994; Freund and Schapire, 1995; Cesa-Bianchi and Lugosi, 2006).

2. Again, the constraint to select from a fixed feasible set $\mathcal{X}$ can be added easily in either case; however, the natural way to add the constraint to the FTRL expression produces a "lazy-projection" algorithm, whereas adding the constraint to the Online Gradient Descent update produces a "greedy-projection" algorithm. This issue is discussed in some depth in Appendix C.2.

The previously introduced algorithms can be applied by enforcing the constraint $x \in \Delta_n$ by adding I_{Δ_n} to r_0, but to instantiate their bounds we can only bound $\|g_t\|_2$ by $\sqrt{n}G_\infty$ in this case, leading to bounds like $\mathcal{O}(G_\infty\sqrt{nT})$. By using a more appropriate regularizer, we can reduce the dependence on the dimension from $\sqrt{n}$ to $\sqrt{\log n}$. In particular, we use the entropic regularizer,

$$H(x) = I_\Delta(x) + \log n + \sum_{i=1}^{n} x_i \log x_i,$$

from which we define the following adaptive regularization schedule:

$$r_{0:t}(x) = \frac{1}{\eta_t} H(x) \qquad \text{where} \qquad \eta_t = \frac{\sqrt{\log n}}{\sqrt{G_\infty^2 + \sum_{s=1}^{t} \|g_s\|_\infty^2}}$$

for $t \geq 0$. Note that as in AdaGrad Dual Averaging, we make the learning rate schedule η_t a function of the observed g_t. The function H (and hence each $r_{0:t}$) is minimized by the uniform distribution $x_1 = (1/n, \ldots, 1/n)$ where $H(x) = 0$, and so these regularizers are centered at x_1. Note also that h is maximized at the corners of Δ_n (e.g., $x = (1, 0, \ldots, 0)$) where it has value $\log n$.

The entropic regularizer H is 1-strongly-convex with respect to the L_1 norm over the probability simplex $\mathcal{X}$ (e.g., Shalev-Shwartz (2012, Ex 2.5)), and it follows that $r_{0:t}$ is 1-strongly convex with respect to the norm $\|x\|_{(t)} = \frac{1}{\sqrt{\eta_t}}\|x\|_1$, and $\|g\|_{(t),\star}^2 = \eta_t \|g\|_\infty^2$. Then, applying Theorem 1, we have

$$\begin{aligned}
\text{Regret}(x^*) &\leq r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t-1),\star}^2 \\
&\leq \frac{\log n}{\eta_{T-1}} + \frac{1}{2}\sum_{t=1}^{T} \eta_{t-1}\|g_t\|_\infty^2 \\
&\leq \frac{\log n}{\eta_{T-1}} + \frac{\sqrt{\log n}}{2}\sum_{t=1}^{T} \frac{\|g_t\|_\infty^2}{\sqrt{\sum_{s=1}^{t}\|g_s\|_\infty^2}} \qquad \text{since } \forall t,\ \|g_t\|_\infty \leq G_\infty \\
&\leq 2\sqrt{\left(G_\infty^2 + \sum_{t=1}^{T-1}\|g_t\|_\infty^2\right)\log n} \qquad \text{Lemma 4 and } \|g_T\|_\infty \leq G_\infty \\
&\leq 2G_\infty\sqrt{T\log n}.
\end{aligned}$$

The last line gives an adaptive ($\forall T$) version of Shalev-Shwartz (2012, Cor. 2.14 and Cor 2.16), but the version of the bound in terms of $\|g_t\|_\infty$ may be much tighter if there are many rounds where the maximum magnitude cost is much less than G_∞. For similar adaptive algorithms, see Stoltz (2005, Thm 2.3) and Stoltz (2011, Thm 1.4, Eq. (1.22)).

4. A General Analysis Technique

In this section, we prove Theorems 1 and 2; the analysis techniques developed will also be used in subsequent sections to analyze composite objectives and Mirror Descent algorithms.

4.1 Inductive Lemmas

In this section we prove the following lemma that lets us analyze arbitrary FTRL-style algorithms:

Lemma 5 (Strong FTRL Lemma) *Let f_t be a sequence of arbitrary (possibly non-convex) loss functions, and let r_t be arbitrary non-negative regularization functions, such that $x_{t+1} = \arg\min_x h_{0:t}(x)$ is well defined, where $h_{0:t}(x) \equiv f_{1:t}(x) + r_{0:t}(x)$. Then, the algorithm that selects these x_t achieves*

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^{T} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t). \tag{16}$$

This lemma can be viewed as a stronger form of the more well-known standard FTRL Lemma (see Kalai and Vempala (2005); Hazan (2008), Hazan (2010, Lemma 1), McMahan and Streeter (2010, Lemma 3), and Shalev-Shwartz (2012, Lemma 2.3)). The strong version has three main advantages over the standard version: 1) it is essentially tight, which improves the final bounds by a constant factor, 2) it can be used to analyze adaptive FTRL-Centered algorithms in addition to FTRL-Proximal, and 3) it relates directly to the primal-dual style of analysis. For completeness, in Appendix A we present the standard version of the lemma, along with the proof of a bound analogous to Theorem 2 (but weaker by a constant factor).

The Strong FTRL Lemma bounds regret by the sum of two factors:

- **Stability** The terms in the sum over t measure how much better x_{t+1} is for the cumulative objective function $h_{0:t}$ than the point actually selected, x_t: namely $h_{0:t}(x_t) - h_{0:t}(x_{t+1})$. These per-round terms can be seen as measuring the stability of the algorithm, an online analog to the role of stability in the stochastic setting (Bousquet and Elisseeff, 2002; Rakhlin et al., 2005; Shalev-Shwartz et al., 2010).

- **Regularization** The term $r_{0:T}(x^*)$ quantifies how much regularization we have added, measured at the comparator point x^*. This captures the intuitive fact that if we could center our regularization at x^* it should not increase regret.

Adding strongly convex regularizers will increase stability (and hence decrease the cost of the stability terms), at the expense of paying a larger regularization penalty

$r_{0:T}(x^*)$. At the heart of the adaptive algorithms we study is the ability to dynamically balance these two competing goals.

The following corollary relates the above statement to the primal-dual style of analysis:

Corollary 6 *Consider the same conditions as Lemma 5, and further suppose the loss functions are linear, $f_t(x) = g_t \cdot x_t$. Then,*

$$h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) = r^\star_{0:t}(-g_{1:t}) - r^\star_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t, \tag{17}$$

which implies

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^{T} r^\star_{0:t}(-g_{1:t}) - r^\star_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t.$$

We make a few remarks before proving these results at the end of this section. Corollary 6 can easily be proved directly using the Fenchel-Young inequality. Our statement directly matches the first claim of Orabona (2013, Lemma 1), and in the non-adaptive case re-arrangement shows equivalence to Shalev-Shwartz (2007, Lemma 1) and Shalev-Shwartz (2012, Lemma 2.20); see also Kakade et al. (2012, Corollary 4). McMahan and Orabona (2014, Thm. 1) give a closely related duality result for regret and reward, and discuss several interpretations for this result, including the potential function view, the connection to Bregman divergences, and an interpretation of $r^\star$ as a benchmark target for reward.

Note, however, that Lemma 5 is strictly stronger than Corollary 6: it applies to non-convex f_t and r_t. Further, even for convex f_t, it can be more useful: for example, we can directly analyze strongly convex f_t with all $r_t(x) = 0$ using the first statement. Lemma 5 is also arguably simpler, in that it does not require the introduction of convexity or the Fenchel conjugate. We now prove the Strong FTRL Lemma:

Proof of Lemma 5 First, we bound a quantity that is essentially our regret if we had used the FTL algorithm against the functions $h_1, \dots h_T$ (for convenience, we

include a $-h_0(x^*)$ term as well):

$$\begin{aligned}
&\sum_{t=1}^{T} h_t(x_t) - h_{0:T}(x^*) \\
&\qquad = \sum_{t=1}^{T} (h_{0:t}(x_t) - h_{0:t-1}(x_t)) - h_{0:T}(x^*) \\
&\qquad \leq \sum_{t=1}^{T} (h_{0:t}(x_t) - h_{0:t-1}(x_t)) - h_{0:T}(x_{T+1}) \quad \text{Since } x_{T+1} \text{ minimizes } h_{0:T} \\
&\qquad \leq \sum_{t=1}^{T} (h_{0:t}(x_t) - h_{0:t}(x_{t+1})),
\end{aligned}$$

where the last line follows by simply re-indexing the $-h_{0:t}$ terms and dropping the the non-positive term $-h_0(x_1) = -r_0(x_1) \leq 0$. Expanding the definition of h on the left-hand-side of the above inequality gives

$$\sum_{t=1}^{T} (f_t(x_t) + r_t(x_t)) - f_{1:T}(x^*) - r_{0:T}(x^*) \leq \sum_{t=1}^{T} (h_{0:t}(x_t) - h_{0:t}(x_{t+1})).$$

Re-arranging the inequality proves the lemma. ■

We remark it is possible to make Lemma 5 an *equality* if we include the term $h_{0:T}(x_{T+1}) - h_{0:T}(x^*)$ on the RHS, since we can assume $r_0(x_1) = 0$ without loss of generality. In this case, we do not need the assumption that $x_{t+1} = \arg\min_x h_{0:t}(x)$, and so the lemma applies to an arbitrary sequence of points $x_1, \ldots, x_T$. On the other hand, if one is actually interested in the performance of the Follow-The-Leader (FTL) algorithm against the h_t (e.g., if all the r_t are uniformly zero), then running the FTL algorithm and choosing $x^* = x_{T+1}$ is particularly natural.

Proof of Corollary 6 Using the definition of the Fenchel conjugate and of x_{t+1},

$$r^\star_{0:t}(-g_{1:t}) = \max_x \ -g_{1:t} \cdot x - r_{0:t}(x) = -\big(\min_x \ g_{1:t} \cdot x + r_{0:t}(x)\big) = -h_{0:t}(x_{t+1}). \quad (18)$$

Now, observe that

$$\begin{aligned}
h_{0:t}(x_t) - r_t(x_t) &= g_{1:t} \cdot x_t + r_{0:t}(x_t) - r_t(x_t) \\
&= g_{1:t-1} \cdot x_t + r_{0:t-1}(x_t) + g_t \cdot x_t \\
&= h_{0:t-1}(x_t) + g_t \cdot x_t \\
&= -r^\star_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t,
\end{aligned}$$

where the last line uses Eq. (18) with $t \to t-1$. Combining this with Eq. (18) again ($-h_{0:t}(x_{t+1}) = r^\star_{0:t}(-g_{1:t})$) proves Eq. (17). ■

4.2 Tools from Convex Analysis

Here we highlight a few key tools from convex analysis that will be used to bound the per-round stability terms that appear in the Strong FTRL Lemma. For more background on convex analysis, see Rockafellar (1970) and Shalev-Shwartz (2007, 2012). The next result generalizes arguments found in earlier proofs for FTRL algorithms:

Lemma 7 *Let $\phi_1 : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a convex function such that $x_1 = \arg\min_x \phi_1(x)$ exists. Let ψ be a convex function such that $\phi_2(x) = \phi_1(x) + \psi(x)$ is strongly convex w.r.t. norm $\|\cdot\|$. Let $x_2 = \arg\min_x \phi_2(x)$. Then, for any $b \in \partial\psi(x_1)$, we have*

$$\|x_1 - x_2\| \leq \|b\|_\star, \tag{19}$$

and for any x',

$$\phi_2(x_1) - \phi_2(x') \leq \frac{1}{2}\|b\|_\star^2.$$

We defer the proofs of the results in this section to Appendix B. When ϕ_1 and ψ are quadratics (with ψ possibly linear) and the norm is the corresponding L_2 norm, both statements in the above lemma hold with equality. For the analysis of composite updates (Section 5), it will be useful to split the change ψ in the objective function ϕ into two components:

Corollary 8 *Let $\phi_1 : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a convex function such that $x_1 = \arg\min_x \phi_1(x)$ exists. Let ψ and Ψ be convex functions such that $\phi_2(x) = \phi_1(x) + \psi(x) + \Psi(x)$ is strongly convex w.r.t. norm $\|\cdot\|$. Let $x_2 = \arg\min_x \phi_2(x)$. Then, for any $b \in \partial\psi(x_1)$ and any x',*

$$\phi_2(x_1) - \phi_2(x') \leq \frac{1}{2}\|b\|_\star^2 + \Psi(x_1) - \Psi(x_2).$$

The concept of strong smoothness plays a key role in the proof of the above lemma, and can also be used directly in the application of Corollary 6. A function ψ is *σ-strongly-smooth* with respect to a norm $\|\cdot\|$ if it is differentiable and for all x, y we have

$$\psi(y) \leq \psi(x) + \nabla\psi(x) \cdot (y - x) + \tfrac{\sigma}{2}\|y - x\|^2. \tag{20}$$

There is a fundamental duality between strongly convex and strongly smooth functions:

Lemma 9 *Let ψ be closed and convex. Then ψ is σ-strongly convex with respect to the norm $\|\cdot\|$ if and only if $\psi^\star$ is $\frac{1}{\sigma}$-strongly smooth with respect to the dual norm $\|\cdot\|_\star$.*

For the strong convexity implies strongly smooth direction see Shalev-Shwartz (2007, Lemma 15), and for the other direction see Kakade et al. (2012, Theorem 3).

4.3 Regret Bound Proofs

In this section, we prove Theorems 1 and 2 using Lemma 5. Stating these two analyses in a common framework makes clear exactly where the "off-by-one" issue arises for FTRL-Centered, and how assuming proximal r_t resolves this issue. The key tool is Lemma 7, though for comparison we also provide a proof of Theorem 1 for linearized functions from Corollary 6 directly using strong smoothness.

General FTRL including FTRL-Centered (Proof of Theorem 1) In order to apply Lemma 5, we work to bound the stability terms in the sum in Eq. (16). Fix a particular round t. For Lemma 7 take $\phi_1(x) = h_{0:t-1}(x)$ and $\phi_2(x) = h_{0:t-1}(x) + f_t(x)$, so $x_t = \arg\min_x \phi_1(x)$, and by assumption ϕ_2 is 1-strongly-convex w.r.t. $\|\cdot\|_{(t-1)}$. Then, applying Lemma 7 to ϕ_2 (with $x' = x_{t+1}$), we have $\phi_2(x_t) - \phi_2(x_{t+1}) \leq \frac{1}{2}\|g_t\|^2_{(t-1),\star}$ for $g_t \in \partial f_t(x_t)$, and so

$$\begin{aligned} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) &= \phi_2(x_t) + r_t(x_t) - \phi_2(x_{t+1}) - r_t(x_{t+1}) - r_t(x_t) \\ &\leq \frac{1}{2}\|g_t\|^2_{(t-1),\star} \end{aligned}$$

where we have used the assumption that $r_t(x) \geq 0$ to drop the $-r_t(x_{t+1})$ term. We can now plug this bound into Lemma 5. However, we need to make one additional observation: the choice of r_T only impacts the bound by increasing $r_{0:T}(x^*)$. Further, r_T does not influence any of the points $x_1, \ldots, x_T$ selected by the algorithm. Thus, for analysis purposes, we can take $r_T(x) = 0$ without loss of generality, and hence replace $r_{0:T}(x^*)$ with $r_{0:T-1}(x^*)$ in the final bound. ■

FTRL-Proximal (Proof of Theorem 2) The key is again to bound the stability terms in the sum in Eq. (16). Fix a particular round t, and take $\phi_1(x) = f_{1:t-1}(x) + r_{0:t}(x) = h_{0:t}(x) - f_t(x)$. Since the r_t are proximal (so x_t is a global minimizer of r_t) we have $x_t = \arg\min_x \phi_1(x)$, and $x_{t+1} = \arg\min_x \phi_1(x) + f_t(x)$. Thus,

$$\begin{aligned} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) &\leq h_{0:t}(x_t) - h_{0:t}(x_{t+1}) && \text{Since } r_t(x) \geq 0 \\ &= \phi_1(x_t) + f_t(x_t) - \phi_1(x_{t+1}) - f_t(x_{t+1}) \\ &\leq \frac{1}{2}\|g_t\|^2_{(t),\star}, && (21) \end{aligned}$$

where the last line follows by applying Lemma 7 to ϕ_1 and $\phi_2(x) = \phi_1(x) + f_t(x) = h_{0:t}(x)$. Plugging into Lemma 5 completes the proof. ■

Primal-dual Analysis of General FTRL on Linearized Functions We give an alternative proof of Theorem 1 for linear functions, $f_t(x) = g_t \cdot x$, using Eq. (17). We remark that in this case $x_t = \nabla r^\star_{1:t-1}(-g_{1:t-1})$ (see Lemma 15 in Appendix B).

By Lemma 9, $r^\star_{1:t-1}$ is 1-strongly-smooth with respect to $\|\cdot\|_{(t-1),\star}$, and so

$$r^\star_{1:t-1}(-g_{1:t}) \leq r^\star_{1:t-1}(-g_{1:t-1}) - x_t \cdot g_t + \frac{1}{2}\|g_t\|^2_{(t-1),\star}, \tag{22}$$

and we can bound the per-round terms in Eq. (17) by

$$\begin{aligned} r^\star_{1:t}(-g_{1:t}) - r^\star_{1:t-1}(-g_{1:t-1}) + x_t \cdot g_t &\leq r^\star_{1:t}(-g_{1:t}) - r^\star_{1:t-1}(-g_{1:t}) + \frac{1}{2}\|g_t\|^2_{(t-1),\star} \\ &\leq \frac{1}{2}\|g_t\|^2_{(t-1),\star}, \end{aligned}$$

where we use Eq. (22) to bound $-r^\star_{1:t-1}(-g_{1:t-1}) + x_t \cdot g_t$, and then used the fact that $r^\star_{1:t-1}(-g_{1:t}) \geq r^\star_{1:t}(-g_{1:t})$ from Lemma 3. ■

5. Additional Regularization Terms and Composite Objectives

In this section, we consider generalized FTRL algorithms where we introduce an additional regularization term $\alpha_t\Psi(x)$ on each round, where Ψ is a convex function taking on only non-negative values, and the weights $\alpha_t \geq 0$ for $t \geq 1$ are non-increasing in t. We further assume Ψ and r_0 are both minimized at x_1, and w.l.o.g. $\Psi(x_1) = 0$ (as usual, additive constant terms do not impact regret). We generalize our definition of h_t to $h_0(x) = r_0(x)$ and

$$h_t(x) = g_t \cdot x + \alpha_t\Psi(x) + r_t(x), \tag{23}$$

so the FTRL update is

$$x_{t+1} = \arg\min_x h_{0:t}(x) = \arg\min_x g_{1:t} \cdot x + \alpha_{1:t}\Psi(x) + r_{0:t}(x). \tag{24}$$

In applications, generally the $g_t \cdot x_t$ terms come from the linearization of a loss ℓ_t, that is $g_t = \partial\ell_t(x_t)$. Here ℓ_t is for example a loss function measuring the prediction error on the tth training example for a model parameterized by x_t. (It is straightforward to replace $g_t \cdot x$ with $\ell_t(x)$ in this section, but for simplicity we assume linearization has been applied).

The Ψ terms often encode a non-smooth regularizer, and might be added for a variety of reasons. For example, the actual convex optimization problem we are solving may itself contain regularization terms. This is perhaps most clear in the case of applying an online algorithm to a batch problem as in Eq. (3). For example:

- An L_2 penalty $\Psi(x) = \|x\|_2^2$ might be added in order to promote generalization in a statistical setting, as in regularized empirical risk minimization.

- An L_1 penalty $\Psi(x) = \|x\|_1$ (as in the LASSO method) might be added to encourage sparse solutions and improve generalization in the high-dimensional setting ($n \gg T$).
- An indicator function might be added by taking $\Psi(x) = I_{\mathcal{X}}(x)$ to force $x \in \mathcal{X}$ where $\mathcal{X}$ is a convex set of feasible solutions.

As discussed in Section 2.4, the case of $\Psi = I_{\mathcal{X}}$ can be handled by our existing results. However, for other choices of Ψ it is generally preferable to only apply the linearization to the part of the objective where it is necessary computationally; in the L_1 case, given loss functions $\ell_t(x) + \lambda_1\|x\|_1$, we might partially linearize by taking $\bar{f}_t(x) = g_t \cdot x + \lambda_1\|x\|_1$, where $g_t \in \partial\ell_t(x_t)$. Recall that the primary motivation for linearization was to reduce the computation and storage requirements of the algorithm. Storing and optimizing over $\ell_{1:t}$ might be prohibitive; however, for common choices of Ψ and r_t, the optimization of Eq. (24) can be represented and solved efficiently (often in closed form). Thus, it is advantageous to consider such a composite representation.

Further, even in the case of a feasible set $\Psi = I_{\mathcal{X}}$, a careful consideration of if and when Ψ is linearized is critical to understanding the connection between Mirror Descent and FTRL. We will see that Mirror Descent *always* linearizes the past penalties $\alpha_{1:t-1}\Psi$, while with FTRL it is possible to avoid this additional linearization as in Eq. (24)—to make this distinction more clear, we will refer to the direct application of Eq. (24) as the Native FTRL algorithm. For $\Psi = I_{\mathcal{X}}$ this gives rise to the distinction between "lazy-projection" and "greedy-projection" algorithms, as discussed in Appendix C.2. And for $\Psi(x) = \|x\|_1$, this distinction makes Native FTRL algorithms preferable to composite-objective Mirror Descent for generating sparse models using L_1 regularization (see Section 6.2).

There are two types of regret bounds we may wish to prove in this setting, depending on whether we group the Ψ terms with the objective g_t, or with the regularizer r_t. We discuss these below.

In the objective We may view the $\alpha_t\Psi(x)$ terms as part of the objective, in that we desire a bound on regret against the functions $f_t^{\Psi}(x) \equiv g_t \cdot x + \alpha_t\Psi(x)$, that is

$$\text{Regret}(x^*, f^{\Psi}) \equiv \sum_{t=1}^{T} f_t^{\Psi}(x_t) - f_t^{\Psi}(x^*).$$

This setting is studied by Xiao (2009) and Duchi et al. (2010b, 2011), though in the less general setting where all $\alpha_t = 1$. We can directly apply Theorem 1 or Theorem 2 to the f^{Ψ} in this case, but this gives us bounds that depend on terms like $\|g_t + g_t^{(\Psi)}\|_{(t),\star}^2$ where $g_t^{(\Psi)} \in \partial(\alpha_t\Psi)(x_t)$; this is fine for $\Psi = I_{\mathcal{X}}$ since we can then always take $g_t^{(\Psi)} = 0$ since $x_t \in \mathcal{X}$, but for general Ψ this bound may be harder

to interpret. Further, adding a fixed known penalty like Ψ should intuitively make the problem no harder, and we would like to demonstrate this in our bounds.

In the regularizer We may wish to measure loss only against the functions $f_t(x) = g_t \cdot x$, that is,

$$\text{Regret}(x^*, g_t) \equiv \sum_{t=1}^{T} g_t \cdot x_t - g_t \cdot x^*,$$

even though we include the terms $\alpha_t \Psi$ in the update of Eq. (24). This approach is natural when we are only concerned with regret on the learning problem, $f_t(x) = \ell_t(x)$, but wish to add (for example) additional L_1 regularization in order to produce sparse models, as in McMahan et al. (2013).

In this case we can apply Theorem 1 to $f_t(x) \leftarrow g_t \cdot x$ and $r_t(x) \leftarrow r_t(x) + \alpha_t \Psi(x)$, noting that if the original $r_{0:t}$ is strongly convex w.r.t. $\|\cdot\|_{(t)}$, then $r_{0:t} + \alpha_{1:t}\Psi$ is as well, since Ψ is convex. However, if r_t is proximal, $r_t + \alpha_t \Psi$ generally will not be, and so a modified result is needed in place of Theorem 2. The following theorem provides this as well as a bound on $\text{Regret}(x^*, f^\Psi)$.

Theorem 10 FTRL-Proximal Bounds for Composite Objectives *Let Ψ be a non-negative convex function minimized at x_1 with $\Psi(x_1) = 0$. Let $\alpha_t \geq 0$ be a non-increasing sequence of constants. Consider Setting 1, and define h_t as in Eq. (23). Suppose the r_t are chosen such that $h_{0:t}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$, and further the r_t are proximal, that is x_t is a global minimizer of r_t.*

When we consider regret against $f_t^\Psi(x) = g_t \cdot x + \alpha_t \Psi(x)$, we have

$$\text{Regret}(x^*, f^\Psi) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t),\star}^2. \tag{25}$$

When we consider regret against only the functions $f_t(x) = g_t \cdot x$, we have

$$\text{Regret}(x^*, g_t) \leq r_{0:T}(x^*) + \alpha_{1:T}\Psi(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t),\star}^2. \tag{26}$$

Proof The proof closely follows the proof of Theorem 2 in Section 4.3, with the key difference that we use Corollary 8 in place of Lemma 7. We will use Lemma 5 to prove both claims. First, observe that the stability terms $h_{0:t}(x_t) - h_{0:t}(x_{t+1})$ depend only on h, and so we can bound them in the same way in both cases.

Take $\phi_1(x) = h_{0:t-1}(x) + r_t(x)$. Since the r_t are proximal (so x_t is a global minimizer of r_t) we have $x_t = \arg\min_x \phi_1(x)$, and $x_{t+1} = \arg\min_x \phi_2(x)$ where

$\phi_2(x) = \phi_1(x) + g_t \cdot x + \alpha_t \Psi(x) = h_{0:t}(x)$. Then, using Corollary 8 lets us replace Eq. (21) with

$$h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) \leq \frac{1}{2}\|g_t\|^2_{(t),\star} + \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}).$$

To apply Lemma 5 we sum over t. Considering only the Ψ terms, we have

$$\sum_{t=1}^{T} \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) = \alpha_1 \Psi(x_1) - \alpha_T \Psi(x_{T+1}) + \sum_{t=2}^{T} \alpha_t \Psi(x_t) - \alpha_{t-1} \Psi(x_t) \leq 0,$$

since $\Psi(x) \geq 0$, $\alpha_t \leq \alpha_{t-1}$, and $\Psi(x_1) = 0$. Thus,

$$\sum_{t=1}^{T} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) \leq \frac{1}{2} \sum_{t=1}^{T} \|g_t\|^2_{(t),\star}.$$

Using this with Lemma 5 applied to $f_t(x) \leftarrow g_t \cdot x + \alpha_t \Psi(x)$ and $r_t \leftarrow r_t$ proves Eq. (25). For Eq. (26), we apply Lemma 5 taking $f_t(x) \leftarrow g_t \cdot x$ and $r_t(x) \leftarrow \alpha_t \Psi(x) + r_t(x)$. ■

For FTRL-Centered algorithms, Theorem 1 immediately gives a bound for $\text{Regret}(x^*, g_t)$. For the $\text{Regret}(x^*, f^\Psi)$ case, we can prove a bound matching Theorem 1 using arguments analogous to the above.

6. Mirror Descent, FTRL-Proximal, and Implicit Updates

Recall Section 3.1 showed the equivalence between constant learning rate Online Gradient Descent and a fixed-regularizer FTRL algorithm. This equivalence is well-known in the case where $r_t(x) = 0$ for $t \geq 1$, that is, there is a fixed stabilizing regularizer r_0 independent of t, and further we take $\mathcal{X} = \mathbb{R}^n$ (e.g., Rakhlin (2008); Hazan (2010); Shalev-Shwartz (2012)). Observe that in this case FTRL-Centered and FTRL-Proximal coincide. In this section, we show how this equivalence extends to adaptive regularizers (equivalently, adaptive learning rates) and composite objectives. This builds on the work of McMahan (2011), but we make some crucial improvements in order to obtain an exact equivalence result for a much broader class of Mirror Descent algorithms and then use this result to derive regret bounds.[3]

3. Subsequent to this work, Sra et al. (2016) analyzed AdaDelay, an adaptive stochastic gradient descent algorithm that allows for potentially increasing learning rates, and Joulani et al. (2016) provided a more general analysis of Mirror Descent algorithms with non-monotonic regularizers in the online setting. Extending the FTRL view presented here to handle such algorithms is an interesting direction for future work.

Adaptive Mirror Descent Even in the non-adaptive case, Mirror Descent can be expressed as a variety of different updates, some of which are equivalent but some of which are not;[4] in particular, the inclusion of the feasible set constraint $I_{\mathcal{X}}$ gives rise to distinct "lazy projection" vs "greedy projection" algorithms—this issue is discussed in detail in Appendix C. To define the adaptive Mirror Descent family of algorithms we first define the Bregman divergence with respect to a convex differentiable function[5] ϕ:

$$\mathcal{B}_\phi(u, v) = \phi(u) - \big(\phi(v) + \nabla\phi(v) \cdot (u - v)\big).$$

The Bregman divergence is the difference at u between ϕ and ϕ's first-order Taylor expansion taken at v. For example, if we take $\phi(u) = \|u\|^2$, then $\mathcal{B}_\phi(u, v) = \|u - v\|^2$.

An adaptive Mirror Descent algorithm is defined by a sequence of continuously differentiable incremental regularizers $r_0, r_1, \ldots$, chosen so $r_{0:t}$ is strongly convex. From this, we define the time-indexed Bregman divergence $\mathcal{B}_{r_{0:t}}$,

$$\mathcal{B}_{r_{0:t}}(u, v) = r_{0:t}(u) - \big(r_{0:t}(v) + \nabla r_{0:t}(v) \cdot (u - v)\big).$$

The adaptive Mirror Descent update is then given by

$$\begin{aligned} \hat{x}_1 &= \arg\min_x r_0(x) \\ \hat{x}_{t+1} &= \arg\min_x \; g_t \cdot x + \alpha_t \Psi(x) + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t). \end{aligned} \tag{27}$$

We use $\hat{x}$ to distinguish this update from an FTRL update we will introduce shortly. Building on the previous section, we allow the update to include an additional regularization term $\alpha_t \Psi(x)$. As before, typically $g_t \cdot x$ should be viewed as a subgradient approximation to a loss function ℓ_t; it will become clear that a key question is to what extent Ψ is also linearized.

Mirror Descent algorithms were introduced in Nemirovsky and Yudin (1983) for the optimization of a fixed non-smooth convex function, and generalized to Bregman divergences by Beck and Teboulle (2003). Bounds for the online case appeared in Warmuth and Jagota (1997); a general treatment in the online case for composite objectives (with a non-adaptive learning rate) is given by Duchi et al. (2010b). Following this existing literature, we might term the update of Eq. (27) Adaptive Composite-Objective Online Mirror Descent; for simplicity we simply refer to Mirror Descent in this work.

4. In particular, it is common to see updates written in terms of $\nabla r^\star(\theta)$ for a strongly convex regularizer r, based on the fact that $\nabla r^\star(-\theta) = \arg\min_x \theta \cdot x + r(x)$ (see Lemma 15 in Appendix B).
5. Certain properties of Bregman divergences require ϕ to be strictly convex, but it provides a convenient notation to define $\mathcal{B}_\phi(u, v)$ for any differentiable convex ϕ.

Implicit updates For the moment, we neglect the Ψ terms and consider convex per-round losses ℓ_t. While standard Online Gradient Descent (or Mirror Descent) linearizes the ℓ_t to arrive at the update $\hat{x}_{t+1} = \arg\min_x \; g_t \cdot x_t + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t)$, we can define the alternative update

$$\hat{x}_{t+1} = \arg\min_x \; \ell_t(x) + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t), \tag{28}$$

where we avoid linearizing the loss ℓ_t. This is often referred to as an implicit update, since for general convex ℓ_t it is no longer possible to solve for $\hat{x}_{t+1}$ in closed form. The implicit update was introduced by Kivinen and Warmuth (1997), and has more recently been studied by Kulis and Bartlett (2010).

Again considering the Ψ terms, the Mirror Descent update of Eq. (27) can be viewed as a partial implicit update: if the real loss per round is $\ell_t(x) + \alpha_t \Psi(x)$, we linearize the $\ell_t(x)$ term but not the $\Psi(x)$ term, taking $f_t(x) = g_t \cdot x + \alpha_t \Psi(x)$. Generally this is done for computational reasons, as for common choices of Ψ such as $\Psi(x) = \|x\|_1$ or $\Psi(x) = I_{\mathcal{X}}(x)$, the update can still be solved in closed form (or at least in a computationally efficient manner, e.g., by projection). However, while $\alpha_t \Psi$ is handled without linearization, we shall see that echoes of the past $\alpha_{1:t-1}\Psi$ are encoded in a linearized fashion in the current state $\hat{x}_t$.

On terminology In the unprojected and non-adaptive case, the Mirror Descent update $\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_r(x, \hat{x}_t)$ is equivalent to the FTRL update $x_{t+1} = \arg\min_x g_{1:t} \cdot x + r(x)$ (see Appendix C). In fact, Shalev-Shwartz (2012, Sec. 2.6) refers to this update (with linearized losses) explicitly as Mirror Descent.

In our view, the key property that distinguishes Mirror Descent from FTRL is that for Mirror Descent, the state of the algorithm is exactly $\hat{x}_t \in \mathbb{R}^n$, the current feasible point. For FTRL on the other hand, the state is a different vector in $\mathbb{R}^n$, for example $g_{1:t}$ for Dual Averaging. The indirectness of the FTRL representation makes it more flexible, since for example multiple values of $g_{1:t}$ can all map to the same coefficient value x_t.

6.1 Mirror Descent is an FTRL-Proximal Algorithm

We will show that the Mirror Descent update of Eq. (27) can be expressed as the FTRL-Proximal update given in Figure 2. In particular, consider a Mirror Descent algorithm defined by the choice of r_t for $t \geq 0$. Then, we define the FTRL-Proximal update

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + g^{(\Psi)}_{1:t-1} \cdot x + \alpha_t \Psi(x) + r^{\mathcal{B}}_{0:t}(x) \tag{29}$$

Mirror Descent

$$\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \alpha_t \Psi(x) + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t) \tag{27}$$

Mirror Descent as FTRL-Proximal

$$\begin{aligned}\hat{x}_{t+1} &= \arg\min_x g_{1:t} \cdot x + g^{(\Psi)}_{1:t-1} \cdot x + \alpha_t \Psi(x) + r_0(x) + \sum_{s=1}^{t} \mathcal{B}_{r_s}(x, x_s) \\ &= \arg\min_x g_{1:t} \cdot x + g^{(\Psi)}_{1:t} \cdot x + r_0(x) + \sum_{s=1}^{t} \mathcal{B}_{r_s}(x, x_s)\end{aligned}$$

where $g^{(\Psi)}_s$ is a suitable subgradient from $\partial(\alpha_s \Psi)(x_{s+1})$

Figure 2: Mirror Descent as normally presented, and expressed as an equivalent FTRL-Proximal update.

for an appropriate choice $g^{(\Psi)}_t \in \partial(\alpha_t \Psi)(x_{t+1})$ (given below), where $r^{\mathcal{B}}_t$ is an incremental proximal regularizer defined in terms of r_t, namely

$$\begin{aligned}r^{\mathcal{B}}_0(x) &\equiv r_0(x) \\ r^{\mathcal{B}}_t(x) &\equiv \mathcal{B}_{r_t}(x, x_t) = r_t(x) - \big(r_t(x_t) + \nabla r_t(x_t) \cdot (x - x_t)\big) \qquad \text{for } t \geq 1.\end{aligned}$$

Note that $r^{\mathcal{B}}_t$ is indeed minimized by x_t and $r^{\mathcal{B}}_t(x_t) = 0$. We require $g^{(\Psi)}_t \in \partial(\alpha_t \Psi)(x_{t+1})$ such that

$$g_{1:t} + g^{(\Psi)}_{1:t} + \nabla r^{\mathcal{B}}_{0:t}(x_{t+1}) = 0. \tag{30}$$

The dependence of $g^{(\Psi)}_t$ on x_{t+1} is not problematic, as $g^{(\Psi)}_t$ is not necessary to compute x_{t+1} using Eq. (29). To see (inductively) that we can always find a a $g^{(\Psi)}_t$ satisfying Eq. (30), note the subdifferential of the objective of Eq. (29) at x is

$$g_{1:t} + g^{(\Psi)}_{1:t-1} + \partial(\alpha_t \Psi)(x) + \nabla r^{\mathcal{B}}_{0:t}(x). \tag{31}$$

Since x_{t+1} is a minimizer, we know 0 is a subgradient, which implies there must be a subgradient $g^{(\Psi)}_t \in \partial(\alpha_t \Psi)(x_{t+1})$ that satisfies Eq. (30). The fact we use a subgradient of Ψ at x_{t+1} rather than x_t is a consequence of the fact we are replicating the behavior of a (partial) implicit update algorithm.

Finally, note the update

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + g^{(\Psi)}_{1:t} \cdot x + r^{\mathcal{B}}_{0:t}(x) \tag{32}$$

is equivalent to Eq. (29), since Equations (30) and (31) imply 0 is in the subgradient of the objective Eq. (29) at the x_{t+1} given by Eq. (32). This update is exactly an FTRL-Proximal update on the functions $f_t(x) = (g_t + g_t^{(\Psi)}) \cdot x$.

With these definitions in place, we can now state and prove the main result of this section, namely the equivalence of the two updates given in Figure 2:

Theorem 11 *The Mirror Descent update of Eq.* (27) *and the FTRL-Proximal update of Eq.* (29) *select identical points.*

Proof The proof is by induction on the hypothesis that $\hat{x}_t = x_t$. This holds trivially for $t = 1$, so we proceed by assuming it holds for t.

First we consider the x_t selected by the FTRL-Proximal algorithm of Eq. (29). Since x_t minimizes this objective, zero must be a subgradient at x_t. Letting $g_s^{(r)} = \nabla r_s(x_s)$ and noting $\nabla r_t^{\mathcal{B}}(x) = \nabla r_t(x) - \nabla r_t(x_t)$, we have $g_{1:t-1} + g_{1:t-1}^{(\Psi)} + \nabla r_{0:t-1}(x_t) - g_{0:t-1}^{(r)} = 0$ following Eq. (31). Since $x_t = \hat{x}_t$ by induction hypothesis, we can rearrange and conclude

$$-\nabla r_{0:t-1}(\hat{x}_t) = g_{1:t-1} + g_{1:t-1}^{(\Psi)} - g_{0:t-1}^{(r)}. \tag{33}$$

For Mirror Descent, the gradient of the objective in Eq. (27) must be zero for $\hat{x}_{t+1}$, and so there exists a $\hat{g}_t^{(\Psi)} \in \partial(\alpha_t \Psi)(\hat{x}_{t+1})$ such that

$$\begin{aligned}
0 &= g_t + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) - \nabla r_{0:t}(\hat{x}_t) \\
&= g_t + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) - \nabla r_{0:t-1}(\hat{x}_t) - g_t^{(r)} && \text{IH and } \nabla r_t(x_t) = g_t^{(r)} \\
&= g_t + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) + g_{1:t-1} + g_{1:t-1}^{(\Psi)} - g_{0:t-1}^{(r)} - g_t^{(r)} && \text{Using Eq. (33)} \\
&= g_{1:t} + g_{1:t-1}^{(\Psi)} + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) - g_{0:t}^{(r)} \\
&= g_{1:t} + g_{1:t-1}^{(\Psi)} + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}^{\mathcal{B}}(\hat{x}_{t+1}).
\end{aligned}$$

The last line implies zero is a subgradient of the objective of Eq. (29) at $\hat{x}_{t+1}$, and so $\hat{x}_{t+1}$ is a minimizer. Since $r_{0:t}$ is strongly convex, this solution is unique and so $\hat{x}_{t+1} = x_{t+1}$. ■

6.2 Comparing Mirror Descent to the Native FTRL-Proximal Algorithm, and the Application to L_1 Regularization

Since we can write Mirror Descent as a particular FTRL update, we can now do a careful comparison to the direct application of Section 5 which gives the Native FTRL-Proximal algorithm. These two algorithms are given in Figure 3, expressed in a way that facilitates comparison.

Mirror Descent

$$\hat{x}_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + \quad g^{(\Psi)}_{1:t-1} \cdot x + \alpha_t \Psi(x) \quad + r^{\mathcal{B}}_{0:t}(x)$$

Native FTRL-Proximal

$$x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + \quad \alpha_{1:t} \Psi(x) \quad + r^{\mathcal{B}}_{0:t}(x)$$

$$(A) \qquad (B) \qquad (C)$$

Figure 3: Mirror Descent expressed as an FTRL-Proximal algorithm compared to the Native FTRL-Proximal algorithm.

Both algorithms use a linear approximation to the loss functions ℓ_t, as seen in column (A) of Figure 3, and the same proximal regularization terms (C). The key difference is in how the non-smooth terms Ψ are handled: Mirror Descent approximates the past $\alpha_s \Psi(x)$ terms for $s < t$ using a subgradient approximation $g^{(\Psi)}_s \cdot x$, keeping only the current $\alpha_t \Psi(x)$ term explicitly. In Native FTRL-Proximal, on the other hand, we represent the full weight of the Ψ terms exactly as $\alpha_{1:t}\Psi(x)$. That is, Mirror Descent is applying significantly more linearization than Native FTRL-Proximal.

Why does this matter? As we will see in Section 6.3, there is no difference in the regret bounds, even though intuitively avoiding unnecessary linearization should be preferable. However, there can be a substantial practical differences for some choices of Ψ. In particular, we focus on the common and practically important case of L_1 regularization, where we take $\Psi(x) = \|x\|_1$. Such regularization terms are often used to produce sparse solutions (x_t where many $x_{t,i} = 0$). Models with few non-zeros can be stored, transmitted, and evaluated much more cheaply than the corresponding dense models.

As discussed in McMahan (2011), it is precisely the explicit representation of the full $\alpha_{1:t}\|x\|_1$ terms that lets Native FTRL produce much sparser solutions when compared with the composite-objective Mirror Descent update with L_1 regularization (equivalent to the FOBOS algorithm of Duchi and Singer (2009)). This argument also applies to Regularized Dual Averaging (RDA, a Native FTRL-Centered algorithm); Xiao (2009) presents experiments showing the advantages of RDA for producing sparse solutions. In the remainder of this section, we explore the application to L_1 regularization in more detail, in order to illustrate the effect of the additional linearization of the $\|x\|_1$ terms used by Mirror Descent as compared to the Native FTRL-Proximal algorithm.

Another way to understand this distinction is the previously mentioned difference in how the two algorithms maintain state. Mirror Descent has exactly one way to

represent a zero coefficient in the ith coordinate, namely $\hat{x}_{t,i} = 0$. The FTRL representation is significantly more flexible, since many state values, say any $g_{1:t,i} \in [-\lambda, \lambda]$, can all correspond to a zero coefficient. This means that FTRL can represent both "we have lots of evidence that $x_{t,i}$ should be zero" (as $g_{1:t,i} = 0$ for example), as well as "we think $x_{t,i}$ is zero right now, but the evidence is very weak" (as $g_{1:t,i} = \lambda$ for example). This means there may be a memory cost for training FTRL, as $g_{1:t,i} \neq 0$ still needs to be stored when $x_{t,i} = 0$, but the obtained models typically provide much better sparsity-accuracy tradeoffs (McMahan, 2011; McMahan et al., 2013).

This distinction is critical even in the non-adaptive case, and so we consider the simplest possible setting: a fixed regularizer $r_0(x) = \frac{1}{2\eta}\|x\|_2^2$ (with $r_t(x) = 0$ for $t \geq 1$), and $\alpha_t \Psi(x) = \lambda \|x\|_1$ for all t. The updates of Figure 3 then simplify to:

Mirror Descent

$$x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + g^{(\Psi)}_{1:t-1} \cdot x + \lambda \|x\|_1 \quad + \frac{1}{2\eta}\|x\|_2^2 \tag{34}$$

Native FTRL

$$x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + t\lambda \|x\|_1 \quad + \frac{1}{2\eta}\|x\|_2^2. \tag{35}$$

The key point is the Native FTRL algorithm uses a much stronger explicit L_1 penalty, $\alpha_{1:t} = t\lambda$ instead of just $\alpha_t = \lambda$.

The closed-form update We can write the update of Eq. (34) as a standard Mirror Descent update (that is, as an optimization over f_t and a regularizer centered at the current x_t):

$$\begin{aligned} x_{t+1} &= \arg\min_x g_t \cdot x + \lambda \|x\|_1 + \frac{1}{2\eta}\|x - x_t\|_2^2 \\ &= \arg\min_x \left(g_t - \frac{x_t}{\eta}\right) \cdot x + \lambda \|x\|_1 + \frac{1}{2\eta}\|x\|_2^2. \end{aligned} \tag{36}$$

The above update decomposes on a per-coordinate basis. Subgradient calculations show that for constants $a > 0$, $b \in \mathbb{R}$, and $\lambda \geq 0$, we have

$$\arg\min_{x \in \mathbb{R}} b \cdot x + \lambda \|x\|_1 + \frac{a}{2}\|x\|^2 = \begin{cases} 0 & \text{when } |b| \leq \lambda \\ -\frac{1}{a}(b - \text{sign}(b)\lambda) & \text{otherwise.} \end{cases} \tag{37}$$

Thus, we can simplify Eq. (36) to

$$x_{t+1} = \begin{cases} 0 & \text{when } |g_t - \frac{x_t}{\eta}| \leq \lambda \\ x_t - \eta(g_t - \lambda) & \text{when } g_t - \frac{x_t}{\eta} > \lambda \quad \text{(implying } x_{t+1} < 0\text{)} \\ x_t - \eta(g_t + \lambda) & \text{otherwise} \quad \text{(i.e., } g_t - \frac{x_t}{\eta} < -\lambda \text{ and } x_{t+1} > 0\text{).} \end{cases}$$

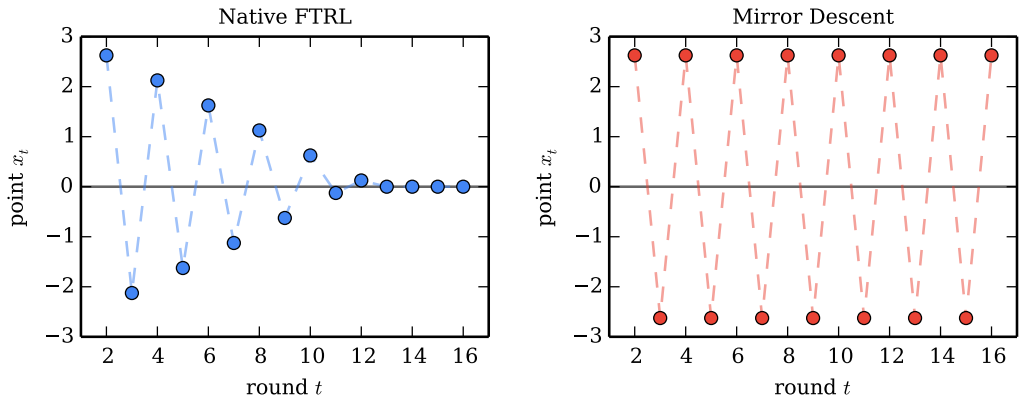

Figure 4: The points selected by Native FTRL and Mirror Descent on the one-dimensional example, using $\alpha_t \Psi(x) = \frac{1}{2}\|x\|_1$. Native FTRL quickly converges to $x^* = 0$, but Mirror Descent oscillates indefinitely.

If we choose $g_t^{(\Psi)} \in \partial\lambda\|x_{t+1}\|_1$ as

$$g_t^{(\Psi)} = \begin{cases} -\lambda & \text{when } x_{t+1} < 0 \\ \lambda & \text{when } x_{t+1} > 0\,, \\ x_t/\eta - g_t & \text{when } x_{t+1} = 0 \end{cases}$$

then Eq. (30) is satisfied, and the update becomes

$$x_{t+1} = x_t - \eta\big(g_t + g_t^{(\Psi)}\big)$$

in all cases, showing how the implicit update can be re-written in terms of a subgradient update using an appropriate subgradient approximation at the *next* point.

A One-Dimensional Example To illustrate the practical significance of the stronger explicit L_1 penalty used by Native FTRL, we compare the updates of Eq. (34) and Eq. (35) on a simple one-dimensional example. The gradients g_t satisfy $\|g_t\|_2 \le G$, and we use a feasible set of radius $R = 2G$. Both algorithms use the theory-recommended fixed learning rate $\eta = \frac{R}{G\sqrt{T}} = \frac{2}{\sqrt{T}}$ (see Section 3), against an adaptive adversary that selects gradients g_t as a function of x_t:

$$g_t = \begin{cases} -\frac{1}{2}(G+\lambda) & \text{when } t = 1 \\ -G & \text{when } t > 1 \text{ and } x_t \le 0 \\ G & \text{when } t > 1 \text{ and } x_t > 0\,. \end{cases}$$

Both algorithms select $x_1 = 0$, and since $g_1 = -\frac{1}{2}(G + \lambda)$ both algorithms select $x_2 = (G - \lambda)/\sqrt{T}$. After this, however, their behavior diverges: Mirror Descent will indefinitely oscillate between x_2 and $-x_2$ for any $\lambda < G$. On the other hand, FTRL learns that $x^* = 0$ is the optimal solution after a constant number of rounds, selecting $x_{t+1} = 0$ for any $t > \frac{G}{2\lambda} + \frac{1}{2}$. The details of this example are worked out in Appendix D

Figure 4 plots the points selected by the algorithms as a function of t, taking $G = 11$, $T = 16$, and $\lambda = 0.5$. This example clearly demonstrates that, though Mirror Descent and Native FTRL have the same regret bounds, Native FTRL is much more likely to produce sparse solutions and can also incur less actual regret.

6.3 Analysis of Mirror Descent as FTRL-Proximal

Having established the equivalence between Mirror Descent and a particular FTRL-Proximal update as given in Figure 2, we now use the general analysis techniques for FTRL developed in this work to prove regret bounds for Mirror Descent algorithm. This is accomplished by applying the Strong FTRL lemma to the FTRL-Proximal expression for Mirror Descent.

First, we observe that in the non-composite case (i.e., all $\alpha_t = 0$), then all $g_t^{(\Psi)} = 0$, and we can apply Theorem 2 directly to Eq. (29) for the loss functions $f_t(x) = g_t \cdot x$, which gives us

$$\text{Regret}(x^*, g_t) \leq r^{\mathcal{B}}_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t),\star} = \sum_{t=1}^{T} \mathcal{B}_{r_t}(x^*, x_t) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t),\star}.$$

In the case of a composite-objective (nontrivial Ψ terms, including feasible set constraints such as $I_{\mathcal{X}}$), we will arrive at the same bound, but must refine our analysis somewhat to encompass the partial implicit update of Eq. (29). This is accomplished in the following theorem:

Theorem 12 *We consider the Mirror Descent update of Eq.* (27) *under the same conditions as Theorem 10. When we consider regret against* $f_t^{\Psi}(x) = g_t \cdot x + \alpha_t \Psi(x)$, *we have*

$$\text{Regret}(x^*, f^{\Psi}) \leq r^{\mathcal{B}}_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t),\star}. \tag{38}$$

When we consider regret against only the functions $f_t(x) = g_t \cdot x$, *we have*

$$\text{Regret}(x^*, g_t) \leq r^{\mathcal{B}}_{0:T}(x^*) + \alpha_{1:T}\Psi(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t),\star}. \tag{39}$$

The bound of Eq. (38) matches Duchi et al. (2011, Prop. 3),[6] and also encompasses Theorem 2 of Duchi et al. (2010b).[7]

Proof First, by Theorem 11, this algorithm can equivalently be expressed as in Eq. (32). To simplify bookkeeping, we define

$$\bar{f}_t(x) = g_t \cdot x + \bar{\Psi}_t(x) \qquad \text{where} \qquad \bar{\Psi}_t(x) = \alpha_t \Psi(x_{t+1}) + g_t^{(\Psi)} \cdot (x - x_{t+1}),$$

Then, the update

$$x_{t+1} = \arg\min_x \bar{f}_{1:t}(x) + r^{\mathcal{B}}_{0:t}(x) \tag{40}$$

is equivalent to Eq. (32), since the objectives differ only in constant terms. Note

$$\bar{\Psi}_t(x_{t+1}) = \alpha_t \Psi(x_{t+1}) \qquad \text{and} \qquad \forall x,\ \alpha_t \Psi(x) \geq \bar{\Psi}_t(x), \tag{41}$$

where the second claim uses the convexity of $\alpha_t \Psi$.

Observe that Eq. (40) defines an FTRL-Proximal algorithm—we can imagine the $\bar{f}_t$ are computed by a black-box given f_t which solves the optimization problem of Eq. (29) in order to compute $g_t^{(\Psi)}$. Thus, we can apply the Strong FTRL Lemma (Lemma 5). Again, the key is bounding the stability terms. Using $h_t(x) = \bar{f}_t(x) + r_t^{\mathcal{B}}(x)$, we have

$$\sum_{t=1}^T h_{1:t}(x_t) - h_{1:t}(x_{t+1}) - r_t(x_t) \leq \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}),$$

using Corollary 8 as in Theorem 10.

We first consider regret against the functions $f_t^{\Psi}(x) = g_t \cdot x + \alpha_t \Psi(x)$. We can apply Lemma 5 to the functions $\bar{f}_t$, yielding

$$\text{Regret}(x^*, \bar{f}_t) \leq r^{\mathcal{B}}_{0:T}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}).$$

6. Mapping our notation to their notation, we have $f_t(x) = \ell_t(x) + \alpha_t \Psi(x) \Rightarrow \phi_t(x) = f_t(x) + \varphi(x)$ and $r_{1:t}(x) \Rightarrow \frac{1}{\eta}\psi_t(x)$. Dividing their Update (4) by η and using our notation, we arrive at exactly the update of Eq. (27). We can take $\eta = 1$ in their bound w.l.o.g.. Then, using the fact that ψ_t in their notation is $r_{1:t}$ in our notation, we have

$$\begin{aligned}
\mathcal{B}_{\psi_{t+1}}(x^*, x_{t+1}) - \mathcal{B}_{\psi_t}(x^*, x_{t+1}) &= \psi_{t+1}(x^*) - (\psi_{t+1}(x_{t+1}) + \nabla\psi_{t+1}(x_{t+1}) \cdot (x - x_{t+1})) \\
&\quad - \big(\psi_t(x^*) - (\psi_t(x_{t+1}) + \nabla\psi_t(x_{t+1}) \cdot (x - x_{t+1}))\big) \\
&= r_{t+1}(x^*) - \big(r_{t+1}(x_{t+1}) + \nabla r_{t+1}(x_{t+1}) \cdot (x - x_{t+1})\big) \\
&= \mathcal{B}_{r_{t+1}}(x^*, x_{t+1}).
\end{aligned}$$

7. We can take their $\alpha = 1$ and $\eta = 1$ w.l.o.g., and also assume our $\Psi(x_1) = 0$. Their r is our Ψ, and the implicitly take our $\alpha_t = 1$; their ψ is our r_0 (with our $r_1, \ldots, r_T$ all uniformly zero). Thus, their bound amounts (in our notation) to: Regret $\leq \mathcal{B}_{r_0}(x^*, x_1) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_\star$, matching exactly the bound of our Theorem 12 (noting $r^{\mathcal{B}}_{0:t}(x^*) = \mathcal{B}_{r_0}(x^*, x_1)$ in this case).

However, this does not immediately yield a bound on regret against the f_t^Ψ. While $\bar{f}_t(x^*) \leq f_t^\Psi(x^*)$, our actual loss $f_t^\Psi(x_t)$ could be larger than $\bar{f}_t(x_t)$. Thus, in order to bound regret against f_t^Ψ, we must add terms $f_t^\Psi(x_t) - \bar{f}_t(x_t) = \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t)$. This gives

$$\begin{aligned}
\text{Regret}(x^*, f_t^\Psi) &\leq \text{Regret}(x^*, \bar{f}_t) + \sum_{t=1}^T \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t) \\
&\leq r_{0:T}^{\mathcal{B}}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|_{(t),\star}^2 + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}) + \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t) \\
&= r_{0:T}^{\mathcal{B}}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|_{(t),\star}^2 + \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}),
\end{aligned}$$

where the equality uses $\bar{\Psi}_t(x_{t+1}) = \alpha_t \Psi(x_{t+1})$. Recalling $\sum_{t=1}^T \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) \leq 0$ from the proof of Theorem 10 completes the proof of Eq. (38).

For Eq. (39), applying Lemma 5 with $r_t \leftarrow \bar{\Psi}_t + r_t^{\mathcal{B}}$ and $f_t(x) \leftarrow g_t \cdot x$ yields

$$\text{Regret}(x^*, g_t) \leq r_{0:T}^{\mathcal{B}}(x^*) + \bar{\Psi}_{1:t}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|_{(t),\star}^2 + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}).$$

Eq. (41) implies $\bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}) \leq \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1})$, and so the sum of these terms again vanishes. Finally, observing $\bar{\Psi}_{1:t}(x^*) \leq \alpha_{1:t} \Psi(x^*)$ completes the proof. ■

7. Conclusions

Using a general and modular analysis, we have presented a unified view of a wide family of algorithms for online convex optimization that includes Dual Averaging, Mirror Descent, FTRL, and FTRL-Proximal, recovering and sometimes improving regret bounds from many earlier works. Our emphasis has been on the case of adaptive regularizers, but the results recover those for a fixed learning rate or regularizer as well.

Acknowledgments

The author thanks Daniel Golovin, Francesco Orabona, and the anonymous reviewers for useful feedback on this work.

References

Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 2002.

Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *Neural Information Processing Systems (NIPS)*, 2007.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3), 2003.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research (JMLR)*, 2, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50, 2004.

John Duchi and Yoram Singer. Efficient learning using forward-backward splitting. In *Neural Information Processing Systems (NIPS)*. 2009.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference On Learning Theory (COLT)*, 2010a.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Conference On Learning Theory (COLT)*, 2010b.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12, 2011.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, 1995.

Geoffrey J. Gordon. Regret bounds for prediction problems. In *Conference on Computational Learning Theory (COLT)*, 1999.

J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games, Volume III*, 1957.

Elad Hazan. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Conference On Learning Theory (COLT)*, 2008.

Elad Hazan. The convex optimization approach to regret minimization, 2010.

Elad Hazan. Introduction to online convex optimization, 2015.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69, December 2007.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvari. A unified modular analysis of online and stochastic optimization: Adaptivity, optimism, non-convexity. In *OPT 2016 (NIPS Workshop on Optimization)*, 2016.

Sham M. Kakade and Shai Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Neural Information Processing Systems (NIPS)*, 2008.

Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research (JMLR)*, 2012.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and Systems Sciences*, 71(3), 2005.

Jyrki Kivinen and Manfred Warmuth. Exponentiated Gradient Versus Gradient Descent for Linear Predictors. *Journal of Information and Computation*, 132, 1997.

Brian Kulis and Peter Bartlett. Implicit online learning. In *International Conference on Machine Learning (ICML)*, 2010.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2), February 1994.

H. Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

H. Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In *Neural Information Processing Systems (NIPS)*, 2013.

H. Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory (COLT)*, 2014.

H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *Conference On Learning Theory (COLT)*, 2010.

H. Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. Ad click prediction: a view from the trenches. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004.

Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical Report Technical Report 2007/76, Catholic University of Louvain, Center for Operations Research and Econometrics, 2007.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Math. Program.*, 120(1), April 2009.

Francesco Orabona. Dimension-free exponentiated gradient. In *Neural Information Processing Systems (NIPS)*, 2013.

Alexander Rakhlin. Lecture notes on online learning, 2008.

Alexander Rakhlin, Sayan Mukherjee, and Tomaso Poggio. Stability results in learning theory. *Analysis and Applications*, 2005.

Ralph T. Rockafellar. *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, 1970.

Shai Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 2012.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 2007.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research (JMLR)*, 2010.

Suvrit Sra, Adams Wei Yu, Mu Li, and Alex Smola. Adadelay: Delay adaptive distributed stochastic optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 957–965, 2016.

Gilles Stoltz. *Incomplete information and internal regret in prediction of individual sequences*. PhD thesis, Paris-Sud XI University, 2005.

Gilles Stoltz. *Contributions to the sequential prediction of arbitrary sequences: applications to the theory of repeated games and empirical studies of the performance of the aggregation of experts*. Habilitation à diriger des recherches, Université Paris-Sud, 2011.

Matthew Streeter and H. Brendan McMahan. Less regret via online conditioning. 2010. URL `http://arxiv.org/abs/1002.4862`.

Matthew Streeter and H. Brendan McMahan. No-regret algorithms for unconstrained online convex optimization. In *Neural Information Processing Systems (NIPS)*, 2012.

Volodimir G. Vovk. Aggregating strategies. In *Workshop on Computational Learning Theory*, 1990.

M. K. Warmuth and A. K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *International Symposium on Artificial Intelligence and Mathematics*, 1997.

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Neural Information Processing Systems (NIPS)*, 2009.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, 2003.

Appendix A. The Standard FTRL Lemma

The following lemma is a well-known tool for the analysis of FTRL algorithms (see Kalai and Vempala (2005); Hazan (2008), Hazan (2010, Lemma 1), and Shalev-Shwartz (2012, Lemma 2.3)):

Lemma 13 (Standard FTRL Lemma) *Let f_t be a sequence of arbitrary (possibly non-convex) loss functions, and let r_t be arbitrary non-negative regularization functions, such that $x_{t+1} = \arg\min_x h_{0:t}(x)$ is well defined (recall $h_{0:t}(x) = f_{1:t}(x) + r_{0:t}(x)$). Then, the algorithm that selects these x_t achieves*

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^{T} f_t(x_t) - f_t(x_{t+1}).$$

The proof of this lemma (e.g., McMahan and Streeter (2010, Lemma 3)) relies on showing that if one could run the Be-The-Leader algorithm by selecting $x_t = \arg\min_x f_{1:t}(x)$ (which requires peaking ahead at f_t to choose x_t), then the algorithm's regret is bounded above by zero.

However, as we see by comparing Theorem 2 and 14 (stated below), this analysis loses a factor of $1/2$ on one of the terms. The key is that being the leader is actually *strictly better* than always using the post-hoc optimal point, a fact that is not captured by the Standard FTRL Lemma. To prove the Strong FTRL Lemma, rather than first analyzing the Be-The-Leader algorithm and showing it has no regret, the key is to directly analyze the FTL algorithm (using a similar inductive argument). The proofs are also similar in that in both the basic bound is proved first for regret against the functions h_t (equivalently, the regret for FTL without regularization), and this bound is then applied to the regularized functions and re-arranged to bound regret against the f_t.

Using Lemma 13, we can prove the following weaker version of Theorem 2:

Theorem 14 Weak FTRL-Proximal Bound *Consider Setting 1, and further suppose the r_t are chosen such that $h_{0:t} = r_{0:t} + f_{1:t}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$, and further the r_t are proximal, that is x_t is a global minimizer of r_t. Then, choosing any $g_t \in \partial f_t(x_t)$ on each round, for any $x^* \in \mathbb{R}^n$,*

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^{T} \|g_t\|_{(t),\star}^2.$$

We prove Theorem 14 using strong smoothness via Lemma 7. An alternative proof that uses strong convexity directly is also possible, closely following Shalev-Shwartz (2012, Sec. 2.5.2).

Proof of Theorem 14 Applying Lemma 13, it is sufficient to consider a fixed t and upper bound $f_t(x_t) - f_t(x_{t+1})$. For this fixed t, define a helper function $\phi_1(x) = f_{1:t-1}(x) + r_{0:t}(x)$. Observe $x_t = \arg\min_x \phi_1(x)$ since x_t is a minimizer of $r_t(x)$, and by definition of the update x_t is a minimizer of $f_{1:t-1}(x) + r_{0:t-1}(x)$. Let $\phi_2(x) = \phi_1(x) + f_t(x) = h_{0:t}(x)$, so ϕ_2 is 1-strongly convex with respect to $\|\cdot\|_{(t)}$ by assumption, and $x_{t+1} = \arg\min_x \phi_2(x)$. Then, we have

$$\begin{aligned} f_t(x_t) - f_t(x_{t+1}) &\leq g_t \cdot (x_t - x_{t+1}) && \text{Convexity of } f_t \text{ and } g_t \in \partial f_t(x_t) \\ &\leq \|g_t\|_{(t),\star} \|x_t - x_{t+1}\|_{(t)} && \text{Property of dual norms} \\ &\leq \|g_t\|_{(t),\star} \|g_t\|_{(t),\star} = \|g_t\|^2_{(t),\star}. && \text{Using Eq. (19) from Lemma 7} \end{aligned}$$

■

Interestingly, it appears difficult to achieve a tight (up to constant factors) analysis of non-proximal FTRL algorithms (e.g., FTRL-Centered algorithms like Dual Averaging) using Lemma 13. The Strong FTRL Lemma, however, allowed us to accomplish this.

Appendix B. Proofs For Section 4.2

We first state a standard technical result (see Shalev-Shwartz (2007, Lemma 15)):

Lemma 15 *Let ψ be 1-strongly convex w.r.t. $\|\cdot\|$, so $\psi^\star$ is 1-strongly smooth with respect to $\|\cdot\|_\star$. Then,*

$$\|\nabla\psi^\star(z) - \nabla\psi^\star(z')\| \leq \|z - z'\|_\star, \tag{42}$$

and

$$\arg\min_x g \cdot x + \psi(x) = \nabla\psi^\star(-g). \tag{43}$$

In order to prove Lemma 7, we first prove a somewhat easier result:

Lemma 16 *Let $\phi_1 : \mathbb{R}^n \to \mathbb{R}$ be strongly convex w.r.t. norm $\|\cdot\|$, and let $x_1 = \arg\min_x \phi_1(x)$, and define $\phi_2(x) = \phi_1(x) + b \cdot x$ for $b \in \mathbb{R}^n$. Letting $x_2 = \arg\min_x \phi_2(x)$, we have*

$$\phi_2(x_1) - \phi_2(x_2) \leq \frac{1}{2}\|b\|^2_\star, \qquad \text{and} \qquad \|x_1 - x_2\| \leq \|b\|_\star.$$

Proof We have

$$-\phi_1^\star(0) = -\max_x 0 \cdot x - \phi_1(x) = \min_x \phi_1(x) = \phi_1(x_1).$$

and similarly,

$$-\phi_1^\star(-b) = -\max_x -b \cdot x - \phi_1(x) = \min_x b \cdot x + \phi_1(x) = b \cdot x_2 + \phi_1(x_2).$$

Since $x_1 = \nabla\phi_1^\star(0)$ and $\phi_1^\star$ is strongly-smooth (Lemma 9), Eq. (20) gives

$$\phi_1^\star(-b) \le \phi_1^\star(0) + x_1 \cdot (-b - 0) + \frac{1}{2}\|b\|_\star^2.$$

Combining these facts, we have

$$\begin{aligned}\phi_1(x_1) + b \cdot x_1 - \phi_1(x_2) - b \cdot x_2 &= -\phi_1^\star(0) + b \cdot x_1 + \phi_1^\star(-b) \\ &\le -\phi_1^\star(0) + b \cdot x_1 + \phi_1^\star(0) + x_1 \cdot (-b) + \frac{1}{2}\|b\|_\star^2 \\ &= \frac{1}{2}\|b\|_\star^2.\end{aligned}$$

For the second part, observe $\nabla\phi_1^\star(0) = x_1$, and $\nabla\phi_1^\star(-b) = x_2$ and so $\|x_1 - x_2\| \le \|b\|_\star$, using both parts of Lemma 15. ■

Proof of Lemma 7 We are given that $\phi_2(x) = \phi_1(x) + \psi(x)$ is 1-strongly convex w.r.t. $\|\cdot\|$. The key trick is to construct an alternative ϕ_1' that is also 1-strongly convex with respect to this same norm, but has x_1 as a minimizer. Fortunately, this is easily possible: define $\phi_1'(x) = \phi_1(x) + \psi(x) - b \cdot x$, and note ϕ_1 is 1-strongly convex w.r.t. $\|\cdot\|$ since it differs from ϕ_2 only by a linear function. Since $b \in \partial\psi(x_1)$ it follows that 0 is in $\partial(\psi(x) - b \cdot x)$ at $x = x_1$, and so $x_1 = \arg\min \phi_1'(x)$. Note $\phi_2(x) = \phi_1'(x) + b \cdot x$. Applying Lemma 16 to ϕ_1' and ϕ_2 completes the proof, noting for any x' we have $\phi_2(x_1) - \phi_2(x') \le \phi_2(x_1) - \phi_2(x_2)$. ■

Proof of Corollary 8 Let $x_2' = \arg\min_x \phi_1(x) + \psi(x)$, so by Lemma 7, we have

$$\phi_1(x_1) + \psi(x_1) - \phi_1(x_2') - \psi(x_2') \le \frac{1}{2}\|b\|_\star^2, \tag{44}$$

Then, noting $\phi_1(x_2') + \psi(x_2') \le \phi_1(x_2) + \psi(x_2)$ by definition, we have

$$\begin{aligned}\phi_2(x_1) - \phi_2(x_2) &= \phi_1(x_1) + \psi(x_1) + \Psi(x_1) - \phi_1(x_2) - \psi(x_2) - \Psi(x_2) \\ &\le \phi_1(x_1) + \psi(x_1) + \Psi(x_1) - \phi_1(x_2') - \psi(x_2') - \Psi(x_2) \\ &\le \frac{1}{2}\|b\|_\star^2 + \Psi(x_1) - \Psi(x_2). \qquad \text{Using Eq. (44).}\end{aligned}$$

Noting that $\phi_2(x_1) - \phi_2(x') \le \phi_2(x_1) - \phi_2(x_2)$ for any x' completes the proof. ■

Appendix C. Non-Adaptive Mirror Descent and Projection

Non-adaptive Mirror Descent algorithms have appeared in the literature in a variety of forms, some equivalent and some not. In this section we briefly review these connections. We first consider the unconstrained case, where the domain of the convex functions is taken to be $\mathbb{R}^n$, and there is no constraint that $x_t \in \mathcal{X}$.

Explicit	$\theta_{t+1} = \theta_t - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$	$\theta_{t+1} = \nabla R(x_t) - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$
Implicit	$x_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_R(x, x_t)$	
FTRL	$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x)$	

Figure 5: Four equivalent expressions for unconstrained Mirror Descent defined by a strongly convex regularizer R. The top-right expression is from by Beck and Teboulle (2003), while the top-left expression matches the presentation of Shalev-Shwartz (2012, Sec 2.6).

C.1 The Unconstrained Case

Figure 5 summarizes a set of equivalent expressions for the unconstrained non-adaptive Mirror Descent algorithm. Here we assume R is a strongly-convex regularizer which is differentiable on $\mathbb{R}^n$ so that the corresponding Bregman divergence $\mathcal{B}_R$ is defined. Recall from Lemma 15,

$$\nabla R^\star(-g) = \arg\min_x g \cdot x + R(x). \tag{45}$$

We now prove that these updates are equivalent:

Theorem 17 *The four updates in Figure 5 are equivalent.*

Proof It is sufficient to prove three equivalences:

- The two explicit formulations are equivalent. For the right-hand version, we have $x_t = \nabla R^\star(\theta_t) = \arg\min_x -\theta_t \cdot x + R(x)$ using Eq. (45). The optimality of x_t for this minimization implies $0 = -\theta_t + \nabla R(x_t)$, or $\nabla R(x_t) = \theta_t$.
- Explicit $\Leftrightarrow$ FTRL: Immediate from Eq. (45) and the fact that $\theta_{t+1} = -g_{1:t}$.
- Implicit $\Leftrightarrow$ FTRL: That is,

$$\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_R(x, \hat{x}_t) \qquad \text{and} \tag{46}$$

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x) \tag{47}$$

  are equivalent. The proof is by induction on the hypothesis $x_t = \hat{x}_t$. We must have from Eq. (46) and the IH that $g_t + \nabla R(\hat{x}_{t+1}) - \nabla R(x_t) = 0$, and from Eq. (47) applied to $t-1$ we must have $\nabla R(x_t) = -g_{1:t-1}$, and so $\nabla R(\hat{x}_{t+1}) = -g_{1:t}$. Then, we have the gradient of the objective of Eq. (47) at $\hat{x}_{t+1}$ is $g_{1:t} + \nabla R(\hat{x}_{t+1}) = 0$, and since the optimum of Eq. (47) is unique, we must have $\hat{x}_{t+1} = x_{t+1}$. The same general technique is used to prove the more general result for adaptive composite Mirror Descent in Theorem 11.

■

C.2 The Constrained Case: Projection onto $\mathcal{X}$

Even in the non-adaptive case (fixed R), the story is already more complicated when we constrain the algorithm to select from a convex set $\mathcal{X}$. For this section we take $R(x) = r(x) + I_{\mathcal{X}}(x)$ where r is continuously differentiable on $\operatorname{dom} I_{\mathcal{X}} = \mathcal{X}$.

In this setting, the two explicit algorithms given in the previous table are no longer equivalent. Figure 6 gives the two resulting families of updates. The classic Mirror Descent algorithm corresponds to the right-hand column, and follows the presentation of Beck and Teboulle (2003). This algorithm can be expressed as a greedy projection, and when $r(x) = \frac{1}{2\eta}\|x\|_2^2$ gives a constant learning rate version of the projected Online Gradient Descent algorithm of Zinkevich (2003). The Lazy column corresponds for example to the "Online Gradient Descent with lazy projections" algorithm (Shalev-Shwartz, 2012, Cor. 2.16).

The relationship to these projection algorithms is made explicit by the last row in the table. We define the projection operator onto $\mathcal{X}$ with respect to Bregman divergence $\mathcal{B}_r$ by

$$\Pi_{\mathcal{X}}^{r}(u) \equiv \operatorname*{arg\,min}_{x \in \mathcal{X}} \mathcal{B}_r(x, u).$$

Expanding the definition of the Bregman divergence, dropping terms independent of x since they do not influence the arg min, and replacing the explicit $x \in \mathcal{X}$ constraint with an $I_{\mathcal{X}}$ term in the objective, we have the equivalent expression

$$\Pi_{\mathcal{X}}^{r}(u) = \operatorname*{arg\,min}_{x} r(x) - \nabla r(u) \cdot x + I_{\mathcal{X}}(x). \tag{48}$$

The names Lazy and Greedy come from the manner in which the projection is used. For Lazy-Projection, the state of the algorithm is simply $g_{1:t}$ which can be updated without any need for projection; projection is applied lazily when we need to calculate x_{t+1}. For the Greedy-Projection algorithm on the other, the state of the algorithm is essentially x_t, and in particular u_{t+1} cannot be calculated without knowledge of x_t, the result of greedily applying projection on the previous round. If the g_t are really linear approximations to some f_t, however, a projection is needed on each round for both algorithms to produce x_t so $g_t \in \partial f_t(x_t)$ can be computed.

Both the Lazy and Greedy families can be analyzed (including in the more general adaptive case) using the techniques introduced in this paper. The Lazy family corresponds to the Native FTRL update of Section 5, namely

$$x_{t+1} = \operatorname*{arg\,min}_{x} g_{1:t} \cdot x + I_{\mathcal{X}}(x) + r_{0:t}(x),$$

which we encode as a single fixed non-smooth penalty $\Psi = I_{\mathcal{X}}$ which arrives on the first round: $\alpha_1 = 1$ and $\alpha_t = 0$ for $t > 1$.

	Lazy	Greedy
Explicit	$\theta_{t+1} = \theta_t - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$	$\theta_{t+1} = \nabla r(x_t) - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$
Implicit		$x_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_r(x, x_t) + I_{\mathcal{X}}(x)$
FTRL	$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x)$	$x_{t+1} = \arg\min_x \left(g_{1:t} + g^{(\Psi)}_{1:t-1}\right) \cdot x + R(x)$
Projection	$u_{t+1} = \arg\min_x g_{1:t} \cdot x + r(x)$ $x_{t+1} = \Pi^r_{\mathcal{X}}(u_{t+1})$	$u_{t+1} = \arg\min_u g_t \cdot u + \mathcal{B}_r(u, x_t)$ $= \nabla r^\star(\nabla r(x_t) - g_t)$ $x_{t+1} = \Pi^r_{\mathcal{X}}(u_{t+1})$

Figure 6: The Lazy and Greedy families of Mirror Descent algorithms, defined via $R(x) = r(x) + I_{\mathcal{X}}(x)$, where r is a differentiable strongly-convex regularizer. These families are not equivalent, but the different updates in each column are equivalent.

The Greedy-Projection Mirror Descent algorithms, on the other hand, can be thought of us receiving loss functions $g_t \cdot x + I_{\mathcal{X}}(x)$ on each round: that is, we have $\alpha_t = 1$ for all t. This family is analyzed using the techniques from Section 6. In this setting, embedding $I_{\mathcal{X}}(x)$ inside R can be seen as a convenience for defining $\nabla R^\star$,

$$\nabla R^\star(-g) = \arg\min_x g \cdot x + r(x) + I_{\mathcal{X}}(x). \tag{49}$$

We have the following equivalence results:

Theorem 18 *The Lazy-Explicit, Lazy-FTRL, and Lazy-Projection updates from the left column of Figure 6 are equivalent.*

Proof First, we show Lazy-Explicit is equivalent to Lazy-FTRL. Iterating the definition of θ_{t+1} in the explicit version gives $\theta_{t+1} = -g_{1:t}$, and so the second line in the update becomes exactly $x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x)$.

Next, we show that Lazy-Projection is equivalent to the Lazy-Explicit update. Optimality conditions for the minimization that defines u_{t+1} imply $\nabla r(u_{t+1}) = -g_{1:t}$. Then, the second equation in the Lazy-Projection update becomes

$$\begin{aligned} x_{t+1} = \Pi^r_{\mathcal{X}}(u_{t+1}) &= \arg\min_x r(x) - \nabla r(u_{t+1}) \cdot x + I_{\mathcal{X}}(x) && \text{Using Eq. (48).} \\ &= \arg\min_x g_{1:t} \cdot x + r(x) + I_{\mathcal{X}}(x), && \text{Since } \nabla r(u_{t+1}) = -g_{1:t}. \end{aligned}$$

which is exactly the Lazy-FTRL update (recalling $R(x) = r(x) + I_{\mathcal{X}}(x)$). ■

Theorem 19 *The Explicit, Implicit, FTRL, and Projected updates in the "Greedy" column of Figure 6 are equivalent.*

Proof We prove the result via the following chain of equivalences:

- Greedy-Explicit ⇔ Greedy-Implicit (c.f. Beck and Teboulle (2003, Prop 3.2)). We again use $\hat{x}$ for the points selected by the implicit version,

$$\begin{aligned}\hat{x}_{t+1} &= \arg\min_x g_t \cdot x + \mathcal{B}_r(x, x_t) + I_{\mathcal{X}}(x) \\ &= \arg\min_x g_t \cdot x + r(x) - \nabla r(x_t) \cdot x + I_{\mathcal{X}}(x),\end{aligned}$$

  where we have dropped terms independent of x in the arg min. On the other hand, plugging in the definition of θ_{t+1}, the explicit update is

$$x_{t+1} = \arg\min_x -(\nabla r(x_t) - g_t) \cdot x + r(x) + I_{\mathcal{X}}(x), \tag{50}$$

  which is equivalent.

- Greedy-Implicit ⇔ Greedy-FTRL: This is a special case of Theorem 11, taking $r_0 \leftarrow r + I_{\mathcal{X}}$, $r_t(x) = r_t^{\mathcal{B}}(x) = 0$ for $t \geq 1$, and $\alpha_t \Psi(x) = I_{\mathcal{X}}(x)$ for $t \geq 1$.

- When $I_{\mathcal{X}} = I_{\mathcal{X}}$, Projection is equivalent to the Greedy-Explicit expression. First, note we can re-write the Greedy-Projection update as

$$\begin{aligned}u_{t+1} &= \arg\min_u -(\nabla r(x_t) - g_t) \cdot u + r(u) \\ x_{t+1} &= \arg\min_{x \in \mathcal{X}} \mathcal{B}_r(x, u_{t+1}).\end{aligned}$$

  Optimality conditions for the first expression imply $\nabla r(u_{t+1}) = \nabla r(x_t) - g_t$. Then, the second update becomes

$$\begin{aligned}x_{t+1} &= \Pi_{\mathcal{X}}^r(u_{t+1}) \\ &= \arg\min_x r(x) - \nabla r(u_{t+1}) \cdot x + I_{\mathcal{X}}(x) && \text{Using Eq. (48).} \\ &= \arg\min_x r(x) - (\nabla r(x_t) - g_t) \cdot x + I_{\mathcal{X}}(x), && \text{Since } \nabla r(u_{t+1}) = \nabla r(x_t) - g_t.\end{aligned}$$

  which is equivalent to the Greedy-Explicit update, e.g., Eq. (50). ■

Appendix D. Details for the One-Dimensional L_1 Example

In this section we provide details for the one-dimensional example presented in Section 6.2. Suppose gradients g_t satisfy $\|g_t\|_2 \leq G$, and we use a feasible set of radius $R = 2G$, so the theory-recommended fixed learning rate is $\eta = \frac{R}{G\sqrt{T}} = \frac{2}{\sqrt{T}}$ (see Section 3).

We first consider the behavior of Mirror Descent: we construct the example so that the algorithm oscillates between two points, $\hat{x}$ and $-\hat{x}$ (allowing the possibility that $\hat{x} = -\hat{x} = 0$). Given alternating gradients of $+G$ and $-G$, in such an oscillation the distance one update takes us must be $\eta(G - \lambda)$, assuming $\lambda < G$. Thus, we can cause the algorithm to oscillate between $\hat{x} = (G - \lambda)/\sqrt{T}$ and $-\hat{x}$. We assume an initial $g_1 = -\frac{1}{2}(G + \lambda)$, which gives us $x_2 = \hat{x}$ for both Mirror Descent and FTRL when $x_1 = 0$.

This construction implies that for any constant L_1 penalty $\lambda < G$, Mirror Descent will never learn the optimal solution $x^* = 0$ (note that after the first round, we can view the g_t as being for example the subgradients of $f_t(x) = G\|x\|_1$). The points x_t selected by Mirror Descent, the gradients, and the subgradients of the L_1 penalty are given by the following table:

t	1	2	3	4	5	$\cdots$
g_t	g_1	G	$-G$	G	$-G$	$\cdots$
x_t	0	$\hat{x}$	$-\hat{x}$	$\hat{x}$	$-\hat{x}$	$\cdots$
$g_t^{(\Psi)}$	λ	$-\lambda$	λ	$-\lambda$	λ	$\cdots$

While we have worked from the standard Mirror Descent update, Eq. (36), it is instructive to verify the FTRL-Proximal representation is indeed equivalent. For example, using the values from the table, for x_5 we have

$$\begin{aligned} x_5 &= \arg\min_x g_{1:4} \cdot x + g_{1:3}^{(\Psi)} \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x\|_2^2 \\ &= \arg\min_x (g_1 + G) \cdot x + \lambda \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x\|_2^2 = -\frac{G - \lambda}{\sqrt{T}} = -\hat{x}, \end{aligned}$$

where we solve the argmin by applying Eq. (37) with $b = g_1 + G + \lambda$.

Now, contrast this with the FTRL update of Eq. (35); we can solve this update in closed form using Eq. (37). First, note that FTRL will not oscillate in the same way, unless $\lambda = 0$. We have that $x_{t+1} = 0$ whenever $|g_{1:t}| < t\lambda$. Note that $g_{1:t}$ oscillates between $g_{1:t} = g_1 = -\frac{1}{2}(G + \lambda)$ on odd rounds t, and $g_{1:t} = g_1 + G = \frac{1}{2}G - \frac{1}{2}\lambda$ on even rounds. Since the magnitude of $g_{1:t}$ is larger on odd rounds, if we have $\frac{1}{2}(G+\lambda) \leq t\lambda$ then x_{t+1} will always be zero; re-arranging, this amounts to $\lambda \geq \frac{G}{2t-1}$. Thus, as with Mirror Descent, we need $\lambda \geq G$ to have $x_2 = 0$ (plugging in $t = 1$) but on subsequent rounds a *much* smaller λ is sufficient to produce sparsity. In the extreme case, taking $\lambda = G/(2T - 1)$ is sufficient to ensure $x_T = 0$, whereas we need a λ value almost $2T$ times larger in order to get $x_T = 0$ from Mirror Descent.