Analysis Techniques for Adaptive Online Learning

H. Brendan McMahan
Google, Inc.
mcmahan@google.com

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. 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 an arbitrary adaptive Mirror Descent algorithm and a corresponding FTRL update, which allows us to analyze any Mirror Descent algorithm 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.

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 predictor $x_t \in \mathbb{R}^n$, 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

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

the difference between the algorithm's loss and the loss of a fixed predictor x^*, potentially chosen with full knowledge of the sequence of f_t. 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, it is often preferable to simply bound $\text{Regret}(x^*)$ by a function of $\|x^*\|$. Online algorithms with good regret bounds can be used for a wide variety of prediction and learning tasks (Cesa-Bianchi and Lugosi, 2006, Shalev-Shwartz, 2012).

Low-regret algorithms for online convex optimization can immediately be applied to online learning problems. 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 a 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 $f_t(x) = \ell(\sigma(a_t \cdot x), y_t)$; straightforward calculations show 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 (Shalev-Shwartz, 2007, Shalev-Shwartz and Singer, 2007, Rakhlin, 2008, McMahan and Streeter, 2010, McMahan, 2011). Hazan (2010) and Shalev-Shwartz (2012) provide a comprehensive survey of analysis techniques for non-adaptive members of this algorithm family, where the regularizer is typically fixed for all rounds and chosen with knowledge of T. In this survey, we allow the regularizer to change adaptively over the course of an unknown-horizon game. Given a sequence $r_0, r_1, r_2, \ldots$ of incremental regularizers, we consider the algorithm that plays $x_1 \in \arg\min_x r_0(x)$, and thereafter plays

$$x_{t+1} = \arg\min_x f_{1:t}(x) + r_{0:t}(x), \tag{1}$$

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 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 where $h_0(x) = r_0(x)$ and $h_t(x) = f_t(x) + r_t(x)$ for $t \geq 1$, 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., $r_{0:t}(x) = \frac{1}{2\eta_t}\|x\|_2^2$; see Section 3.2 for a review of important definitions and results from convex analysis). The name FTRL comes from a style of analysis that considers the regret of the Be-The-Leader algorithm (playing $x_t = \arg\min_x f_{1:t}(x)$) and then of Follow-The-Leader (playing $x_t = \arg\min_x f_{1:t-1}(x)$) (Kalai and Vempala, 2005).

In addition to providing regret bounds for all rounds t, this framework is also particularly suitable for analyzing algorithms that adapt their regularization or norms based on the observed data, for example those of McMahan and Streeter (2010) and Duchi et al. (2011).[1]

Outline In Section 2, we elaborate on the family of algorithms encompassed by update Eq. (1). We then state two very general regret bounds, Theorems 1 and 2. While these results are not new, they are stated in enough generality to cover many known results for general and strongly convex functions; in Section 2.2 we use them to derive more concrete bounds for many standard online algorithms. Section 3 then proves these theorems using a modular FTRL approach: Lemma 6 (the "Strong FTRL Lemma") reduces proving bounds to a per-round quantity, which is in turn bounded using some tools from convex analysis encapsulated as Lemma 8. Section 4 considers the special case of a composite objective, where $f_t(x) = \ell_t(x) + \Psi(x)$, where for example ℓ_t is a smooth loss and Ψ is a possibly non-smooth regularizer. Finally, Section 5 proves the equivalence of an arbitrary adaptive Mirror Descent algorithm and a certain FTRL-Proximal algorithm, and uses this to prove regret bounds for Mirror Descent.

Summary of Contributions A principal goal of this work is to provide a useful summary 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.

[1] This work first appeared simultaneously with McMahan and Streeter (2010) as Duchi et al. (2010a).

Algorithm 1 General Scheme for Linearized FTRL

```
Parameters: Initial regularization function $r_0$, scheme for choosing $r_t$.
$z \leftarrow \mathbf{0} \in \mathbb{R}^n$ // Maintains $g_{1:t}$
$x_1 \leftarrow \arg\min_x \; z \cdot x + r_0(x)$
for $t = 1, 2, \ldots$ do
    Play $x_t$, observe loss function $f_t$, incur loss $f_t(x_t)$
    Compute sub-gradient $g_t \in \partial f_t(x_t)$
    Choose $r_t$, possibly using $x_t$ and $g_t$
    $z \leftarrow z + g_t$
    $x_{t+1} \leftarrow \arg\min_x \; z \cdot x + r_{1:t}(x)$
end for
```

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 3.1. This also relates the FTRL analysis technique to a certain form of primal-dual analysis.

By analyzing both FTRL-Proximal algorithms (introduced in the next section) and Dual Averaging 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 player is unconstrained (any $x_t \in \mathbb{R}^n$ is feasible).

Perhaps the most significant new contribution is given in Section 5, where we show that *all* Mirror Descent algorithms (including adaptive algorithms for composite objectives) are in fact particular instances of the FTRL-Proximal algorithm schema, and can be analyzed in a straightforward manner using the general tools developed for the analysis of FTRL. Thus, we present a unified analysis of both FTRL and Mirror Descent in the most general setting.

2 FTRL: Regret Bounds and Applications

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. (1), 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 minimized at a fixed stationary point (e.g., the origin), or are chosen so they are minimized at the current x_t. After discussing these options, we state general regret bounds and apply them to specific algorithms.

Linearization and the Optimization of Lower Bounds In practice, it may be computationally infeasible to solve the optimization problem of Eq. (1). 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 1 gives the general scheme. For convex differentiable f_t, let x_t be defined as above, and let $g_t = \triangledown f_t(x_t)$. Then, a key observation of Zinkevich (2003) was that convexity implies for any comparator x^*, $f_t(x_t) - f_t(x^*) \leq g_t \cdot (x_t - x^*)$. 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. 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, resulting in a much easier to compute update Eq. (1). For example, if we take $r_{0:t}(x) = \frac{1}{2\eta}\|x\|_2^2$ then we can solve Eq. (1) in closed form, yielding $x_{t+1} = -\eta g_{1:t}$ (that is, this FTRL algorithm is exactly constant step size online gradient descent). However, whenever possible we will state our results in terms of general f_t, since one can always simply take $f_t = \bar{f}_t$ when appropriate.

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 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 bounds are possible by taking $\bar{f}_t$ to be an appropriate quadratic lower bound.

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. (1) as well as immediately applying to linearized variants.

Prox-Functions, Regularization, and the FTRL-Proximal Algorithm We refer to the functions $r_{0:t}$ as regularization functions, with r_t the incremental increase in regularization on round t (generally we will assume $r_t(x) \geq 0$). This is regularization in the sense of Follow-The-Regularized-Leader, and these r_t terms should be viewed as part of the algorithm itself. In Dual Averaging, $r_{0:t}$ is called the *prox-function* (Nesterov, 2009), and is generally assumed to be minimized at a fixed constant point, without loss of generality the origin. We use the term Dual Averaging to refer specifically to this case, though more typically Dual Averaging also implies linearizing the f_t (we will say more about linearization in Section 2.2).

We will refer to the algorithm as *FTRL-Proximal* when each incremental regularization function r_t is globally minimized by x_t, and call such r_t incremental proximal regularizers. When we make neither a proximal nor origin-centered assumption on the r_t, we refer to general FTRL algorithms. Using proximal regularizers will prove useful in the analysis, as in addition to $x_t = \arg\min_x f_{1:t-1}(x) + r_{0:t-1}(x)$ by Eq. (1), this choice of r_t ensures

$$x_t = \arg\min_x f_{1:t-1}(x) + r_{0:t}(x) \tag{2}$$

as well. This means we can view both x_t and x_{t+1} as being defined in terms of minimization with respect to the regularizer $r_{0:t}(x)$, which simplifies the analysis and to some extent strengthens the results.

The actual convex optimization problem we are solving may itself contain regularization terms, this time in the usual machine learning sense of the word. For example, we might have $f_t(x) = \ell_t(x) + \lambda_1\|x\|_1$, where the $\lambda_1\|x\|_1$ is an L_1 regularization term as in the LASSO method, and ℓ_t measures the loss on the tth training example. The algorithms here handle this seamlessly; we note only that it is generally preferable to only apply the linearization to the part of the objective where it is necessary computationally; in this case, for example, we would take $\bar{f}_t(x) = g_t \cdot x + \lambda_1\|x\|$, where $g_t \in \partial\ell_t(x_t)$. We consider such composite-objectives explicitly in Section 4, as well as the option to view the $\lambda_1\|x\|_1$ terms as part of the r_t. Note that whether we treat the non-smooth terms as part of r_t or f_t does not change the algorithm, but the proofs change slightly, as does the meaning of the regret bound.

FTRL-Proximal algorithms are close relatives of online gradient descent and online Mirror Descent; however, it is exactly this ability to encode the explicit L_1 penalty (or some other non-smooth regularizer) in the update that gives them a significant advantage. In fact, we show in Section 5 that all Mirror Descent algorithms can be viewed (and analyzed) exactly as members of the FTRL-Proximal family; however, at least in the case of composite objectives, more natural and generally preferable FTRL-Proximal algorithms exist (namely, the ones that never linearize the non-smooth penalties).

2.1 Analysis Techniques and General Regret Bounds

We break the analysis of adaptive FTRL algorithms into three main components, which helps to modularize the arguments. In Section 3.1 we provide two inductive lemmas that 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 terms. Generally, this reduces the problem of bounding Regret to that of bounding these per-round terms. The first of these results is often referred to as the *FTRL Lemma*; we term the second the *Strong FTRL Lemma*, but for linear functions it is also closely connected to the primal-dual analysis of online algorithms. In Section 3.2 we review some standard results from convex analysis, and prove lemmas that make bounding the per-round terms straightforward. The general regret bounds are then proved in Section 3.3 as straightforward corollaries of these results. Before stating the main theorems, we introduce some additional notation and definitions.

Notation and Definitions We consider extended convex functions $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, where the domain of ψ is the set $\{x : \psi(x) < \infty\}$. We write $\partial\psi(x)$ for the subdifferential of f at x. A subgradient $g \in \partial\psi(x)$ satisfies $\psi(y) \geq \psi(x) + g \cdot (y - x)$ for all y. A function $\psi : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ is *σ-strongly convex* w.r.t. a norm $\|\cdot\|$ on $\mathcal{X}$ if for all $x, y \in \mathcal{X}$ and any $g \in \partial\psi(x)$, we have

$$\psi(y) \geq \psi(x) + g \cdot (y - x) + \tfrac{\sigma}{2}\|y - x\|^2. \tag{3}$$

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

$$\psi^\star(g) \equiv \sup_x g \cdot x - \psi(x).$$

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 $\|\cdot\|_{(t),\star}$. We denote the indicator function on a convex set $\mathcal{X}$ by

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

We can summarize our basic assumptions as follows:

Setting 1. *We consider the algorithm that plays according to Eq.* (1) *based on r_t that satisfy $r_t(x) \geq 0$ for $t \in \{0, 1, \ldots, T\}$, against a sequence of convex cost functions $f_t : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$.*

We can now introduce the theorems which will be our main focus. First, we consider a bound for FTRL-Proximal:

Theorem 1w. 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 $\|\cdot\|_{(t)}$ for $x \in \operatorname{dom} h_{0:t}$, 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 $x^* \in \mathbb{R}^n$,*

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

The proof of this theorem relies on the standard FTRL Lemma (Lemma 5 in Section 3.1). Theorem 1w is the best possible using this lemma, but using the Strong FTRL Lemma (Lemma 6), we can improve this result by a constant factor:

Theorem 1. Strong FTRL-Proximal Bound *Under the same conditions as Theorem 1*w*, we can improve the bound to*

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

Finally, we have a bound for any FTRL algorithm (including Dual Averaging):

Theorem 2. General FTRL Bound *Consider Setting 1, and suppose the r_t are chosen such that $h_{0:t} + f_{t+1}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$ for $x \in \operatorname{dom} r_{0:t}$. Then,*

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

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 x, and Theorems 1 and 2 produce *identical* bounds (and algorithms).[2] On the other hand, when the f_t are not strongly convex (e.g., linear), a sufficient condition for all these theorems is choosing the r_t such that $r_{0:t}$ is 1-strongly-convex w.r.t. $\|\cdot\|_{(t)}$. It is worth emphasizing here 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 $\|g_t\|^2_{(t),\star}$ term); this is not possible using Theorem 2, since we have $\|g_t\|^2_{(t-1),\star}$. This makes constructing AdaGrad-style adaptive learning rate algorithms for FTRL-Proximal easier (McMahan and Streeter, 2010), whereas with Dual Averaging algorithms one must start with slightly more regularization. We will see this in more detail in the next section.

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).

Theorem 2 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 2 is usually applied to regularizers where 0 is a global minimizer of $r_{0:t}$ for each t. The theorem does not

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

require this; however, such a condition is usually necessary to bound $r_{0:T-1}(x^*)$ and hence $\mathrm{Regret}(x^*)$ in terms of $\|x^*\|$.

Less general versions of these theorems often assume that each $r_{0:t}$ is say α_t-strongly-convex with respect to a fixed norm $\|\cdot\|$. Our results include this as a special case, see Lemma 3 in Section 3.2 as well as discussion in the next section.

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

$$\mathrm{Regret}(x^*) \leq r_0(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_\star^2. \tag{4}$$

Theorem 1w can also be applied in this way, but it again loses a factor of $\frac{1}{2}$; this gives, e.g., Shalev-Shwartz (2012, Theorem 2.11).

2.2 Application to Specific Algorithms

Before proving these theorems, we apply them to a variety of specific algorithms. We will use the following lemma, which collects some straightforward facts for the sequence of incremental regularizers r_t. This lemma lets us encode a non-increasing learning rate schedule $\eta_t = \frac{1}{\sigma_{0:t}}$ implicitly in the definition of the norms $\|\cdot\|_{(t)}$, which simplifies notation and generalizes the results. 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 $r^\star_{0:t}(x) \leq r^\star_{0:t-1}(x)$. 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. $\|x\|_{(t)} = \sqrt{\sigma_{0:t}}\|x\|$.*

Constant learning-rate gradient descent As a warm-up, we first consider a non-adaptive algorithm, constant learning-rate unprojected online gradient descent, which plays

$$x_{t+1} = x_t - \eta g_t, \tag{5}$$

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 gradient descent and FTRL, which we will use to analyze this algorithm. If we take FTRL with $r_0(x) = \frac{1}{2\eta}\|x\|^2$ and $r_t(x) = 0$ for $t \geq 1$, we have the update

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

which we can again solve in closed form to see $x_{t+1} = -\eta g_{1:t}$ as well. Applying either Theorem 2 or 1 gives the bound of Eq. (4), in this case

$$\mathrm{Regret}(x^*) \leq \frac{1}{2\eta}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^{T} \eta\|g_t\|_2^2.$$

Suppose we are concerned with x^* where $\|x^*\|_2 \leq R$ and the g_t satisfy $\|g_t\|_2 \leq G$. Then plugging in these values, we find that choosing $\eta = \frac{R}{G\sqrt{T}}$ minimizes the above expression, which leads to the standard bound $\mathrm{Regret}(x^*) \leq RG\sqrt{T}$.

Dual Averaging If we choose $r_t(x) = \frac{\sigma_t}{2}\|x\|_2^2$ for constants $\sigma_t \geq 0$, then $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \sqrt{\sigma_{0:t}}\|x\|_2$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{0:t}}}\|x\|_2$. It will be convenient to let $\eta_t = \frac{1}{\sigma_{0:t}}$, and as the notation suggests, η_t is exactly analogous to a learning rate as in gradient descent. Note that any non-increasing learning rate schedule can be expressed in this manner by choosing: $\sigma_t = 1/\eta_t - 1/\eta_{t-1}$. With this definition, plugging into Theorem 2 then gives

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

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

$$\text{Regret}(x^*) \leq \frac{\sqrt{2}}{2}\left(R + \frac{\|x^*\|_2^2}{R}\right)G\sqrt{T}. \tag{6}$$

When in fact $\|x^*\| \leq R$, we have Regret $\leq \sqrt{2}RG\sqrt{T}$, but the bound of Eq. (6) is valid (and meaningful) for arbitrary $x^* \in \mathbb{R}^n$.

Dual Averaging can also be restricted to play from a closed bounded feasible set $\mathcal{X}$ by including $I_{\mathcal{X}}$ in r_0; this does not change the bound of Eq. (6) except for adding a $I_{\mathcal{X}}(x^*)$ term, which means the bound is only non-vacuous for $x^* \in \mathcal{X}$. If we believe some $x^* \notin \mathcal{X}$ could perform very well, then this is a significant sacrifice that should only be made if external constraints require we play $x_t \in \mathcal{X}$.

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 where the $\lambda_1\|x\|_1$ terms are considered part of the f_t in Section 4.

FTRL-Proximal Suppose $\mathcal{X} \subseteq \{x \mid \|x\|_2 \leq R\}$, and we choose $r_0(x) = I_{\mathcal{X}}(x)$ and for $t > 1$, $r_t(x) = \frac{\sigma_t}{2}\|x - x_t\|_2^2$. Then $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \sqrt{\sigma_{1:t}}\|x\|_2$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{1:t}}}\|x\|_2$. Note $r_{0:t}(x^*) \leq \frac{\sigma_{1:t}}{2}(2R)^2$ for any $x^* \in \mathcal{X}$, since each $x_t \in \mathcal{X}$. Thus, applying Theorem 1, we have

$$\forall x^* \in \mathcal{X}, \quad \text{Regret}(x^*) \leq \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|^2 + \frac{1}{2\eta_T}(2R)^2, \tag{7}$$

where we let $\eta_t = \frac{1}{\sigma_{1:t}}$. Choosing $\eta_t = \frac{\sqrt{2}R}{G\sqrt{t}}$ and assuming $\|x^*\| \leq \mathbb{R}$, we have

$$\text{Regret}(x^*) \leq 2\sqrt{2}RG\sqrt{T}. \tag{8}$$

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).

FTRL-Proximal with Diagonal Matrix Learning Rates For simplicity, first consider a 1-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. (7). This gives

$$\text{Regret}(x^*) \le 2\sqrt{2}R\sqrt{\sum_{s=1}^t g_s^2}, \tag{9}$$

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

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

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

For a proof see Auer et al. (2002) or Streeter and McMahan (2010, Lemma 1). The bound of Eq. (9) gives us a fully adaptive version of Eq. (8): 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 a diagonal AdaGrad-style algorithm for n-dimensions we need only apply the above technique on a per-coordinate basis. Note Streeter and McMahan (2010) takes this approach directly; the more general analysis here allows us to handle arbitrary feasible sets and L_1 or other non-smooth regularization. Define $r_t(x) = \frac{1}{2}\|Q_t^{\frac{1}{2}}(x - x_t)\|_2^2$ for positive semi-definite Q_t, so $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \|(Q_{1:t})^{\frac{1}{2}}x\|_2$, which has dual norm $\|x\|_{(t),\star} = \|(Q_{1:t})^{-\frac{1}{2}}x\|_2$. In particular, we define diagonal Q_t so that ith diagonal entry of $Q_{1:t}$ is $\sqrt{\sum_{s=1}^t g_{s,i}^2}$, and let $r_0(x) = I_{\mathcal{X}}(x)$ for a closed and bounded convex $\mathcal{X}$. Then, plugging into Theorem 1w recovers McMahan and Streeter (2010, Theorem 2), and we can improve by a constant factor using Theorem 1. Essentially, this bound amounts to summing Eq. (9) across all n dimensions; a careful analysis shows this bound is at least as good (and often better) than that of Eq. (8).

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 this algorithm is easy and efficient to implement. In fact, 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 represented using two length-n vectors, one to maintain the linear terms ($g_{1:t}$ plus some 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. For full pseudo-code which also incorporates L_1 and L_2 regularization, see McMahan et al. (2013).

AdaGrad Dual Averaging Similar ideas can be applied Dual Averaging (where we center each r_t at the origin), but again 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

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

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).

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

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

where we have used the standard 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

$$f_t(x) \geq \underbrace{f_t(x_t) + g_t(x - x_t) + \frac{1}{2}\|x - x_t\|_2^2}_{=\bar{f}_t}.$$

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; this gives rise to the online gradient descent algorithm for strongly convex functions given by Hazan et al. (2007).

3 A General Analysis Technique

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

3.1 Inductive Lemmas

In this section we consider two lemmas that let us analyze arbitrary FTRL-style algorithms. The first is quite well known:

Lemma 5 (Standard FTRL Lemma)**.** *Let f_t be a sequence of arbitrary (e.g., 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 plays these x_t achieves*

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

For example, see Kalai and Vempala (2005), Hazan (2008), Hazan (2010, Lemma 1), and Shalev-Shwartz (2012, Lemma 2.3). 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 playing $x_t = \arg\min_x f_{1:t}(x)$ (which requires peaking ahead at f_t to choose x_t), then the player's regret is bounded above by zero. However, as we see by comparing Theorems 1W and 1, 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 playing the post-hoc optimal point. The following result captures this fact, and hence allows for tighter bounds:

Lemma 6 (Strong FTRL Lemma)**.** *Under the same conditions as Lemma 5, we can tighten the bound to*

$$\text{Regret}(x^*) \le r_{0:T}(x^*) + \sum_{t=1}^{T} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t). \tag{10}$$

The Standard FTRL Lemma measures regret in terms of how much better x_{t+1} is for the loss function f_t than the point actually played, x_t: $f_t(x_t) - f_t(x_{t+1})$. Neglecting the $-r_t(x_t)$ term (which is mostly book keeping), the Strong FTRL Lemma does the same thing, but measures the improvement achieved by x_{t+1} not on f_t, but on the full objective function $h_{0:t}$. These per-round terms can be seen as measuring the stability of the algorithm, an online analogue to the role of of stability in the stochastic setting, see for example Rakhlin et al. (2005) and Shalev-Shwartz et al. (2010). On the other hand, given (say) linear functions, stability can only be enforced by the use of a strong regularizer, leading to an increase in the $r_{0:t}(x^*)$ term. At the heart of the adaptive algorithms we study is the ability to dynamically balance these two competing goals.

We immediately have the following corollary, which relates the above statement to the primal-dual style of analysis:

Corollary 7. *Consider the same conditions as Lemma 6, but further suppose the loss functions are linear, $f_t(x) = g_t \cdot x_t$. Then,*

$$h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) = r^{\star}_{0:t}(-g_{1:t}) - r^{\star}_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t, \tag{11}$$

which implies

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

We make a few remarks before proving these results at the end of this section. Corollary 7 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 linear case simple 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 the an interpretation of $r^\star$ as a benchmark target for reward.

Note, however, that Lemma 6 is strictly stronger than Corollary 7: it applies to non-convex f_t and r_t. Further, even for convex f_t, it can be more directly useful: for example, we can directly analyze strongly-convex f_t with all $r_t(x) = 0$ using the first statement. Lemma 6 is also arguably simpler, in that it does not require the introduction of convexity or the Fenchel conjugate.

For readers familiar with the proof of the Standard FTRL Lemma, we point out that rather than first analyzing the Be-The-Leader algorithm and showing it has no regret, the key to proving the strong version is to immediately 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.

Proof of Lemma 6. First, we bound a quantity that is essentially our regret if we had played the FTL algorithm against the functions $h_1, \ldots h_T$ (for convenience, we include a $-h_0(x^*)$ term as well):

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

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

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

Re-arranging the inequality proves the lemma. □

We remark it is possible to make Lemma 6 an *equality* if we include the non-positive term $h_{1:T}(x_{T+1}) - h_{1:T}(x^*)$ on the RHS, since we can assume $r_0(x_1) = 0$ without loss of generality. Further, if one is actually interested in the performance of the FTL algorithm against the h_t (e.g., if all the r_t are uniformly zero), then choosing $x^* = x_{T+1}$ is natural.

Proof of Corollary 7. Using the definition of the Fenchel conjugate and of x_{t+1}, we have

$$r_{0:t}^\star(-g_{1:t}) = \max_x \; -g_{1:t} \cdot x - r_{0:t}(x) = -\big(\min_x \; g_{1:t} \cdot x + r_{0:t}(x)\big) = -h_{0:t}(x_{t+1}). \tag{12}$$

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. (12) with $t \to t-1$. Combining this with Eq. (12) again $(-h_{0:t}(x_{t+1}) = r^\star_{0:t}(-g_{1:t}))$ proves Eq. (11). □

3.2 Tools from Convex Analysis

Here we highlight a few key tools from convex analysis that will be applied to bounding the per-round terms that appear in the preceding lemmas. For more background on convex analysis, see Shalev-Shwartz (2012), Rockafellar (1997), Shalev-Shwartz (2007). The following lemma is a powerful tool for bounding the per-round terms of both Lemma 5 and 6. We defer the proofs of the results in this section to Appendix A.

Lemma 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 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*

$$\|x_1 - x_2\| \le \|b\|_\star, \tag{13}$$

and for any x',

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

When ϕ_1 and ψ are quadratics (one 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 4), it will be useful to split the change ψ in the objective function ϕ into two components:

Corollary 9. *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',*

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

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 7. A function ψ is *σ-strongly-smooth* with respect to a norm $\|\cdot\|$ if it is differentiable and for all x, y we have

$$\psi(y) \le \psi(x) + \nabla\psi(x) \cdot (y - x) + \tfrac{\sigma}{2}\|y - x\|^2. \tag{14}$$

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

Lemma 10. *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 strongly convexity implies strongly smooth direction see Shalev-Shwartz (2007, Lemma 15), and for the other direction see Kakade et al. (2012, Theorem 3).

3.3 Regret Bound Proofs

3.3.1 Analysis of FTRL-Proximal using the Standard FTRL Lemma

In this section, we prove Theorem 1w using strong smoothness via Lemma 8. An alternative proof that uses strong convexity directly can also be done, closely following Shalev-Shwartz (2012, Sec. 2.5.2).

Proof of Theorem 1w Applying Lemma 5, it is sufficient 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

$$\begin{aligned} f_t(x_t) - f_t(x_{t+1}) &\leq g_t \cdot (x_t - x_{t+1}) && \text{Convexity of } f_t \text{ and } g_t \in \partial f_t(x_t) \\ &\leq \|g_t\|_{(t),\star} \|x_t - x_{t+1}\|_{(t)} && \text{Property of dual norms} \\ &\leq \|g_t\|_{(t),\star} \|g_t\|_{(t),\star} = \|g_t\|^2_{(t),\star}. && \text{Using Eq. (13) from Lemma 8} \end{aligned}$$

□

Interestingly, it appears difficult to achieve a tight (up to constant factors) analysis of non-proximal FTRL algorithms (e.g., Dual Averaging) using Theorem 5. The Strong FTRL Lemma, however, will allow us to accomplish this.

3.3.2 Analysis using the Strong FTRL Lemma

In this section, we prove Theorem 1 and Theorem 2 using Lemma 6. Stating these two analyses in a common framework makes clear exactly where the "off-by-one" issue arises for Dual Averaging, and how assuming proximal r_t resolves this issue. The key tool is Lemma 8, though for comparison we also provide an analysis of Theorem 2 from Corollary 7 directly using strong smoothness.

Proximal Regularizers (Proof of Theorem 1) We work to bound the terms in the sum in Eq. (10). 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,

$$\begin{aligned} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) &\leq h_{0:t}(x_t) - h_{0:t}(x_{t+1}) && \text{Since } r_t(x) \geq 0 \\ &= \phi_1(x_t) + f_t(x_t) - \phi_1(x_{t+1}) - f_t(x_{t+1}) \\ &\leq \frac{1}{2}\|g_t\|^2_{(t),\star}, && (15) \end{aligned}$$

where the last line follows by applying Lemma 8 to ϕ_1 and $\phi_2(x) = \phi_1(x) + f_t(x) = h_{0:t}(x)$. Plugging into Lemma 6 completes the proof. □

Non-proximal Regularizers (Proof Theorem 2) Again fix a particular round t. For Lemma 8 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 8 to ϕ_2 (with $x' = x_{t+1}$), we have

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

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 6. However, we need to make one additional observation: the choice of r_T does not impact $\|\cdot\|_{(T),\star}$, and only increases $r_{0:T}(x^*)$. Further, r_T does not influence any of the points $x_1, \dots, x_T$ played by the algorithm. Thus, for analysis purposes, we can take $r_T(x) = 0$ without loss of generality, and hence replace $r_{0:T}$ with $r_{0:T-1}$ in the final bound. □

The final argument in this proof is another manifestation of the "off-by-one" difference between FTRL-Proximal and Dual Averaging. The FTRL-Proximal bound essentially depends on $r_1, \dots, r_T$ (we can essentially take $r_0(x) = 0$), whereas Dual Averaging depends on $r_0, \dots, r_{T-1}$.

Non-proximal Regularizers via Potential functions We give an alternative proof of Thm 2 for linear functions, $f_t(x) = g_t \cdot x$, using Eq. (11). Recall in this case $x_t = \nabla r^\star_{1:t-1}(-g_{1:t-1})$, and by Lemma 10, $r^\star_{1:t-1}$ is 1-strongly-smooth with respect to $\|\cdot\|_{(t-1),\star}$, and so

$$r^\star_{1:t-1}(-g_{1:t}) \leq r^\star_{1:t-1}(-g_{1:t-1}) - x_t \cdot g_t + \frac{1}{2}\|g_t\|^2_{(t-1),\star}, \tag{16}$$

and we can bound the per-round terms in Eq. (11) 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. (16) 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.

4 Composite Losses

In this section, we consider the special case where $f_t(x) = g_t \cdot x + \alpha_t \Psi(x)$, where Ψ is convex function taking on only non-negative values on its domain, and the $\alpha_t \in \mathbb{R}$ are non-increasing. We further assume Ψ and r_0 are both minimized at 0, and WLOG $\Psi(0) = 0$ (as usual, additive constant terms do not impact regret). In applications, generally we will take $g_t = \partial \ell_t(x_t)$, where ℓ_t is for example a loss function measuring the prediction error on the tth training example for a model parameterized by x_t. The $\alpha_t > 0$ are constants, and Ψ can generally be viewed as a non-smooth regularization term, e.g., $\Psi(x) = \|x\|_1$. Such a formulation is studied by Xiao (2009), Duchi et al. (2010b, 2011).

We can immediately apply Theorem 1 or Theorem 2, but this would give us terms in the bound that depend on terms like $\|g_t + g_t^{(\Psi)}\|^2_{(t),\star}$ 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 make the problem no harder, and we would like to demonstrate this in our analysis. This is accomplished in a straightforward manor by using Corollary 9 in place of Lemma 8. For simplicity, we only consider the extension of Theorem 2 for FTRL-Proximal:

Theorem 11. *Let Ψ be a convex function with $\Psi(x) \geq 0$, and $\Psi(x_1) = 0$. Consider the same conditions as Theorem 1*W*, but define $f_t(x) = g_t \cdot x + \alpha_t \Psi(x)$ for non-increasing constants $\alpha_t \geq 0$. Note in general, $g_t \notin \partial f_t(x_t)$. Then FTRL-Proximal still attains an identical regret bound in terms of the g_t,* $\text{Regret}(x^*) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t),\star}$.

Proof sketch. The proof closely follows the proof of Theorem 1 in Section 3.3.2. 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 $x_{t+1} = \arg\min_x \phi_1(x) + g_t \cdot x + \alpha_t \Psi(x)$. Then, using Corollary 9 lets us replace Eq. (15) 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 6 we sum over t, and we can bound the Ψ terms separately as

$$\sum_{t=1}^{T} \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) = \alpha_1 \Psi(x_1) - \alpha_T \Psi(x_{T+1}) + \sum_{t=2}^{T} \alpha_t \Psi(x_t) - \alpha_{t-1} \Psi(x_t) \leq 0, \quad (17)$$

since $\Psi(x) \geq 0$, $\alpha_t \leq \alpha_{t-1}$, and $\Psi(x_1) = 0$. Thus, we have the same per-round bound as in Eq. (15), and so by Lemma 6 we arrive at the same bound as in Theorem 1. □

As an alternative to considering the $\alpha_t \Psi(x)$ terms as part of the f_t (and hence part of the objective on which we wish to measure regret), in some cases it may be more natural to consider these terms part of the regularization functions r_t (that is, part of the algorithm), e.g. $r_t(x) = r'_t(x) + \alpha_t \Psi(x)$ where r' are the regularizers prescribed by the algorithm to guarantee low regret. 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 smaller models (as in McMahan et al. (2013)). If the original $r'_{0:t}$ were strongly convex w.r.t. $\|\cdot\|_{(t)}$, then $r_{0:t}$ has this property as well (since Ψ is convex), and so in this case Theorems 1 and 2 apply immediately.

5 Mirror Descent, FTRL, and Implicit Updates

Recall in Section 2.2, our first example showed the equivalence between constant-step-size gradient descent and a fixed-regularizer FTRL algorithm. This equivalence is well-known in the case where $r_t(x) = 0$ for $t > 0$, 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 with origin-centered regularizers and proximal regularizers coincide. In this section, we show how this equivalence extends to adaptive regularizers (equivalently, adaptive learning rates) and composite objectives (including projection onto a feasible set). This builds on the work of McMahan (2011), but we make some crucial improvements in order to obtain an exact equivalence result for all Mirror Descent algorithms.

Mirror Descent Even in the non-adaptive case, Mirror Descent can be expressed as a variety of different updates, some equivalent but some not;[3] we provide a summary in Appendix B. Building on the previous section, we consider composite loss functions $f_t(x) = g_t \cdot x + \alpha_t \Psi(x)$. Generally we view g_t 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.

To define the Mirror Descent algorithm we first define the Bregman divergence with respect to a strictly-convex differentiable function ϕ:

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

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

We define an adaptive Mirror Descent algorithm by a sequence of continuously differentiable incremental regularizers $r_0, r_1, \dots$, chosen so $r_{0:t}$ is strongly convex. From this, we define the time-indexed Bregman divergence $\mathcal{B}_{r_{0:t}}$; to simplify notation we define $\mathcal{B}_t \equiv \mathcal{B}_{r_{0:t}}$. The Mirror Descent update is then given by

$$\hat{x}_{t+1} = \arg\min_x \; g_t \cdot x + \alpha_t \Psi(x) + \mathcal{B}_t(x, \hat{x}_t). \tag{18}$$

We use $\hat{x}$ to distinguish this update from an FTRL update we will introduce shortly.

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 in by Duchi et al. (2010b).

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

$$\hat{x}_{t+1} = \arg\min_x \; \ell_t(x) + \mathcal{B}_t(x, \hat{x}_t), \tag{19}$$

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 composite Mirror Descent update Eq. (18) 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 $\Psi(x)$ 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).

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 $x_{t+1} = \arg\min_x g_{1:t} + r(x)$ (see Appendix B). In fact, Shalev-Shwartz (2012, Sec. 2.6) refers to this update (with

[3] In particular, it is common to see updates written in terms of $\triangledown r^\star(\theta)$ for a strongly convex regularizer r, based on the fact that $\triangledown r^\star(-\theta) = \arg\min_x \theta \cdot x + r(x)$ (see Lemma 14 in Appendix A).

linearized losses) explicitly as Mirror Descent. However, we prefer to use the name FTRL or Dual Averaging for this case.

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. This difference becomes critical when one considers the introduction of L_1 regularization to introduce sparsity. 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 representations, 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 result is typically much better sparsity-accuracy tradeoffs (McMahan, 2011, McMahan et al., 2013).

5.1 Adaptive Mirror Descent is an FTRL-Proximal Algorithm

We will show that the Mirror Descent update Eq. (18) can be expressed as a particular FTRL update (that is, as the argmin of a sum of lower bounds of the loss functions f_t, plus suitable adaptive regularization terms). In particular, consider a Mirror Descent algorithm defined by the choice of $r_0, r_1, \ldots, r_T$. Then, we define the FTRL-style update

$$x_{t+1} = \arg\min_x \bar{f}_{1:t-1}(x) + f_t(x) + \tilde{r}_{0:t}(x), \tag{20}$$

where $\bar{f}_t$ is an appropriate lower bound on f_t, and $\tilde{r}_t$ is an incremental proximal regularizer defined in terms of r_t, namely

$$\tilde{r}_t(x) = r_t(x) - \big(r_t(x_t) + \triangledown r_t(x_t) \cdot (x - x_t)\big). \tag{21}$$

Note that $\tilde{r}_t$ is indeed minimized by x_t, $\tilde{r}_t(x_t) = 0$, and in fact $\tilde{r}_t(x)$ is essentially[4] the Bregman divergence $\mathcal{B}_{r_t}(x, x_t)$. This careful choice for $\tilde{r}_t$ allows us to significantly improve on the results of McMahan (2011), which given an r_t minimized at zero defined $\tilde{r}'_t(x) = r_t(x - x_t)$. The choice $\tilde{r}'_t$ is equivalent to our $\tilde{r}_t$ for quadratic r_t, but does not lead to a clean equivalence statement for arbitrary Mirror Descent algorithms (and may not even be applicable if the domain of r_t is bounded, since $x - x_t$ could fall outside this domain).

We define $\bar{f}_t$ in terms of $\bar{\Psi}_t(x)$, a linear lower bound on $\alpha_t \Psi(x)$ that is tight at x_{t+1} (*not* x_t); formally, we take

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

with $g_t^{(\Psi)} \in \partial(\alpha_t \Psi)(x_{t+1})$ such that

$$g_{1:t} + g_{1:t}^{(\Psi)} + \triangledown \tilde{r}_{0:t}(x_{t+1}) = 0. \tag{22}$$

To see (inductively) that this is always possible, note we can re-write Eq. (20) as

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + g_{1:t-1}^{(\Psi)} \cdot x + \alpha_t \Psi(x) + \tilde{r}_{0:t}(x) + (\text{constant}). \tag{23}$$

[4] This is a slight abuse of notation, as we do not require that the r_t are strictly convex, for example $r_t(x) = \tilde{r}_t(x) = 0$ for $t > 0$ is common in modeling the non-adaptive case.

The subdifferential of the objective of Eq. (23) at a point x is

$$g_{1:t} + g^{(\Psi)}_{1:t-1} + \partial(\alpha_t\Psi)(x) + \nabla\tilde{r}_{0:t}(x). \tag{24}$$

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. (22). 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. We would usually drop the constant terms in $\bar{\Psi}_t(x)$, as they do not impact argmins or regret, but it will simplify bookkeeping to have $\bar{\Psi}_t(x_{t+1}) = \alpha_t\Psi(x_{t+1})$ in the arguments below. With these definitions in place, we can now state and prove the main result of this section:

Theorem 12. *The Mirror Descent update of Eq.* (18) *and the FTRL-Proximal update of Eq.* (20) *play 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 played by the FTRL-Proximal algorithm of Eq. (20). Since x_t minimizes this objective, zero must be a subgradient at x_t. Letting $g^{(r)}_s = \nabla r_s(x_s)$ and noting $\nabla\tilde{r}_t(x) = \nabla r_t(x) - \nabla r_t(x_t)$, we have $g_{1:t-1} + g^{(\Psi)}_{1:t-1} + \nabla r_{0:t-1}(x_t) - g^{(r)}_{0:t-1} = 0$ following Eq. (24). Since $x_t = \hat{x}_t$ by induction hypothesis, we can rearrange and conclude

$$-\nabla r_{0:t-1}(\hat{x}_t) = g_{1:t-1} + g^{(\Psi)}_{1:t-1} - g^{(r)}_{0:t-1}. \tag{25}$$

For Mirror Descent, the gradient of the objective in Eq. (18) must be zero for $\hat{x}_{t+1}$, and so there exists a $\hat{g}^{(\Psi)}_t \in \partial(\alpha_t\Psi)(\hat{x}_{t+1})$ such that

$$\begin{aligned}
0 &= g_t + \hat{g}^{(\Psi)}_t + \nabla r_{0:t}(\hat{x}_{t+1}) - \nabla r_{0:t}(\hat{x}_t) \\
&= g_t + \hat{g}^{(\Psi)}_t + \nabla r_{0:t}(\hat{x}_{t+1}) - \nabla r_{0:t-1}(\hat{x}_t) - g^{(r)}_t && \text{IH and } \nabla r_t(x_t) = g^{(r)}_t \\
&= g_t + \hat{g}^{(\Psi)}_t + \nabla r_{0:t}(\hat{x}_{t+1}) + g_{1:t-1} + g^{(\Psi)}_{1:t-1} - g^{(r)}_{0:t-1} - g^{(r)}_t && \text{Using Eq. (25)} \\
&= g_{1:t} + g^{(\Psi)}_{1:t-1} + \hat{g}^{(\Psi)}_t + \nabla r_{0:t}(\hat{x}_{t+1}) - g^{(r)}_{0:t} \\
&= g_{1:t} + g^{(\Psi)}_{1:t-1} + \hat{g}^{(\Psi)}_t + \nabla\tilde{r}_{0:t}(\hat{x}_{t+1}).
\end{aligned}$$

The last line implies zero is a subgradient of the objective of Eq. (20) 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 $\hat{x}_{t+1} = x_{t+1}$. □

In light of this theorem, it is worth considering what algorithm one should run in practice. Since we can now write composite-objective Mirror Descent as a particular FTRL update, it is worth comparing this form to the direct application of FTRL-Proximal (as per Section 4):

$$\begin{array}{lllllll}
 & & & (A) & & (B) & (C) \\
\text{MD} & x_{t+1} & = \arg\min_x & g_{1:t}\cdot x & + & g^{(\Psi)}_{1:t-1}\cdot x + \alpha_t\Psi(x) & +\tilde{r}_{0:t}(x) \\
\text{FTRL-Proximal} & x_{t+1} & = \arg\min_x & g_{1:t}\cdot x & + & \alpha_{1:t}\Psi(x) & +\tilde{r}_{0:t}(x)
\end{array}$$

Both algorithms use a linear approximation to the loss functions ℓ_t (A), 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 $\bar{\Psi}_t$, keeping only the current $\alpha_t\Psi(x)$ term explicitly. With a direct application of FTRL-Proximal, on the other hand, we represent the full weight of the Ψ terms exactly as $\alpha_{1:t}\Psi(x)$.

As pointed out in McMahan (2011), this is precisely the reason FTRL algorithms (including Dual Averaging) are able to produce much sparser solutions when we take $\Psi(x) = \|x\|_1$, compared with the composite-objective Mirror Descent update with L_1 regularization (equivalent to the FOBOS algorithm of Duchi and Singer (2009)). We explore this further in the following section:

5.2 Example Application: L_1 Regularization

It is worthwhile to consider the simplest application of L_1 regularization: a static regularizer $r_0(x) = \frac{1}{2\eta}\|x\|_2^2$, and $\alpha_t\Psi(x) = \lambda\|x\|_1$ for all t. The updates become

$$\begin{array}{llllll} \text{MD} & x_{t+1} = \arg\min\limits_x & g_{1:t}\cdot x & + g^{(\Psi)}_{1:t-1}\cdot x + \lambda\|x\|_1 & + \dfrac{1}{2\eta}\|x\|_2^2 & (26) \\ \text{FTRL} & x_{t+1} = \arg\min\limits_x & g_{1:t}\cdot x & + t\lambda\|x\|_1 & + \dfrac{1}{2\eta}\|x\|_2^2 & (27) \end{array}$$

and it is clear FTRL uses a much stronger explicit L_1 penalty ($\alpha_{1:t} = t\lambda$ instead of just $\alpha_t = \lambda$). We can write the Mirror Descent update Eq. (26) in its native form as

$$\begin{aligned} x_{t+1} &= \arg\min_x g_t\cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x - x_t\|_2^2 \\ &= \arg\min_x \big(g_t - \frac{x_t}{\eta}\big)\cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x\|_2^2. \end{aligned} \tag{28}$$

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

$$\arg\min_{x\in\mathbb{R}} b\cdot x + \lambda\|x\|_1 + \frac{a}{2}\|x\|^2 = \begin{cases} 0 & \text{when } |b| \leq \lambda \\ -\frac{1}{a}(b - \operatorname{sign}(b)\lambda) & \text{otherwise} \end{cases}. \tag{29}$$

Thus, we can simplify Eq. (28) to

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

In fact, if we define $g_t^{(\Psi)} \in \partial\lambda\|x_{t+1}\|_1$ and take $g_t^{(\Psi)} = x_t/\eta - g_t$ when $x_{t+1} = 0$ in particular, 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 the subgradient approximation at the *next* point).

A Simple Example Now, we consider a specific example (WLOG, in one dimension). 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 2.2).

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 $\hat{x} = -\hat{x} = 0$). In fact, 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 $\lambda < G$, the algorithm 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 played by Mirror Descent, the gradients, and the sub-gradients 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. (28), it is straightforward and instructive to verify the FTRL-Proximal representation is indeed equivalent. For example, using the values from the table, for x_5 we have

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

where we solve the argmin by applying Eq. (29) with $b = g_1 + G + \lambda$.

Now, contrast this with the FTRL update of Eq. (27); we can solve this update in closed form using Eq. (29). First, note that FTRL will not oscillate in the same way, unless $\lambda = 0$. We have immediately 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.

5.3 Analysis of Mirror Descent as Implicit Update FTRL

Having established the equivalence between Mirror Descent and a particular FTRL-Proximal update (namely Eq. (20)), we now show how we can use the general analysis techniques for FTRL developed in this work to prove regret bounds for any adaptive 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 so $\bar{f}_t = f_t$ for all t, and we can apply Theorem 1 directly to Eq. (20), which gives us

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

In the case of a composite-objective (nontrivial Ψ terms), we will arrive at the same bound, but must refine our analysis somewhat to encompass the partial implicit update of Eq. (20).

Theorem 13. *Under the same conditions as Theorem 11, the Mirror Descent update expressed in FTRL form in Eq.* (20) *has*

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

We note this bound matches Duchi et al. (2011, Prop. 3),[5] and also encompasses Theorem 2 of Duchi et al. (2010b).[6]

Proof sketch. First, the update

$$x_{t+1} = \arg\min_x \bar{f}_{1:t}(x) + \tilde{r}_{0:t}(x) \tag{30}$$

is equivalent to Eq. (20), since Equations (22) and (24) imply 0 is in the subgradient of the objective Eq. (20) at the x_{t+1} given by Eq. (30).

Observe that Eq. (30) defines a standard (non-implicit) FTRL-Proximal algorithm run on the linear functions $\bar{f}_t$ (we can imagine the $\bar{f}_t$ are computed by a black-box given f_t which solves the optimization problem of Eq. (20) in order to compute $g_t^{(\Psi)}$). This means we can use the strong FTRL lemma (Lemma 6) to bound the regret against the functions $\bar{f}_t$:

$$\begin{aligned}
\text{Regret}(x^*, \bar{f}_t) &\le \tilde{r}_{0:T}(x^*) + \sum_{t=1}^{T} \bar{h}_{1:t}(x_t) - \bar{h}_{1:t}(x_{t+1}) - r_t(x_t) \\
&\le \tilde{r}_{0:T}(x^*) + \sum_{t=1}^{T} \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}(x_t) - \bar{\Psi}_t(x_{t+1}),
\end{aligned}$$

where the second line follows the proof of Theorem 11, using the helpers

$$\phi_1(x) = g_{1:t-1} \cdot x + g^{(\Psi)}_{1:t-1} \cdot x + \tilde{r}_{0:t}(x) \qquad \text{and} \qquad \phi_2(x) = \phi_1(x) + g_t \cdot x + \bar{\Psi}_t(x)$$

for Corollary 9. However, we actually care about our regret against the functions f_t, not $\bar{f}_t$. We have $\bar{f}_t(x^*) \le f_t(x^*)$, but we must add terms $f_t(x_t) - \bar{f}_t(x_t) = \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t)$ to

[5] 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. (18). We can take $\eta = 1$ in their bound WLOG. Then, using the fact that ψ_t in their notation is $r_{1:t}$ in our notation, we have

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

[6] We can take their $\alpha = 1$ and $\eta = 1$ WLOG, 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 $\le \mathcal{B}_{r_0}(x^*, x_1) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_\star$, matching exactly the bound of our Theorem 13 (noting $\tilde{r}_{0:t}(x^*) = \mathcal{B}_{r_0}(x^*, x_1)$ in this case).

account for the fact our actual loss $f_t(x_t)$ could be larger than $\bar{f}_t(x_t)$. This gives

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

where the equality uses $\bar{\Psi}_t(x_{t+1}) = \alpha_t\Psi(x_{t+1})$. The result then follows from the argument of Eq. (17) in the proof of Theorem 11. □

6 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. Our emphasis has been on the case of adaptive regularizers or learning rates, but the results immediately recover those for a fixed learning rate or regularizer as well.

Section 2.2 introduced some specific algorithms in this family, but we should emphasize that our results hold in a similar straightforward fashion for a wide array of other problems. A notable example is the family of expert and bandit algorithms; our results lead immediately to adaptive versions by considering Dual Averaging with an entropic regularizer. The necessary properties of this regularizer, as well as closed-forms for the update, are well know (e.g., Shalev-Shwartz (2012), Ex. 2.5 and Eq. 2.10), with our results providing the necessary adaptive machinery

We have focused entirely on online algorithms and regret bounds, but note that 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). On the flip side, we note that 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 batch stochastic optimization problems.

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 *NIPS*, 2007.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3), 2003.

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*, 50:2050–2057, 2004.

John Duchi and Yoram Singer. Efficient learning using forward-backward splitting. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 495–503. 2009.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, 2010a.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, 2010b.

John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Elad Hazan. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*, 2008.

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

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Mach. Learn.*, 69:169–192, December 2007. ISSN 0885-6125. doi: 10.1007/s10994-007-5016-8.

S. Kakade and Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *NIPS*, 2008.

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

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and Systems Sciences*, 71(3), 2005. ISSN 0022-0000.

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 *ICML*, 2010.

H. Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

H. Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In *NIPS*, 2013.

H. Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Proceedings of the 27th Annual Conference on Learning Theory (COLT)*, 2014.

H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *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 *KDD*, 2013.

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

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

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

Francesco Orabona. Dimension-free exponentiated gradient. In *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, 1997.

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. *JMLR*, 2010.

Matthew Streeter and H. Brendan McMahan. No-regret algorithms for unconstrained online convex optimization. In *NIPS*, 2012.

Matthew J. Streeter and H. Brendan McMahan. Less regret via online conditioning. 2010.

M. K. Warmuth and A. K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *Proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, 1997.

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS*, 2009.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

A Proofs For Section 3.2

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

Lemma 14. *Let ψ be 1-strongly convex w.r.t. $\|\cdot\|$, so $\psi^\star$ is 1-strongly smooth with respect to $\|\cdot\|_\star$. Then,*

$$\|\nabla\psi^\star(z) - \nabla\psi^\star(z')\| \le \|z - z'\|_\star, \tag{31}$$

and

$$\arg\min_x g \cdot x + \psi(x) = \nabla\psi^\star(-g). \tag{32}$$

In order to prove Lemma 8, we first prove a somewhat easier result:

Lemma 15. *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*

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

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 10), Eq. (14) 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 14. □

Proof of Lemma 8. 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 15 to ϕ_1' and ϕ_2 completes the proof, noting for any x' we have $\phi_2(x_1) - \phi_2(x') \le \phi_2(x_1) - \phi_2(x_2)$. □

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

Table 1: Four equivalent expressions for unconstrained mirror descent. 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).

Proof of Corollary 9. Let $x_2' = \arg\min_x \phi_1(x) + \psi(x)$, so by Lemma 8, we have

$$\phi_1(x_1) + \psi(x_1) - \phi_1(x_2') - \psi(x_2') \leq \frac{1}{2}\|b\|_\star^2, \tag{33}$$

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

$$\begin{aligned} \phi_2(x_1) - \phi_2(x_2) &= \phi_1(x_1) + \psi(x_1) + \Psi(x_1) - \phi_1(x_2) - \psi(x_2) - \Psi(x_2) \\ &\leq \phi_1(x_1) + \psi(x_1) + \Psi(x_1) - \phi_1(x_2') - \psi(x_2') - \Psi(x_2) \\ &\leq \frac{1}{2}\|b\|_\star^2 + \Psi(x_1) - \Psi(x_2). && \text{Using Eq. (33).} \end{aligned}$$

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

B 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 $x_t \in \mathcal{X}$.

B.1 The Unconstrained Case

Table 1 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 14,

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

Table 1 presents four equivalent ways of writing the mirror descent update in this setting.

Theorem 16. *The four updates in Table 1 are equivalent.*

Proof. The proof is straightforward, but included for completeness.

- 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. (32). The optimality of x_t for this minimization implies $0 = -\theta_t + \nabla R(x_t)$, or $\nabla R(x_t) = \theta_t$.
- Explicit ⇔ FTRL: Immediate from Eq. (32) and the fact that $\theta_{t+1} = -g_{1:t}$.

- Implicit ⇔ FTRL: That is,

$$\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_R(x, \hat{x}_t) \qquad \text{and} \tag{34}$$

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x) \tag{35}$$

  are equivalent. The proof is straightforward by induction on the hypothesis $x_t = \hat{x}_t$. Namely, we must have from Eq. (34) and the IH that $g_t + \nabla R(\hat{x}_{t+1}) - \nabla R(x_t) = 0$, and from Eq. (35) 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. (35) at $\hat{x}_{t+1}$ is $g_{1:t} + \nabla R(\hat{x}_{t+1}) = 0$, and since the optimum of Eq. (35) 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.

□

B.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 play from a closed bounded set $\mathcal{X}$, or more generally if we add a non-smooth term Ψ. Some care is needed notationally in how we introduce the non-smooth term. For this section we take $R(x) = r(x) + \Psi(x)$ where r is continuously differentiable on $\operatorname{dom}\Psi$.

In this setting, the two explicit algorithms given in the previous table are, in fact, no longer equivalent. Table 2 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). When $\Psi = I_{\mathcal{X}}$, this corresponds to a greedy-projection algorithm; when further $r(x) = \frac{1}{2\eta}\|x\|_2^2$, this becomes the projected online gradient descent algorithm of Zinkevich (2003) for example. 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, where we assume $\Psi = I_{\mathcal{X}}$. In this case, we define the projection operator onto $\mathcal{X}$ with respect to Bregman divergence $\mathcal{B}_r$ by

$$\pi^r_{\mathcal{X}}(u) \equiv \arg\min_{x \in \mathcal{X}} \mathcal{B}_r(x, u).$$

Expanding the definition of the Bregman divergence and dropping terms independent of x, and replacing the explicit $x \in \mathcal{X}$ constraint with a $I_{\mathcal{X}}$ term in the objective, we have the equivalent expression

$$\pi^r_{\mathcal{X}}(u) = \arg\min_x r(x) - \nabla r(u) \cdot x + I_{\mathcal{X}}(x). \tag{36}$$

Both of these families can be analyzed in the general adaptive case using the techniques introduced in this paper. The Lazy family corresponds to the composite FTRL update (Section 4), namely

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

That is, we have a single fixed non-smooth penalty which arrives on the first round: $\alpha_1 = 1$ and $\alpha_t = 0$ for $t > 1$.

	Lazy	Greedy
Explicit	$\theta_{t+1} = \theta_t - g_t$ $x_{t+1} = \triangledown R^\star(\theta_{t+1})$	$\theta_{t+1} = \triangledown r(x_t) - g_t$ $x_{t+1} = \triangledown R^\star(\theta_{t+1})$
Implicit		$x_{t+1} =$ $\arg\min_x g_t \cdot x + \mathcal{B}_r(x, x_t) + \Psi(x)$
FTRL	$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x)$	Eq. (20) where $\tilde{r}_{0:t}(x) = r(x) + I_{\mathcal{X}}(x)$
Projection $\Psi = I_{\mathcal{X}}$	$u_{t+1} = \arg\min_x g_{1:t} \cdot x + r(x)$ $x_{t+1} = \pi^r_{\mathcal{X}}(u_{t+1})$	$x_{t+1} = \pi^r_{\mathcal{X}}(\triangledown r^\star(\triangledown r(x_t) - g_t))$

Table 2: Two families of mirror descent algorithms. The "Greedy" family is the traditional definition of mirror descent. Here $R(x) = r(x) + \Psi(x)$, where r is a differentiable strongly-convex regularizer.

The Greedy Mirror Descent algorithms, on the other hand, can be thought of us receiving loss functions $g_t \cdot x + \Psi(x)$ on each round: that is, we have $\alpha_t = 1$ for all t. This is the family we analyze in Section 5. Clearly, this is a quite different problem than in the Lazy case. In this setting, embedding $\Psi(x)$ inside R can be seen as a convenience for defining $\triangledown R^\star$. We have the following result:

Theorem 17. *The Lazy-Explicit and Lazy-FTRL updates are equivalent. When $\Psi = I_{\mathcal{X}}$, the Lazy-Projection update is also 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 becomes

$$\arg\min_x g_{1:t} \cdot x + R(x).$$

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

$$\begin{aligned} x_{t+1} = \pi^r_{\mathcal{X}}(u_{t+1}) &= \arg\min_x r(x) - \triangledown r(u_{t+1}) \cdot x + I_{\mathcal{X}}(x) && \text{Using Eq. (36).} \\ &= \arg\min_x g_{1:t} \cdot x + r(x) + I_{\mathcal{X}}(x), && \text{Since } \triangledown r(u_{t+1}) = -g_{1:t}. \end{aligned}$$

which is exactly the Lazy-FTRL update (recalling $R(x) = r(x) + \Psi(x)$).

□

Theorem 18. *The Explicit, Implicit, and FTRL updates in the "Greedy" column of Table 2 are equivalent. When $\Psi = I_{\mathcal{X}}$, the Greedy-Projection update is equivalent as well.*

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 played by the implicit version,

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

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

$$x_{t+1} = \arg\min_x -(\nabla r(x_t) - g_t) \cdot x + r(x) + \Psi(x), \tag{37}$$

which is equivalent.

- Greedy-Implicit ⇔ Greedy-FTRL: This is a special case of Theorem 12.
- When $\Psi = I_{\mathcal{X}}$, 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

$$\begin{aligned}x_{t+1} &= \pi^r_{\mathcal{X}}(u_{t+1}) \\ &= \arg\min_x r(x) - \nabla r(u_{t+1}) \cdot x + I_{\mathcal{X}}(x) && \text{Using Eq. (36).} \\ &= \arg\min_x r(x) - (\nabla r(x_t) - g_t) \cdot x + I_{\mathcal{X}}(x), && \text{Since } \nabla r(u_{t+1}) = \nabla r(x_t) - g_t.\end{aligned}$$

which is equivalent to the Greedy-Explicit update, e.g., Eq. (37).

□