A Unified View of Regularized Dual Averaging and Mirror Descent with Implicit Updates

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

September 21, 2011

arXiv:1009.3240v2 [cs.LG] 20 Sep 2011

Abstract

We study three families of online convex optimization algorithms: follow-the-proximally-regularized-leader (FTRL-Proximal), regularized dual averaging (RDA), and composite-objective mirror descent. We first prove equivalence theorems that show all of these algorithms are instantiations of a general FTRL update. This provides theoretical insight on previous experimental observations. In particular, even though the FOBOS composite mirror descent algorithm handles L_1 regularization explicitly, it has been observed that RDA is even more effective at producing sparsity. Our results demonstrate that FOBOS uses subgradient approximations to the L_1 penalty from previous rounds, leading to less sparsity than RDA, which handles the cumulative penalty in closed form. The FTRL-Proximal algorithm can be seen as a hybrid of these two, and outperforms both on a large, real-world dataset.

Our second contribution is a unified analysis which produces regret bounds that match (up to logarithmic terms) or improve the best previously known bounds. This analysis also extends these algorithms in two important ways: we support a more general type of composite objective and we analyze implicit updates, which replace the subgradient approximation of the current loss function with an exact optimization.

Keywords: online learning, online convex optimization, subgradient methods, regret bounds, follow-the-leader algorithms

1 Introduction

We consider the problem of online convex optimization, and in particular its application to online learning. On each round $t = 1, \dots, T$, we must pick a point $x_t \in \mathbb{R}^n$. A convex loss function f_t is then revealed, and we incur loss $f_t(x_t)$. Our regret at the end of T rounds with respect to a comparator point $\mathring{x}$ is

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

In Section 4 we provide a unified regret analysis of three prominent algorithms for online convex optimization. In recent years, these algorithms have received significant attention because they have straightforward and efficient implementations and offer state-of-the-art performance for many large-scale applications. In particular, we consider:

- Follow-the-Proximally-Regularized-Leader (FTPRL), introduced with adaptive learning rates (regularization) by McMahan and Streeter (2010).
- Regularized Dual Averaging (RDA), introduced by Xiao (2009) and extended with adaptive learning rates by Duchi et al. (2010a).
- Composite-Objective Mirror Descent (COMID) algorithms (Duchi et al., 2010b), including FOBOS (Duchi and Singer, 2009).

As pointed out by Duchi et al. (2010b), the analyses of RDA and COMID cited above are completely different. In contrast, we provide a unified analysis of these algorithms. One of our contributions is simply demonstrating that this large and important family of algorithms can be analyzed using a common argument, but our analysis also generalizes previous results in several important ways. First, we extend all of these algorithm to handle implicit updates, which replace the first-order approximation on the current loss function with an exact optimization. In many practical situations this update can be solved efficiently, and offers both theoretical and practical benefits compared to the first-order update.

We also extend the ability of these algorithms to handle composite objectives (objectives that include a fixed non-smooth term Ψ). Previous work considers loss functions on each round of the form $f_t(x) + \Psi(x)$, where f_t is approximated by a linear function, but the optimization over Ψ is exact. However, as discussed below, continuing to add a new copy of $\Psi(x)$ on each round may be undesirable in some cases; to address this, we analyze loss functions of the form $f_t(x) + \alpha_t \Psi(x)$ where α_t is a non-increasing sequence of non-negative numbers. This is useful, for example, if one wishes to encode a Bayesian prior in the online setting (see Section 2.2). Our proof technique has the advantage that handling this general form of composite updates requires only a few extra lines beyond the non-composite proof. The original analysis of FTPRL by McMahan and Streeter (2010) did not support composite updates. In addition to remedying this, we prove a new stronger version of the "FTRL/BTL Lemma" which tightens the analysis of FTPRL by a constant factor. The new lemma is quite general and may be of independent interest.

Our unified analysis relies on a formulation of all of these algorithms as instances of follow-the-regularized-leader, which we develop in Section 3. A preliminary version of these equivalence results appeared in (McMahan, 2010). Our equivalence theorems apply to algorithms that use arbitrary strongly convex regularization; however, these results show that the most interesting strict equivalences occur in the case of quadratic regularization. Thus, for the analysis of Section 4 we restrict attention to this case, namely to algorithms where the incremental strong convexity is of the form

$$R_t(x) = \frac{1}{2}\left\|Q_t^{\frac{1}{2}}(x - y)\right\|_2^2$$

where $y \in R^n$ and Q_t is a positive-semidefinite matrix. This is less general than previous results in terms of arbitrary strongly-convex functions or Bregman divergences.

Application to Sparse Models via L_1 Regularization On the surface, follow-the-regularized-leader algorithms like regularized dual averaging (Xiao, 2009) appear quite different from gradient descent (and more generally, mirror descent) style algorithms like FOBOS (Duchi and Singer, 2009). However, the results of Section 3 show that in the case of quadratic stabilizing regularization there are only two differences between the algorithms:

- How they choose to center the additional strong convexity used to guarantee low regret: RDA centers this regularization at the origin, while FOBOS centers it at the current feasible point.
- How they handle an arbitrary non-smooth regularization function Ψ. This includes the mechanism of projection onto a feasible set and how L_1 regularization is handled.

To make these differences precise while also illustrating that these families are actually closely related, we consider a third algorithm, FTRL-Proximal. When the non-smooth term Ψ is omitted, this algorithm is in fact identical to FOBOS. On the other hand, its update is essentially the same as that of dual averaging, except that additional strong convexity is centered at the current feasible point (see Table 1).

Previous work has shown experimentally that dual averaging with L_1 regularization is much more effective at introducing sparsity than FOBOS (Xiao, 2009, Duchi et al., 2010a). Our equivalence theorems provide a theoretical explanation for this: while RDA considers the cumulative L_1 penalty $t\lambda\|x\|_1$ on round t, FOBOS (when viewed as a global optimization using our equivalence theorem) considers $\phi_{1:t-1} \cdot x + \lambda\|x\|_1$, where ϕ_s is a certain subgradient approximation of $\lambda\|x_s\|_1$ (we use $\phi_{1:t-1}$ as shorthand for $\sum_{s=1}^{t-1} \phi_s$, and extend the notation to sums over matrices and functions as needed).

An experimental comparison of FOBOS, RDA, and FTRL-Proximal, presented in Section 5, demonstrates the validity of the above explanation. The FTRL-Proximal algorithm behaves very similarly to RDA in terms of sparsity, confirming that it is the cumulative subgradient approximation to the L_1 penalty that causes decreased sparsity in FOBOS.

In recent years, online gradient descent and stochastic gradient descent (its batch analogue) have proven themselves to be excellent algorithms for large-scale machine learning. In the simplest case FTRL-Proximal is identical, but when L_1 or other non-smooth regularization is needed, FTRL-Proximal significantly outperforms FOBOS, and can outperform RDA as well. Since the implementations of FTRL-Proximal and RDA only differ by a few lines of code, we recommend trying both and picking the one with the best performance in practice.

2 Algorithms and Regret Bounds

We begin by establishing notation and introducing more formally the algorithms we consider. We consider loss functions $f_t(x) + \alpha_t \Psi(x)$, where Ψ is a fixed (typically non-smooth) regularization function. In a typical online learning setting, given an example (θ_t, y_t) where $\theta_t \in \mathbb{R}^n$ is a feature vector and $y_t \in \{-1, 1\}$ is a label, we take $f_t(x) = \text{loss}(\theta_t \cdot x, y_t)$. For example, for logistic regression we use log-loss, $\text{loss}(\theta_t \cdot x, y_t) = \log(1 + \exp(-y_t \theta_t \cdot x))$. All of the algorithms we consider support composite updates (consideration of Ψ explicitly rather than through a gradient $\triangledown f_t(x_t)$) as well as positive semi-definite matrix learning rates Q which can be chosen adaptively (the interpretation of these matrices as learning rates will be clarified in Section 3).

We first consider the specific algorithms used in the L_1 experiments of Section 5; we use the standard reduction to linear functions, letting $g_t = \triangledown f_t(x_t)$. The first algorithm we consider is from the gradient-descent family, namely FOBOS, which plays

$$x_{t+1} = \arg\min_x g_t \cdot x + \lambda\|x\|_1 + \frac{1}{2}\|Q_{1:t}^{\frac{1}{2}}(x - x_t)\|_2^2.$$

		(A)		(B)	(C)
COMID	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\phi_{1:t-1} \cdot x + \alpha_t \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}}(x - x_s) \rVert^2$
RDA	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\alpha_{1:t} \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}}(x - 0) \rVert^2$
FTPRL	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\alpha_{1:t} \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}}(x - x_s) \rVert^2$
AOGD	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\phi_{1:t-1} \cdot x + \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}}(x - 0) \rVert_2^2$

Table 1: The algorithms considered in this paper, expressed as particular instances of the update of Eq. (3). The fact that we can express COMID and adaptive online gradient descent (AOGD) in this way is a consequence of Theorems 4 and 7. Each algorithms' objective has three components: (A) An approximation to the sum of previous loss functions $f_{1:t}$, where the first $t-1$ functions are approximated by linear terms, and f_t is included exactly (exactly including f_t make the updates implicit). (B) Terms for the non-smooth composite terms $\alpha_t \Psi$. COMID approximates the terms for $\alpha_{1:t-1}\Psi$ by subgradients, while RDA and FTPRL consider them exactly. And finally, (C), stabilizing regularization needed to ensure low regret.

We state this algorithm implicitly as an optimization, but a gradient-descent style closed-form update can also be given (Duchi and Singer, 2009). The algorithm was described in this form as a specific composite-objective mirror descent (COMID) algorithm by Duchi et al. (2010b).

The regularized dual averaging (RDA) algorithm of Xiao (2009) plays

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + t\lambda \lVert x \rVert_1 + \frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}}(x - 0) \rVert_2^2.$$

In contrast to FOBOS, the RDA optimization is over the sum $g_{1:t}$ rather than just the most recent gradient g_t. We will show (in Theorem 7) that when $\lambda = 0$ and the f_t are not strongly convex, this algorithm is in fact equivalent to the adaptive online gradient descent (AOGD) algorithm of Bartlett et al. (2007).

RDA is directly defined as a FTRL algorithm, and hence is also an instance of the more general primal-dual algorithmic schema of Shalev-Shwartz and Singer (2006); see also Kakade et al. (2009). However, these general results are not sufficient to prove the original bounds for RDA, nor the versions here that extend to implicit updates.

The FTRL-Proximal algorithm plays

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + t\lambda \lVert x \rVert_1 + \frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}}(x - x_s) \rVert_2^2.$$

This algorithm was introduced by McMahan and Streeter (2010), but without support for an explicit Ψ.

One of our principle contributions is showing the close connection between all four of these algorithms; Table 1 summarizes the key results from Theorems 4 and 7, writing AOGD and FOBOS in a form that makes the relationship to RDA and FTRL-Proximal explicit.

In our equivalence analysis, we will consider arbitrary convex functions R_t and $\tilde{R}_t$ in place of the $\frac{1}{2}\left\|Q_t^{\frac{1}{2}}x\right\|_2^2$ and $\frac{1}{2}\left\|Q_t^{\frac{1}{2}}(x - x_t)\right\|_2^2$ that appear here, as well as arbitrary convex $\Psi(x)$ in place of $\lambda\|x\|_1$.

2.1 Implicit and Composite Updates for FTRL

The algorithms we consider can be expressed as follow-the-regularized-leader (FTRL) algorithms that perform implicit and composite updates. The standard subgradient FTRL algorithm uses the update

$$x_{t+1} = \arg\min_x \left(\sum_{s=1}^{t} \triangledown f_s(x_s) \right) \cdot x + R_{1:t}(x).$$

In this update, each previous (potentially non-linear) loss function f_s is approximated by the gradient at x_s (when f_s is not differentiable, we can use a subgradient at x_s in place of the gradient). The functions R_t are incremental regularization added on each round; for example $R_{1:t}(x) = \sqrt{t}\|x\|^2$ is a standard choice, corresponding to regularized dual averaging.

Implicit update rules are usually defined for mirror descent algorithms, but we can define an analogous update for FTRL:

$$x_{t+1} = \arg\min_x \left(\sum_{s=1}^{t-1} \triangledown f_s(x_{s+1}) \right) \cdot x + f_t(x) + R_{1:t}(x).$$

This update replaces the subgradient approximation of f_t with the possibly non-linear f_t. Closed-form implicit updates for the squared error case were derived by (Kivinen and Warmuth, 1997); the term implicit updates was coined later (Kivinen et al., 2006). Our formulation is similar to the online coordinate-dual-ascent algorithm briefly mentioned by Shalev-Shwartz and Kakade (2008). In general, computing the implicit update might require solving an arbitrary convex optimization problem (hence, the name implicit), however, in many useful applications it can be computed in closed form or by optimizing a one-dimensional problem. We discuss the advantages of implicit updates in Section 2.2.

Analysis of implicit updates has proved difficult. Kulis and Bartlett (2010) provide the only other regret bounds for implicit updates that match those of the explicit-update versions. While their analysis handles more general divergences, it only applies to mirror-descent algorithms. Our analysis handles composite objectives and applies FTRL algorithms as well as mirror descent. Our analysis also quantifies the one-step improvement in the regret bound obtained by the implicit update, showing the inequality is in fact strict when the implicit update is non-trivial.

When f_t is not differentiable, we use the update

$$x_{t+1} = \arg\min_x g'_{1:t-1} \cdot x + f_t(x) + R_{1:t}(x), \tag{1}$$

where g'_t is a subgradient of f_t at x_{t+1} (that is, $g'_t \in \partial f_t(x_{t+1})$) such that $g'_{1:t-1} + g'_t + \triangledown R_{1:t}(x_{t+1}) = 0$. The existence of such a subgradient is proved below, in Theorem 3.

In many applications, we have a fixed convex function Ψ that we also wish to include in the optimization, for example $\Psi(x) = \|x\|_1$ (L_1-regularization to induce sparsity) or the indicator function on a feasible set $\mathcal{F}$ (see Section 2.4). While it is possible to approximate this function via subgradients as well, when computationally feasible it is often better to handle Ψ directly. For example, in the case where $\Psi(x) = \|x\|_1$, subgradient approximations will in general not lead to sparse solutions. In this case, closed-form updates for optimizations including Ψ are often possible, and produce much better sparsity (Xiao, 2009, Duchi and Singer, 2009). We can include such a term directly in FTRL, giving the composite objective update

$$x_{t+1} = \arg\min_x \left(\sum_{s=1}^{t} \nabla f_s(x_s) \right) \cdot x + \alpha_{1:t}\Psi(x) + R_{1:t}(x), \tag{2}$$

where α_t is the weight on Ψ on round t.

This formulation, which allows for an arbitrary sequence of non-negative, non-increasing α_t's, is more general than that supported by the original analysis of COMID or RDA. Xiao (2010, Sec 6.1) shows that RDA does allow a varying schedule where $\alpha_{1:t} = c + 1/\sqrt{t}$ for a constant c, by incorporating part of the Ψ term in the regularization function R_t; this is less general than our analysis, which allows the schedule α_t to be chosen independently of the learning rate.

Finally, we can combine these ideas to define an implicit update with a composite objective. In the general case where f_t is not differentiable, we have the update

$$x_{t+1} = \arg\min_x g'_{1:t-1} \cdot x + f_t(x) + \alpha_{1:t}\Psi(x) + R_{1:t}(x), \tag{3}$$

where $g'_t \in \partial f_t(x_{t+1})$ such that $\exists \phi_t \in \partial\Psi(x_{t+1})$ where $g'_{1:t-1} + g'_t + \phi_t + \nabla R_{1:t}(x_{t+1}) = 0$. The existence of such a subgradient again follows from Theorem 3.

It is worth noting that our analysis of implicit updates applies immediately to standard first-order updates. Let f_t^w designate the loss function provided by the *world*, and let f_t^u be the loss function in the *update* Eq. (3). Then we recover the non-implicit algorithms by taking $f_t^u(x) \leftarrow \nabla f_t^w(x_t) \cdot x$.

2.2 Motivation for Implicit Updates and Composite Objectives

Implicit updates offer a number of advantages over using a subgradient approximation. Kulis and Bartlett (2010) discusses several important examples. They also observe that empirically, implicit updates outperform or nearly outperform linearized updates, and show more robustness to scaling of the data.

Learning problems that use importance weights on examples are also a good candidate for implicit updates. Importance weights can be used to compress the training data, by replacing n copies of an example with one copy with weight n. They also arise in active learning algorithms (Beygelzimer et al., 2010) and situations where the training and test distributions differ (covariate shift, e.g. Sugiyama et al. (2008)). Recent work has demonstrated experimentally that implicit updates can significantly outperform first-order updates both on importance weighted and standard learning problems (Karampatziakis and Langford, 2010).

The following simple examples demonstrates the intuition for these improvements. The key is that the linearization of f_t over-estimates the decrease in loss under f_t achieved by moving in the direction $\nabla f_t(x_t)$. The farther x_{t+1} is chosen from x_t, and the more non-linear the f_t, the worse this approximation can be. Consider gradient descent in one dimension with $f_t(x) = \frac{1}{2}(x-3)^2$ and $x_t = 2$. Then $\nabla f_t(2) = -1$, and if we choose a learning rate $\eta_t > 1$, we will actually overshoot the optimum for f_t (such a learning rate could be indicated by the theory if the feasible set is large, for example). Implicit updates, on the other hand, will never choose $x_{t+1} > 3$, rather $x_{t+1} \to 3$ as $\eta_t \to \infty$. Thus, we see implicit updates can be significantly better behaved with large learning rates. Note that an importance weight of n is equivalent to multiplying the learning rate by n, so when importance weights can be large, implicit updates can be particularly beneficial.

The overshooting issue is even more pronounced with non-smooth objectives, for example, $f_t(x) = g \cdot x + \|x\|_1$. A standard gradient descent update will in general never set $x_{t+1} = 0$ despite the L_1 regularization; handling the L_1 term via an implicit update solves this problem. This is exactly the insight that COMID algorithms like FOBOS exploit; by analyzing general implicit updates, we achieve an analysis of these algorithms while also supporting a much larger class of updates.

When the functional form of the non-smooth component of the objective (for example $\|x\|_1$) is fixed across rounds, it is preferable to perform an explicit optimization involving the total non-smooth contribution $\alpha_{1:t}\Psi$ (RDA and FTPRL) rather than just the round t contribution $\alpha_t \Psi$ (COMID). While RDA supports this type of non-smooth objective, it requires the weight on Ψ to be fixed across rounds. We generalize this to non-increasing per-round contributions in this work.

Suppose one is performing online logistic regression, and believes a priori that the coefficients have a Laplacian distribution. Then, L_1-penalized logistic regression corresponds to MAP estimation (e.g., Lee et al. (2006)); suppose the prior corresponds to a total penalty of $\lambda\|x\|_1$. If the size of the dataset T is known in advance, then we can use $\alpha_t = \lambda/T$, and by making multiple passes over the data, we will converge to the MAP estimate. However, in the online setting we will in general not know T in advance, and we may wish to use an online algorithm for computational efficiency. In this case, any fixed value of α_t will correspond to strengthening the prior each time we see a new example, which is undesirable. With the generalized notion of composite updates introduced here, this problem is overcome by choosing $\alpha_1 \Psi(x) = \lambda\|x\|_1$, and $\alpha_t = 0$ for $t \geq 2$. Thus, the fixed penalty on the coefficients is correctly encoded, independent of T.

2.3 Summary of Regret Bounds

In Section 4, we analyze the update rule of Equation (3) when

$$R_t(x) = \frac{1}{2}\left\|Q_t^{\frac{1}{2}}(x - y_t)\right\|^2, \tag{4}$$

where $\|\cdot\| = \|\cdot\|_2$ here and throughout. The points $y_t \in \mathbb{R}^n$ are the centers for the additional regularization added on each round. Choosing $y_t = 0$ leads to an analysis of RDA with implicit updates, and choosing $y_t = x_t$ yields the follow-the-proximally-regularized-leader algorithm with implicit updates. Using $y_t = x_t$ together with a modified choice of f_t leads to composite-objective mirror descent (see Section 3).

The generalized learning rates Q_t can be chosen adaptively using techniques from McMahan and Streeter (2010) and Duchi et al. (2010a), which leads to improved regret bounds, as well as algorithms that perform much better in practice (Streeter and McMahan, 2010). Since in this work we provide suitable regret bounds in terms of arbitrary Q_t, the adaptive techniques can be applied directly. Doing so complicates the exposition somewhat, and so for simplicity and easy of comparison to previous results we state specific regret bounds for scalar learning rates:

Corollary 1. *Let Ψ be the indicator function on a feasible set $\mathcal{F}$, and let $D = \max_{a,b\in\mathcal{F}} \|a - b\|$. So that our bounds are comparable, suppose $\max_{a\in\mathcal{F}} \|a\| = \frac{D}{2}$ (for example, if $\mathcal{F}$ is* *symmetric). Let f_t be a sequence of convex loss functions such that $\|\nabla f_t(x)\| \leq G$ for all t and all $x \in \mathcal{F}$. Then for FTPRL we set $x_t = y_t$ and have*

$$\textit{Regret} \leq DG\sqrt{2T}.$$

Implicit-update mirror descent obtains the same bound. For regularized dual averaging we choose $y_t = 0$ for all t, and obtain

$$\textit{Regret} \leq \frac{1}{2}DG\sqrt{2T} + \frac{GD}{\sqrt{2}}\ln T + \mathcal{O}(1).$$

These bounds are achieved with an adaptive learning rate that depends only on t (T need not be known in advance). If T is known, then the $\sqrt{2}$ constant on the $\sqrt{T}$ terms can be eliminated. The regret bounds with per-coordinate adaptive rates are at least as good, and often better. This corollary is a direct consequence of the following general result:

Theorem 2. *Let Ψ be an extended convex function on $\mathbb{R}^n$ with $\Psi(x) \geq 0$ and $0 \in \partial\Psi(0)$, let f_t be a sequence of convex loss functions, and let $\alpha_t \in \mathbb{R}$ be non-negative and non-increasing real numbers ($0 \leq \alpha_{t+1} \leq \alpha_t$). Consider the FTRL algorithm that plays $x_1 = 0$ and afterwards plays according to Equation* (3)*,*

$$x_{t+1} = \arg\min_x g'_{1:t-1} \cdot x + f_t(x) + \alpha_{1:t}\Psi(x) + R_{1:t}(x),$$

using incremental quadratic regularization functions $R_t(x) = \frac{1}{2}\left\|Q_t^{\frac{1}{2}}(x - y_t)\right\|^2$ where $Q_1 \in S^n_{++}$, $Q_t \in S^n_+$ for $t > 1$, and $y_t \in \mathbb{R}^n$. Then there exist $\tilde{g}_t \in R^n$ such that

$$\begin{aligned}\textit{Regret}(f) &\leq R_{1:T}(\mathring{x}) + \alpha_{1:T}\Psi(\mathring{x}) + \sum_{t=1}^{T}(g_t - \frac{1}{2}\tilde{g}_t)^\top Q_{1:t}^{-1}\tilde{g}_t - g_t Q_{1:t}^{-1} Q_t(y_t - x_t) \\ &\leq R_{1:T}(\mathring{x}) + \alpha_{1:T}\Psi(\mathring{x}) + \sum_{t=1}^{T}\frac{1}{2}\left\|Q_{1:t}^{-\frac{1}{2}}g_t\right\|^2 - \delta_{1:t} - g_t Q_{1:t}^{-1} Q_t(y_t - x_t)\end{aligned}$$

versus any point $\mathring{x} \in \mathbb{R}^n$, for any $g_t \in \partial f_t(x_t)$, with $\delta \geq 0$.

We will show that $\tilde{g}_t$ is a certain subgradient of f_t, and in fact when all $\alpha_t = 0$, then $\tilde{g}_t \in \partial f_t(x_{t+1})$. If f_t is strictly convex, then in general $\tilde{g}_t \neq g_t$, and so the inequality between the first and second bounds can be strict; in fact, we will show that on rounds t where the implicit-update is non-trivial, $\delta > 0$, indicating a one-step advantage for implicit updates. When all $\alpha_t = 0$, δ_t is one-half the improvement in the objective function of Equation (3) obtained by solving for the optimum point rather than using a solution from the linearized problem; the proof of Lemma 11 makes this precise.

For RDA, we take $y_t = 0$, and for FTPRL and implicit-update mirror descent we take $y_t = x_t$. Since no restrictions are placed on the y_t in the theorem, the final right-hand term being subtracted could have be positive, negative, or zero.

If we treat $\alpha_t\Psi$ as an intrinsic part of the problem, that is, we are measuring loss against $f_t(x) + \alpha_t\Psi(x)$, then the $\alpha_{1:T}\Psi(\mathring{x})$ term disappears from the regret bound.

2.4 Notation and Technical Background

We use the notation $g_{1:t}$ as a shorthand for $\sum_{s=1}^t g_s$. Similarly we write $Q_{1:t}$ for a sum of matrices Q_t, and we use $f_{1:t}$ to denote the function $f_{1:t}(x) = \sum_{s=1}^t f_s(x)$. We assume the summation binds more tightly than exponents, so $Q^{\frac{1}{2}}_{1:t} = (Q_{1:t})^{\frac{1}{2}}$. We write $x^\top y$ or $x \cdot y$ for the inner product between $x, y \in \mathbb{R}^n$. We write "the functions f_t" for the sequence of functions $(f_1, \ldots, f_T)$.

We write S^n_+ for the set of symmetric positive semidefinite $n \times n$ matrices, with S^n_{++} the corresponding set of symmetric positive definite matrices. Recall $A \in S^n_{++}$ means $\forall x \neq 0,\ x^\top A x > 0$. Since $A \in S^n_+$ is symmetric, $x^\top A y = y^\top A x$ (we often use this result implicitly). For $B \in S^n_+$, we write $B^{1/2}$ for the square root of B, the unique $X \in S^n_+$ such that $XX = B$ (see, for example, Boyd and Vandenberghe (2004, A.5.2)).

Unless otherwise stated, convex functions are assumed to be extended, with domain $\mathbb{R}^n$ and range $\mathbb{R} \cup \{\infty\}$ (see, for example (Boyd and Vandenberghe, 2004, 3.1.2)). For a convex function f, we let $\partial f(x)$ denote the set of subgradients of f at x (the subdifferential of f at x). By definition, $g \in \partial f(x)$ means $f(y) \geq f(x) + g^\top(y - x)$ for all y. When f is differentiable, we write $\triangledown f(x)$ for the gradient of f at x. In this case, $\partial f(x) = \{\triangledown f(x)\}$. All mins and argmins are over $\mathbb{R}^n$ unless otherwise noted. We make frequent use of the following standard results, summarized as follows:

Theorem 3. *Let $R : \mathbb{R}^n \to \mathbb{R}$ be strongly convex with continuous first partial derivatives, and let Φ and f be arbitrary (extended) convex functions. Then,*

A. Let $U(x) = R(x) + \Phi(x)$. Then, there exists a unique pair (x^*, ϕ^*) such that both

$$\phi^* \in \partial\Phi(x^*) \qquad \textit{and} \qquad x^* = \arg\min_x R(x) + \phi^* \cdot x.$$

Further, this x^* is the unique minimizer of U, and $\triangledown R(x^*) + \phi^* = 0$.

B. Let $V(x) = R(x) + \Phi(x) + f(x)$ and $\mathring{x} = \arg\min_x V(x)$. Then, there exists a $g \in \partial f(\mathring{x})$ such that

$$\mathring{x} = \arg\min_x R(x) + \Phi(x) + g \cdot x.$$

Proof. First we consider part A. Since R is strongly convex, U is strongly convex, and so has a unique minimizer x^* (see for example, (Boyd and Vandenberghe, 2004, 9.1.2)). Let $r = \triangledown R$. Since x^* is a minimizer of U, there must exist a $\phi^* \in \partial\Phi(x^*)$ such that $r(x^*) + \phi^* = 0$, as this is a necessary (and sufficient) condition for $0 \in \partial U(x^*)$. It follows that $x^* = \arg\min_x R(x) + \phi^* \cdot x$, as $r(x^*) + \phi^*$ is the gradient of this objective at x^*. Suppose some other (x', ϕ') satisfies the conditions of the theorem. Then, $r(x') + \phi' = 0$, and so $0 \in \partial U(x')$, and so x' is a minimizer of U. Since this minimizer is unique, $x' = x^*$, and $\phi' = -r(x^*) = \phi^*$. An equivalent condition to $x^* = \arg\min_x R(x) + \phi^* \cdot x$ is $\triangledown R(x^*) + \phi^* = 0$.

For part B, by definition of optimality, there exists a $\phi \in \partial\Phi(\mathring{x})$ and a $g \in \partial f(\mathring{x})$ such that $g + \phi + \triangledown R(\mathring{x}) = 0$. Choosing this g, define

$$\hat{x} = \arg\min_x R(x) + \Phi(x) + g \cdot x.$$

Applying part A with $R(x) \leftarrow R(x) + g \cdot x$, there exists a unique pair $(\hat{x}, \hat{\phi})$ such that $\hat{\phi} \in \partial\Phi(\hat{x})$ and $\triangledown R(\hat{x}) + \hat{\phi} + g = 0$. Since $(\mathring{x}, \phi)$ satisfy this equation, we conclude $\mathring{x} = \hat{x}$. $\square$

Feasible Sets In some applications, we may be restricted to only play points from a convex feasible set $\mathcal{F} \subseteq \mathbb{R}^n$, for example, the set of (fractional) paths between two nodes in a graph. A feasible set is also necessary to prove regret bounds against linear functions.

With composite updates, Equations (2) and (3), this is accomplished for free by choosing Ψ to be the indicator function $I_{\mathcal{F}}$ on $\mathcal{F}$, where $I_{\mathcal{F}}(x) = 0$ for $x \in \mathcal{F}$ and ∞ otherwise. It is straightforward to verify that

$$\arg\min_{x \in \mathbb{R}^n} g_{1:t} \cdot x + R_{1:t}(x) + I_{\mathcal{F}}(x) \quad = \quad \arg\min_{x \in \mathcal{F}} g_{1:t} \cdot x + R_{1:t}(x),$$

and so in this work we can generalize (for example) the results of (McMahan and Streeter, 2010) for specific feasible sets without specifically discussing $\mathcal{F}$, and instead considering arbitrary extended convex functions Ψ. Note that in this case the choice of α_t does not matter as long as $\alpha_1 > 0$.

3 Mirror Descent Follows The Leader

In this section we consider the relationship between mirror descent algorithms (the simplest example being online gradient descent) and FTRL algorithms. Let $f_t(x) = g_t \cdot x + \Psi(x)$.

Let R_1 be strongly convex, with all the R_t convex. We assume that $\min_x R_1(x) = 0$, and assume that $x = 0$ is the unique minimizer unless otherwise noted.

Follow The Regularized Leader (FTRL) The simplest follow-the-regularized-leader algorithm plays

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + \frac{\sigma_{1:t}}{2} \|x\|_2^2, \tag{5}$$

where $\sigma_{1:t} \in \mathbb{R}$ is the amount of stabilizing strong convexity added.

A more general update is

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

where we add an additional convex function R_t on each round. When $\arg\min_{x \in \mathbb{R}^n} R_t(x) = 0$, we call the functions R_t (and associated algorithms) *origin-centered*. We can also define *proximal* versions of FTRL[1] that center additional regularization at the current point rather than at the origin. In this section, we write $\tilde{R}_t(x) = R_t(x - x_t)$ and reserve the R_t notation for origin-centered functions. Note that $\tilde{R}_t$ is only needed to select x_{t+1}, and x_t is known to the algorithm at this point, ensuring the algorithm only needs access to the first t loss functions when computing x_{t+1} (as required).

Mirror Descent The simplest version of mirror descent is gradient descent using a constant step size η, which plays

$$x_{t+1} = x_t - \eta g_t = -\eta g_{1:t}. \tag{6}$$

In order to get low regret, T must be known in advance so η can be chosen accordingly (or a doubling trick can be used). But, since there is a closed-form solution for the point x_{t+1} in terms of $g_{1:t}$ and η, we generalize this to a "revisionist" algorithm that on each round plays the point that gradient descent with constant step size would have played if it had used step size η_t on rounds 1 through $t - 1$. That is, $x_{t+1} = -\eta_t g_{1:t}$. When $R_t(x) = \frac{\sigma_t}{2}\|x\|_2^2$ and $\eta_t = \frac{1}{\sigma_{1:t}}$, this is equivalent to the FTRL of Equation (5).

[1] We adapt the name "proximal" from (Do et al., 2009), but note that while similar proximal regularization functions were considered, that paper deals only with gradient descent algorithms, not FTRL.

In general, we will be more interested in gradient descent algorithms which use an adaptive step size that depends (at least) on the round t. Using a variable step size η_t on each round, gradient descent plays:

$$x_{t+1} = x_t - \eta_t g_t. \tag{7}$$

An intuition for this update comes from the fact it can be re-written as

$$x_{t+1} = \arg\min_x g_t \cdot x + \frac{1}{2\eta_t}\|x - x_t\|_2^2.$$

This version captures the notion (in online learning terms) that we don't want to change our hypothesis x_t too much (for fear of predicting badly on examples we have already seen), but we do want to move in a direction that decreases the loss of our hypothesis on the most recently seen example. Here, this is approximated by the linear function g_t, but implicit updates use the exact loss f_t.

Mirror descent algorithms use this intuition, replacing the L_2-squared penalty with an arbitrary Bregman divergence. For a differentiable, strictly convex R, the corresponding Bregman divergence is

$$\mathcal{B}_R(x, y) = R(x) - \big(R(y) + \triangledown R(y) \cdot (x - y)\big)$$

for any $x, y \in \mathbb{R}^n$. We then have the update

$$x_{t+1} = \arg\min_x g_t \cdot x + \frac{1}{\eta_t}\mathcal{B}_R(x, x_t), \tag{8}$$

or explicitly (by setting the gradient of (8) to zero),

$$x_{t+1} = r^{-1}(r(x_t) - \eta_t g_t) \tag{9}$$

where $r = \triangledown R$. Letting $R(x) = \frac{1}{2}\|x\|_2^2$ so that $\mathcal{B}_R(x, x_t) = \frac{1}{2}\|x - x_t\|_2^2$ recovers the algorithm of Equation (7). One way to see this is to note that $r(x) = r^{-1}(x) = x$ in this case.

We can generalize this even further by adding a new strongly convex function R_t to the Bregman divergence on each round. Namely, let

$$\mathcal{B}_{1:t}(x, y) = \sum_{s=1}^{t} \mathcal{B}_{R_s}(x, y),$$

so the update becomes

$$x_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_{1:t}(x, x_t) \tag{10}$$

or equivalently $x_{t+1} = (r_{1:t})^{-1}(r_{1:t}(x_t) - g_t)$ where $r_{1:t} = \sum_{s=1}^{t} \triangledown R_t = \triangledown R_{1:t}$ and $(r_{1:t})^{-1}$ is the inverse of $r_{1:t}$. The step size η_t is now encoded implicitly in the choice of R_t.

Composite-objective mirror descent (COMID) (Duchi et al., 2010b) handles Ψ functions[2] as part of the objective on each round: $f_t(x) = g_t \cdot x + \Psi(x)$. Using our notation, the COMID update is

$$x_{t+1} = \arg\min_x \eta g_t \cdot x + \mathcal{B}(x, x_t) + \eta\Psi(x),$$

which can be generalized to

$$x_{t+1} = \arg\min_x g_t \cdot x + \Psi(x) + \mathcal{B}_{1:t}(x, x_t), \tag{11}$$

[2] Our Ψ is denoted r in (Duchi et al., 2010b)

where the learning rate η has been rolled into the definition of $R_1, \dots, R_t$. When Ψ is chosen to be the indicator function on a convex set, COMID reduces to standard mirror descent with greedy projection.

3.1 An Equivalence Theorem for Proximal Regularization

The following theorem shows that mirror descent algorithms can be viewed as FTRL algorithms:

Theorem 4. *Let R_t be a sequence of differentiable origin-centered convex functions ($\triangledown R_t(0) = 0$), with R_1 strongly convex, and let Ψ be an arbitrary convex function. Let $x_1 = \hat{x}_1 = 0$. For a sequence of loss functions $f_t(x) + \Psi(x)$, let the sequence of points played by the implicit-update composite-objective mirror descent algorithm be*

$$\hat{x}_{t+1} = \arg\min_x \; f_t(x) + \alpha_t \Psi(x) + \tilde{\mathcal{B}}_{1:t}(x, \hat{x}_t), \tag{12}$$

where $\tilde{R}_t(x) = R_t(x - \hat{x}_t)$, and $\tilde{\mathcal{B}}_t = \mathcal{B}_{\tilde{R}_t}$, so $\tilde{\mathcal{B}}_{1:t}$ is the Bregman divergence with respect to $\tilde{R}_1 + \cdots + \tilde{R}_t$. Consider the alternative sequence of points x_t played by a proximal FTRL algorithm, applied to these same f_t, defined by

$$x_{t+1} = \arg\min_x \; (g'_{1:t-1} + \phi_{1:t-1}) \cdot x + f_t(x) + \alpha_t \Psi(x) + \tilde{R}_{1:t}(x) \tag{13}$$

for some $g'_t \in \partial f_t(x_{t+1})$ and $\phi_t \in \partial(\alpha_t \Psi)(x_{t+1})$. Then, these algorithms are equivalent, in that $x_t = \hat{x}_t$ for all $t > 0$.

We defer the proof to the end of this section. The Bregman divergences used by mirror descent in the theorem are with respect to the proximal functions $\tilde{R}_{1:t}$, whereas typically (as in Equation (10)) these functions would not depend on the previous points played. We will show when $R_t(x) = \frac{1}{2}\|Q_t^{\frac{1}{2}} x\|_2^2$, this issue disappears. Considering arbitrary Ψ functions and implicit updates also complicates the theorem statement somewhat. The following corollary sidesteps these complexities, to state a simple direct equivalence result:

Corollary 5. *Let $f_t(x) = g_t \cdot x$. Then, the following algorithms play identical points:*

- *Gradient descent with positive semi-definite learning rates Q_t, defined by:*

$$x_{t+1} = x_t - Q_{1:t}^{-1} g_t.$$

- *FTRL-Proximal with regularization functions $\tilde{R}_t(x) = \frac{1}{2}\left\|Q_t^{\frac{1}{2}}(x - x_t)\right\|_2^2$, which plays*

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

Proof. Let $R_t(x) = \frac{1}{2} x^\top Q_t x$. It is easy to show that $R_{1:t}$ and $\tilde{R}_{1:t}$ differ by only a linear function, and so (by a standard result) $\mathcal{B}_{1:t}$ and $\tilde{\mathcal{B}}_{1:t}$ are equal, and simple algebra reveals

$$\mathcal{B}_{1:t}(x, y) = \tilde{\mathcal{B}}_{1:t}(x, y) = \frac{1}{2}\|Q_{1:t}^{\frac{1}{2}}(x - y)\|_2^2.$$

Then, it follows from Equation (9) that the first algorithm is a mirror descent algorithm using this Bregman divergence. Taking $\Psi(x) = 0$ and hence $\phi_t = 0$, the result follows from Theorem 4. □

Extending the approach of the corollary to FOBOS, we see the only difference between that algorithm and FTRL-Proximal is that FTRL-Proximal optimizes over $t\Psi(x)$, whereas in Equation (13) we optimize over $\phi_{1:t-1} \cdot x + \Psi(x)$ (see Table 1). Thus, FOBOS is equivalent to FTRL-Proximal, except that FOBOS approximates all but the most recent Ψ function by a subgradient.

The behavior of FTRL-Proximal can thus be different from COMID when a non-trivial Ψ is used. While we are most concerned with the choice $\Psi(x) = \lambda\|x\|_1$, it is also worth considering what happens when Ψ is the indicator function on a feasible set $\mathcal{F}$. Then, Theorem 4 shows that mirror descent on $f_t(x) = g_t \cdot x + \Psi(x)$ (equivalent to COMID in this case) approximates previously seen Ψs by their subgradients, whereas FTRL-Proximal optimizes over Ψ explicitly. In this case, it can be shown that the mirror-descent update corresponds to the standard greedy projection (Zinkevich, 2003), whereas FTRL-Proximal corresponds to a lazy projection (McMahan and Streeter, 2010).[3]

For the analysis in Section 4, we will use this special case for quadratic regularization:

Corollary 6. *Consider Implicit-Update Composite-Objective Mirror Descent, which plays*

$$\hat{x}_{t+1} = \arg\min f_t(x) + \alpha_t \Psi(x) + \frac{1}{2}\left\|Q_{1:t}^{\frac{1}{2}}(x - \hat{x}_t)\right\|^2. \tag{14}$$

Then an equivalent FTPRL update is

$$x_{t+1} = \arg\min_x \quad (g'_{1:t-1} + \phi_{1:t-1}) \cdot x + f_t(x) + \alpha_t \Psi(x) + \frac{1}{2}\sum_{s=1}^{t}\left\|Q_s^{\frac{1}{2}}(x - x_s)\right\|^2 \tag{15}$$

for some $g'_t \in \partial f_t(x_{t+1})$ *and* $\phi_t \in \partial(\alpha_t\Psi)(x_{t+1})$.

Again let f_t^w be the loss functions provided by the world, and let f_t^u be the functions defining the update and used in the above corollary. Then, we encode implicit mirror descent by taking $f_t^u(x) \leftarrow f_t^w(x) + \alpha_t\Psi(x)$. We recover standard (non-implicit) COMID by taking $f_t^u(x) \leftarrow \nabla f_t^w(x_t) \cdot x + \alpha_t\Psi(x)$. Applying this result leads to the expression for COMID in Table 1.

Note that in both cases, the Ψ listed separately in Eq. (3) is taken to be zero; the Ψ specified in the problem only enters into the update through the f_t^u. That is, we don't actually need the machinery developed in this work for composite updates, rather we get an analysis of mirror-descent style composite updates via our analysis of *implicit* updates. The machinery for explicitly handling the full $\alpha_{1:t}\Psi$ penalty should be used in practice, however (see Section 2.2). Note also that the standard COMID algorithm can thus be viewed as a half-implicit algorithm: it uses an implicit update with respect to the Ψ term, but applies an immediate subgradient approximation to f_t^w.

We conclude the section with the proof of the main equivalence result.

Proof of Theorem 4 For simplicity we consider the case where f_t is differentiable.[4] By applying Theorem 3 to Eq. (13) (taking Φ to be all the terms other than the cumulative

[3] Zinkevich (2004, Sec. 5.2.3) describes a different lazy projection algorithm, which requires an appropriately chosen constant step-size to get low regret. FTRL-Proximal does not suffer from this problem, because it always centers the additional regularization R_t at points in $\mathcal{F}$, whereas our results show the algorithm of Zinkevich centers the additional regularization *outside* of $\mathcal{F}$, at the optimum of the unconstrained optimization. This leads to the high regret in the case of standard adaptive step sizes, because the algorithm can get "stuck" too far outside the feasible set to make it back to the other side.

[4] This ensures both g'_t and ϕ_t are uniquely determined; the proof still holds for general convex f_t, but only the sum $g'_t + \phi_t$ will be uniquely determined.

regularization), there exists a $\phi_t \in \partial(\alpha_t \Psi)(x_{t+1})$ such that $g'_t = \triangledown f_t(x_{t+1})$ and

$$g'_{1:t} + \phi_{1:t} + \triangledown \tilde{R}_{1:t}(x_{t+1}) = 0. \tag{16}$$

Similarly, applying Theorem 3 to Eq. (12) implies there exists a $\hat{\phi}_t \in \partial(\alpha_t \Psi)(\hat{x}_{t+1})$ such that $\hat{g}'_t = \triangledown f_t(\hat{x}_{t+1})$ and

$$\hat{g}'_t + \hat{\phi}_t + \triangledown \tilde{R}_{1:t}(\hat{x}_{t+1}) - \triangledown \tilde{R}_{1:t}(\hat{x}_t) = 0, \tag{17}$$

recalling that $\triangledown_u B_R(u, v) = \triangledown R(u) - \triangledown R(v)$.

We now proceed by induction on t, with the induction hypothesis that $x_t = \hat{x}_t$. The base case $t = 1$ follows from the assumption that $\hat{x}_1 = x_1 = 0$. Suppose the induction hypothesis holds for t. Taking Eq. (16) for $t - 1$ gives $g'_{1:t-1} + \phi_{1:t-1} + \triangledown \tilde{R}_{1:t-1}(x_t) = 0$, and since $\triangledown \tilde{R}_t(x_t) = 0$, we have

$$-\triangledown \tilde{R}_{1:t}(x_t) = g'_{1:t-1} + \phi_{1:t-1} \tag{18}$$

Beginning from Eq. (17),

$$\begin{aligned} \hat{g}'_t + \hat{\phi}_t + &\triangledown \tilde{R}_{1:t}(\hat{x}_{t+1}) - \triangledown \tilde{R}_{1:t}(\hat{x}_t) \\ &= \hat{g}'_t + \hat{\phi}_t + \triangledown \tilde{R}_{1:t}(\hat{x}_{t+1}) - \triangledown \tilde{R}_{1:t}(x_t) && \text{by the I.H.} \\ &= g'_{1:t-1} + \hat{g}'_t + \phi_{1:t-1} + \hat{\phi}_t + \triangledown \tilde{R}_{1:t}(\hat{x}_{t+1}), && (19) \end{aligned}$$

where the last line uses Eq. (18). The proof follows by applying Lemma 3 to Eqs. (16) and (19), and considering the pairs $(\hat{g}'_t + \hat{\phi}_t, \hat{x}_{t+1})$ and $(g'_t + \phi_t, x_{t+1})$. The equality $\phi_t = \hat{\phi}_t$ follows from the fact that $g'_t = \hat{g}'_t$ since f_t is differentiable. ■

3.2 An Equivalence Theorem for Origin-Centered Regularization

For the moment, suppose $\Psi(x) = 0$. So far, we have shown conditions under which gradient descent on $f_t(x) = g_t \cdot x$ with an adaptive step size is equivalent to follow-the-proximally-regularized-leader. In this section, we show that mirror descent on the *regularized* functions $f^R_t(x) = g_t \cdot x + R_t(x)$, with a certain natural step-size, is equivalent to a follow-the-regularized-leader algorithm with origin-centered regularization. For simplicity, in this section we restrict our attention to linear f_t (equivalently, non-implicit updates). The extension to implicit updates is straightforward.

The algorithm schema we consider next was introduced by Bartlett et al. (2007, Theorem 2.1). Letting $R_t(x) = \frac{\sigma_t}{2}\|x\|^2_2$ and fixing $\eta_t = \frac{1}{\sigma_{1:t}}$, their adaptive online gradient descent algorithm is

$$x_{t+1} = x_t - \eta_t \triangledown f^R_t(x_t) = x_t - \eta_t(g_t + \sigma_t x_t)).$$

We show (in Corollary 8) that this algorithm is identical to follow-the-leader on the functions $f^R_t(x) = g_t \cdot x + R_t(x)$, an algorithm that is minimax optimal in terms of regret against quadratic functions like f^R (Abernethy et al., 2008). As with the previous theorem, the difference between the two is how they handle an arbitrary Ψ. If one uses $\tilde{R}_t(x) = \frac{\sigma_t}{2}\|x - x_t\|^2_2$ in place of $R_t(x)$, this algorithm reduces to standard online gradient descent (Do et al., 2009).

The key observation of Bartlett et al. (2007) is that if the underlying functions f_t have strong convexity, we can roll that into the R_t functions, and so introduce less additional stabilizing regularization, leading to regret bounds that interpolate between $\sqrt{T}$ for linear functions and $\log T$ for strongly convex functions. Their work did not consider composite objectives (Ψ terms), but our equivalence theorems show their adaptivity techniques can be lifted to algorithms like RDA and FTRL-Proximal that handle such non-smooth functions more effectively than mirror descent formulations.

We will prove our equivalence theorem for a generalized versions of the algorithm. Instead of vanilla gradient descent, we analyze the mirror descent algorithm of Equation (11), but now g_t is replaced by $\triangledown f_t^R(x_t)$, and we add the composite term $\Psi(x)$.

Theorem 7. *Let $f_t(x) = g_t \cdot x$, and let $f_t^R(x) = g_t \cdot x + R_t(x)$, where R_t is a differentiable convex function. Let Ψ be an arbitrary convex function. Consider the composite-objective mirror-descent algorithm which plays*

$$\hat{x}_{t+1} = \arg\min_x \triangledown f_t^R(\hat{x}_t) \cdot x + \Psi(x) + \mathcal{B}_{1:t}(x, \hat{x}_t), \tag{20}$$

and the FTRL algorithm which plays

$$x_{t+1} = \arg\min_x f_{1:t}^R(x) + \phi_{1:t-1} \cdot x + \Psi(x), \tag{21}$$

for $\phi_t \in \partial\Psi(x_{t+1})$ such that $g_{1:t} + \triangledown R_{1:t}(x_{t+1}) + \phi_{1:t-1} + \phi_t = 0$. If both algorithms play $\hat{x}_1 = x_1 = 0$, then they are equivalent, in that $x_t = \hat{x}_t$ for all $t > 0$.

The most important corollary of this result is that it lets us add the adaptive online gradient descent algorithm to Table 1. It is also instructive to specialize to the simplest case when $\Psi(x) = 0$ and the regularization is quadratic:

Corollary 8. *Let $f_t(x) = g_t \cdot x$ and $f_t^R(x) = g_t \cdot x + \frac{\sigma_t}{2}\|x\|_2^2$. Then the following algorithms play identical points:*

- *FTRL, which plays $x_{t+1} = \arg\min_x f_{1:t}^R(x)$.*
- *Gradient descent on the functions f^R using the step size $\eta_t = \frac{1}{\sigma_{1:t}}$, which plays*

$$x_{t+1} = x_t - \eta_t \triangledown f_t^R(x_t)$$

- *Revisionist constant-step size gradient descent with $\eta_t = \frac{1}{\sigma_{1:t}}$, which plays*

$$x_{t+1} = -\eta_t g_{1:t}.$$

The last equivalence in the corollary follows from deriving the closed form for the point played by FTRL. We now proceed to the proof of the general theorem:

Proof of Theorem 7 The proof is by induction, using the induction hypothesis $\hat{x}_t = x_t$. The base case for $t = 1$ follows by inspection. Suppose the induction hypothesis holds for t; we will show it also holds for $t + 1$. Again let $r_t = \triangledown R_t$ and consider Equation (21). Since R_1 is assumed to be strongly convex, applying Theorem 3 gives us that x_t is the unique solution to $\triangledown f_{1:t-1}^R(x_t) + \phi_{1:t-1} = 0$ and so $g_{1:t-1} + r_{1:t-1}(x_t) + \phi_{1:t-1} = 0$. Then, by the induction hypothesis,

$$-r_{1:t-1}(\hat{x}_t) = g_{1:t-1} + \phi_{1:t-1}. \tag{22}$$

Now consider Equation (20). Since R_1 is strongly convex, $\mathcal{B}_{1:t}(x, \hat{x}_t)$ is strongly convex in its first argument, and so by Theorem 3 we have that $\hat{x}_{t+1}$ and some $\phi'_t \in \partial\Psi(\hat{x}_{t+1})$ are the unique solution to

$$\triangledown f_t^R(\hat{x}_t) + \phi'_t + r_{1:t}(\hat{x}_{t+1}) - r_{1:t}(\hat{x}_t) = 0,$$

since $\triangledown_p \mathcal{B}_R(p,q) = r(p) - r(q)$. Beginning from this equation,

$$\begin{aligned}
0 &= \triangledown f_t^R(\hat{x}_t) + \phi_t' + r_{1:t}(\hat{x}_{t+1}) - r_{1:t}(\hat{x}_t) \\
&= g_t + r_t(\hat{x}_t) + \phi_t' + r_{1:t}(\hat{x}_{t+1}) - r_{1:t}(\hat{x}_t) \\
&= g_t + r_{1:t}(\hat{x}_{t+1}) + \phi_t' - r_{1:t-1}(\hat{x}_t) \\
&= g_t + r_{1:t}(\hat{x}_{t+1}) + \phi_t' + g_{1:t-1} + \phi_{1:t-1} && \text{Eq (22)} \\
&= g_{1:t} + r_{1:t}(\hat{x}_{t+1}) + \phi_{1:t-1} + \phi_t'.
\end{aligned}$$

Applying Theorem 3 to Equation (21), (x_{t+1}, ϕ_t) are the unique pair such that

$$g_{1:t} + r_{1:t}(x_{t+1}) + \phi_{1:t-1} + \phi_t = 0$$

and $\phi_t \in \partial\Psi(x_{t+1})$, and so we conclude $\hat{x}_{t+1} = x_{t+1}$ and $\phi_t' = \phi_t$. ■

4 Regret Analysis

In this section, we prove the regret bounds of Theorem 2 and Corollary 1. Recall the general update we analyze is

$$x_{t+1} = \arg\min_x g'_{1:t-1} \cdot x + f_t(x) + \alpha_{1:t}\Psi(x) + R_{1:t}(x) \tag{3}$$

where $g_t' \in \partial f_t(x_{t+1})$. It will be useful to consider the equivalent (by Theorem 3) update

$$x_{t+1} = \arg\min_x g'_{1:t} \cdot x + \alpha_{1:t}\Psi(x) + R_{1:t}(x). \tag{23}$$

We can view this alternative update as running FTRL on the linear approximations of f_t taken at x_{t+1},

$$\bar{f}_t(x) = f_t(x_{t+1}) + g_t' \cdot (x - x_{t+1}).$$

To see the equivalence, note the constant terms in $\bar{f}$ change neither the argmin nor regret. This is still an implicit update, as implementing the update requires an oracle to compute an appropriate subgradient g_t' (say, by finding x_{t+1} via Equation (3)).

This re-interpretation is essential, as it lets us analyze a follow-the-leader algorithm on convex functions; note that the objective function of Equation (3) is not the sum of one convex function per round, as when moving from x_{t-1} to x_t we effectively add $g'_{t-1} \cdot x - f_{t-1}(x) + f_t(x)$ to the objective, which is not in general convex. By immediately applying an appropriate linearization of the loss functions, we avoid this non-convexity.

The affine functions $\bar{f}$ lower bound f_t, and so can be used to lower bound the loss of any $\mathring{x}$; however, in contrast to the more typical subgradient approximations taken at x_t, these linear functions are not tight at x_t, and so our analysis must also account for the additional loss $f_t(x_t) - \bar{f}_t(x_t)$. Before formalizing these arguments in the proof of Theorem 2, we prove the following lemma. We will use this lemma to get a tight bound on the regret of the algorithm against the linearized functions $\bar{f}$, but it is in fact much more general.

Lemma 9 (Strong 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. Define $f_t^R(x) =$*

$f_t(x) + R_t(x)$. *Then, if we play* $x_{t+1} = \arg\min_x f^R_{1:t}(x)$, *our regret against the functions* f_t *versus an arbitrary point* $\mathring{x}$ *is bounded by*

$$\textit{Regret} \le R_{1:T}(\mathring{x}) + \sum_{t=1}^{T} \Big(f^R_{1:t}(x_t) - f^R_{1:t}(x_{t+1}) - R_t(x_t)\Big).$$

A weaker (though sometimes easier to use) version of this lemma, stating

$$\text{Regret} \le R_{1:T}(\mathring{x}) + \sum_{t=1}^{T} \big(f_t(x_t) - f_t(x_{t+1})\big),$$

has been used previously (Kalai and Vempala, 2005, Hazan, 2008, McMahan and Streeter, 2010). In the case of linear functions with quadratic regularization, as in the analysis of McMahan and Streeter (2010), the weaker version loses a factor of $\frac{1}{2}$ (corresponding to a $\sqrt{2}$ in the final bound). The key is that in that case, being the leader is *strictly better* than playing the post-hoc optimal point. Quantifying this difference leads to the improved bounds for FTPRL in this paper.

Proof of Lemma 9 First, we consider regret against the functions f^R for not playing $\mathring{x}$:

$$\begin{aligned}
\text{Regret}(f^R) &= \sum_{t=1}^{T}(f^R_t(x_t) - f^R_t(\mathring{x})) && \text{by definition}\\
&= \sum_{t=1}^{T} f^R_t(x_t) - f^R_{1:T}(\mathring{x}) &&\\
&= \sum_{t=1}^{T}(f^R_{1:t}(x_t) - f^R_{1:t-1}(x_t)) - f^R_{1:T}(\mathring{x}) && \text{where } f_{1:0}(x) = 0\\
&\le \sum_{t=1}^{T}(f^R_{1:t}(x_t) - f^R_{1:t-1}(x_t)) - f^R_{1:T}(x_{T+1}) && \text{since } x_{T+1} \text{ minimizes } f^R_{1:T}\\
&= \sum_{t=1}^{T}(f^R_{1:t}(x_t) - f^R_{1:t}(x_{t+1})), &&
\end{aligned}$$

where the last line follows by simply re-indexing the $-f^R_{1:t}$ terms. Equivalently, applying the definitions of regret and f^R,

$$\sum_{t=1}^{T}(f_t(x_t) + R_t(x_t)) - f_{1:T}(\mathring{x}) - R_{1:T}(\mathring{x}) \le \sum_{t=1}^{T}(f^R_{1:t}(x_t) - f^R_{1:t}(x_{t+1})).$$

Re-arranging the inequality proves the theorem. ■

With this lemma in hand, we turn to our main proof. It is worth noting that the second half of the proof simplifies significantly when we choosing $x_t = y_t$, as in FTPRL.

Proof of Theorem 2 Recall $\bar{f}_t(x) = f_t(x_{t+1}) + g'_t \cdot (x - x_{t+1})$, a linear approximation of f_t taken at the next point, x_{t+1}. We can bound the regret of our algorithm (expressed as an FTRL algorithm on the functions $\bar{f}_t$, Equation (23)) against the functions $\bar{f}_t$ by applying

Lemma 9 to the functions $\bar{f}_t$ with regularization functions $R'_t(x) = R_t(x) + \alpha_t \Psi(x)$. Because we are taking the linear approximation at x_{t+1} instead of x_t, it may be the case that our actual loss $f_t(x_t)$ on round t is greater than the loss under $\bar{f}_t$, that is we may have $f_t(x_t) > \bar{f}_t(x_t)$. Thus, we must account for this additional regret. From the definition of regret we have

$$\begin{aligned}\text{Regret}(f) &= \text{Regret}(\bar{f}) + \sum_{t=1}^{T} (f_t(x_t) - \bar{f}_t(x_t)) + (\bar{f}_{1:t}(\mathring{x}) - f_{1:t}(\mathring{x})) \\ &\leq \text{Regret}(\bar{f}) + \sum_{t=1}^{T} (f_t(x_t) - \bar{f}_t(x_t))\end{aligned}$$

since $\bar{f}_t$ lower bounds f_t, and letting $\bar{f}^R_t(x) = \bar{f}_t(x) + R'_t(x)$,

$$\leq \underbrace{R'_{1:T}(\mathring{x}) + \sum_{t=1}^{T} (\bar{f}^R_{1:t}(x_t) - \bar{f}^R_{1:t}(x_{t+1}) - R'_t(x_t))}_{\text{Lemma 9 on } \bar{f}_t \text{ and } R'_t} \quad + \quad \underbrace{\sum_{t=1}^{T} (f_t(x_t) - \bar{f}_t(x_t))}_{\text{Underestimate of real loss at } x_t}.$$

Let Δ_t be the contribution of the non-regularization terms for a particular t,

$$\begin{aligned}\Delta_t &= \bar{f}^R_{1:t}(x_t) - \bar{f}^R_{1:t}(x_{t+1}) + f_t(x_t) - \bar{f}_t(x_t), \\ &= \bar{f}_{1:t}(x_t) + R'_{1:t}(x_t) - \bar{f}_{1:t}(x_{t+1}) - R'_{1:t}(x_{t+1}) + f_t(x_t) - \bar{f}_t(x_t), \\ &= \bar{f}_{1:t-1}(x_t) + R'_{1:t}(x_t) - \bar{f}_{1:t}(x_{t+1}) - R'_{1:t}(x_{t+1}) + f_t(x_t), \\ &= (\bar{f}_{1:t-1}(x_t) + R'_{1:t}(x_t) + f_t(x_t)) - (\bar{f}_{1:t}(x_{t+1}) + R'_{1:t}(x_{t+1})).\end{aligned}$$

For the terms containing x_{t+1}, using the fact that $\bar{f}_t(x_{t+1}) = f(x_{t+1})$, we have

$$\bar{f}_{1:t}(x_{t+1}) + R'_{1:t}(x_{t+1}) = \bar{f}_{1:t-1}(x_{t+1}) + R'_{1:t}(x_{t+1}) + f_t(x_{t+1}). \tag{24}$$

For a fixed t, we define two helper functions h_1 and h_2. Let

$$h_2(x) = \bar{f}_{1:t-1}(x) + R_{1:t}(x) + \alpha_{1:t}\Psi(x) + f_t(x),$$

so $\Delta_t = h_2(x_t) - h_2(x_{t+1})$. Define

$$h_1(x) = \bar{f}_{1:t-1}(x) + R_{1:t-1}(x) + \alpha_{1:t-1}\Psi(x).$$

Then we can write

$$h_2(x) = h_1(x) + f_t(x) + R_t(x) + \alpha_t \Psi(x).$$

By definition of our updates, $x_t = \arg\min_x h_1(x)$ (using Eq. (23)) and $x_{t+1} = \arg\min_x h_2(x)$.

Now, suppose we choose regularization $R_t(x) = \frac{1}{2}\|Q_t^{\frac{1}{2}}(x - y_t)\|^2$. The remainder of the proof is accomplished by bounding $h_2(x_t) - h_2(x_{t+1})$, with the aid of two lemmas (stated and proved below). First, by expanding h_1 and dropping constant terms (which cancel from Δ_t), we have

$$\begin{aligned}h_1(x) &= \frac{1}{2} x^\top Q_{1:t-1} x + \Big(g'_{1:t-1} - \frac{1}{2}\sum_{s=1}^{t-1} Q_s y_s\Big) \cdot x + \alpha_{1:t-1}\Psi(x) \\ &= \frac{1}{2}\big\|Q^{\frac{1}{2}}_{1:t-1}(x - x_t)\big\|^2 + \hat{\Psi}(x) + k'_t \qquad \text{Lemma 10}\end{aligned}$$

for some constant $k'_t \in \mathbb{R}$. Recall $Q^{\frac{1}{2}}_{1:t-1} = (Q_1 + \cdots + Q_{t-1})^{\frac{1}{2}}$. Now, we can apply Lemma 11. The constant k'_t cancels out, and we take $Q_a = Q_{1:t-1}$, $Q_b = Q_t$, $\Phi_a = \hat{\Psi}$, $\Phi_b = \alpha_t \Psi$, $x_1 = x_t$, etc. Thus, letting $d_t = y_t - x_t$,

$$\begin{aligned}\Delta_t &= h_1(x_t) - h_2(x_{t+1}) \\ &\leq (g_t - \frac{1}{2}\tilde{g}_t)^\top Q^{-1}_{1:t}\tilde{g}_t + \frac{1}{2}\left\| Q^{-\frac{1}{2}}_{1:t}(Q_t d_t)\right\|^2 - g Q^{-1}_{1:t} Q_t d_t + \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}). \qquad (25)\end{aligned}$$

We now re-incorporate the $-R'_t(x_t)$ terms not included in the definition of Δ_t. Note $R'_t(x) \geq R_t(x)$, and $R_t(x_t) = \frac{1}{2}\left\| Q^{\frac{1}{2}}_t d_t \right\|^2$. Then

$$\begin{aligned}\frac{1}{2}\left\| Q^{-\frac{1}{2}}_{1:t}(Q_t d_t)\right\|^2 - \frac{1}{2}\left\| Q^{\frac{1}{2}}_t d_t \right\|^2 &= \frac{1}{2} d_t^\top Q_t^\top Q^{-1}_{1:t} Q_t d_t - \frac{1}{2} d_t^\top Q_t d_t \\ &\leq \frac{1}{2} d_t^\top Q_t^\top Q_t^{-1} Q_t d_t - \frac{1}{2} d_t^\top Q_t d_t = 0\end{aligned}$$

where we have used the fact that $Q_{1:t} \succeq Q_t \succeq 0$ implies $Q_t^{-1} \succeq Q^{-1}_{1:t} \succeq 0$. Combining this result with Eq. (25) and adding back the $R_{1:t}(\mathring{x})$ term gives

$$\text{Regret} \leq R_{1:t}(\mathring{x}) + \sum_{t=1}^{T} \Big((g_t - \frac{1}{2}\tilde{g}_t)^\top Q^{-1}_{1:t}\tilde{g}_t - g_t Q^{-1}_{1:t} Q_t (y_t - x_t) + \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) \Big).$$

Defining $\alpha_{T+1} = 0$, observe

$$\begin{aligned}\sum_{t=1}^{T} \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) &= \sum_{t=1}^{T} \alpha_t \Psi(x_t) - \alpha_{t+1}\Psi(x_{t+1}) + (\alpha_{t+1} - \alpha_t)\Psi(x_{t+1}) \\ &\leq \sum_{t=1}^{T} \alpha_t \Psi(x_t) - \alpha_{t+1}\Psi(x_{t+1}) \\ &= \alpha_1 \Psi(x_1) - \alpha_{T+1}\Psi(x_{T+1}) = 0\end{aligned}$$

where the inequality uses the fact that $0 \leq \alpha_{t+1} \leq \alpha_t$ and $\Psi(x) \geq 0$. The last equality follows from $\Psi(x_1) = \Psi(0) = 0$ and $\alpha_{T+1} = 0$. Thus we conclude

$$\text{Regret} \leq R_{1:t}(\mathring{x}) + \sum_{t=1}^{T} (g_t - \frac{1}{2}\tilde{g}_t)^\top Q^{-1}_{1:t}\tilde{g}_t - g_t^\top Q^{-1}_{1:t} Q_t (y_t - x_t).$$

The second inequality in the theorem statement follows from Equation (28) of Lemma 11.
■

Only $R_{1:t}(\mathring{x})$ and the last term in the bound depend on the center of the regularization y_t; the final term can either increase or decrease regret, depending on the relationship between g_t and $y_t - x_t$ (note g_t is not known when y_t is selected). If we consider the simple case where all $Q_t = \sigma_t I$, observe that if $-g_t \cdot (y_t - x_t) > 0$ then (roughly speaking) both the new regularization penalty and the gradient of the loss function are pulling x_{t+1} away from x_t in the same direction, and so regret from this term will be larger.

Proof of Corollary 1 We first consider FTPRL. Let $Q_t = \sigma_t I$, and define σ_t such that $\sigma_{1:t} = G\sqrt{2t}/D$. Then taking Theorem 2 with $x_t = y_t$ gives

$$\begin{aligned}
\text{Regret} &\le \sum_{t=1}^{T} \frac{\sigma_t}{2} \left\| \mathring{x} - x_t \right\|^2 + \sum_{t=1}^{T} \frac{{g_t}^2}{2\sigma_{1:t}} \\
&\le \frac{\sigma_{1:T}}{2} D^2 + \sum_{t=1}^{T} \frac{G^2}{2\sigma_{1:t}} \\
&= \frac{GD\sqrt{2T}}{2} + \frac{GD}{2\sqrt{2}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \\
&\le DG\sqrt{2T},
\end{aligned}$$

where the last inequality uses the fact that $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \le 2\sqrt{T}$.

Recall the characterization of implicit-update mirror descent from Section 3. Thus, in this case we have $f^u_t(x) \leftarrow f^w_t(x) + I_{\mathcal{F}}(x)$. Let $g^w_t = \triangledown f^w_t(x_t)$, so in the analysis we have $g^u_t = g^w_t + \phi_t$. Following standard arguments, e.g. (Bartlett et al., 2007, Duchi et al., 2010b), it is straightforward to use the Pythagorean theorem for Bregman divergences to show $\frac{1}{2}\left\| g^w_t \right\|^2 \ge \frac{1}{2}\left\| g^u_t \right\|^2$, and then the result follows as for FTPRL.

For regularized dual averaging we have $y_t = 0$. Again let $Q_t = \sigma_t I$, and define σ_t such that $\sigma_{1:t} = 2G\sqrt{2t}/D$. Then, Theorem 2 gives

$$\text{Regret} \le \sum_{t=1}^{T} \frac{\sigma_t}{2} \left\| \mathring{x} \right\|^2 + \sum_{t=1}^{T} \frac{{g_t}^2}{2\sigma_{1:t}} - \frac{\sigma_t}{\sigma_{1:t}} g_t \cdot x_t.$$

The proof is largely similar to that for FTPRL, but we must deal with an extra term. First, note for $t \ge 2$,

$$\sigma_t = \sigma_{1:t} - \sigma_{1:t-1} = \frac{2G\sqrt{2}}{D}(\sqrt{t} - \sqrt{t-1}) \le \frac{2G\sqrt{2}}{D}\left(\frac{1}{2\sqrt{t-1}}\right) \le \frac{2G}{D\sqrt{t}},$$

where we have used $\sqrt{t} - \sqrt{t-1} \le \frac{1}{2\sqrt{t-1}}$ and for $t \ge 2$, $1/\sqrt{t-1} \le \sqrt{2}/\sqrt{t}$. Then, noting the term for $t = 1$ is zero since $x_1 = 0$,

$$\sum_{t=1}^{T} -\frac{\sigma_t}{\sigma_{1:t}} g_t \cdot x_t \le GD \sum_{t=2}^{T} \frac{\sigma_t}{\sigma_{1:t}} \le GD \sum_{t=2}^{T} \frac{D}{2G\sqrt{2t}} \frac{2G}{D\sqrt{t}} \le \frac{GD}{\sqrt{2}} \sum_{t=2}^{T} \frac{1}{t} \le \frac{GD}{\sqrt{2}}(\ln T + 1).$$

Applying this observation,

$$\begin{aligned}
\text{Regret} &\le \sum_{t=1}^{T} \frac{\sigma_t}{2} \left\| \mathring{x} \right\|^2 + \sum_{t=1}^{T} \frac{{g_t}^2}{2\sigma_{1:t}} - \frac{\sigma_t}{\sigma_{1:t}} g_t \cdot x_t \\
&\le \frac{\sigma_{1:T}}{2}\left(\frac{D}{2}\right)^2 + \sum_{t=1}^{T} \frac{G^2}{2\sigma_{1:t}} + \frac{GD}{\sqrt{2}} \ln T + \mathcal{O}(1) \\
&= \frac{GD\sqrt{2T}}{4} + \frac{GD}{2\sqrt{2}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} + \frac{GD}{\sqrt{2}} \ln T + \mathcal{O}(1) \\
&\le \frac{1}{2} DG\sqrt{2T} + \frac{GD}{\sqrt{2}} \ln T + \mathcal{O}(1).
\end{aligned}$$

■

We now prove the two lemmas used in bounding the $h_1(x_t) - h_2(x_{t+1})$ terms in the proof of Theorem 2.

Lemma 10. *Let Ψ be a convex function defined on $\mathbb{R}^n$, and let $Q \in S^n_{++}$. Define*

$$h(x) = \frac{1}{2}x^\top Qx + b \cdot x + \Psi(x),$$

and let $x^* = \arg\min_x h(x)$. Then, we can rewrite h as

$$h(x) = \frac{1}{2}\left\|Q^{\frac{1}{2}}(x - x^*)\right\|^2 + \hat{\Psi}(x) + k,$$

where $k \in \mathbb{R}$ and $\hat{\Psi}$ is convex with $0 \in \partial\hat{\Psi}(x^*)$.

Proof. Since $Q \in S^n_{++}$, the function $\frac{1}{2}x^\top Qx$ is strongly convex, and so using Theorem 3, h has a unique minimizer x^* and there exists a (unique) ϕ such that

$$Qx^* + b + \phi = 0 \tag{26}$$

with $\phi \in \partial\Psi(x^*)$. Define $\hat{\Psi}(x) = \Psi(x) - \phi \cdot x$, and note $0 \in \partial\hat{\Psi}(x^*)$. Then,

$$\begin{aligned}
h(x) =& \frac{1}{2}x^\top Qx + b \cdot x + \Psi(x) \\
=& \frac{1}{2}x^\top Qx + (b + \phi) \cdot x + \hat{\Psi}(x) && \text{Defn. } \hat{\Psi}(x) \\
=& \frac{1}{2}x^\top Qx - x^\top Qx^* + \hat{\Psi}(x) && \text{Eq. (26)} \\
=& \frac{1}{2}\left\|Q^{\frac{1}{2}}(x - x^*)\right\|^2 + \hat{\Psi}(x) - \frac{1}{2}\left\|Q^{\frac{1}{2}}x^*\right\|^2,
\end{aligned}$$

where $\hat{\Psi}$ and $k = -\frac{1}{2}\left\|Q^{\frac{1}{2}}x^*\right\|^2$ satisfy the requirements of the theorem. □

Lemma 11. *Let $x_1 \in \mathbb{R}^n$, let Φ_a be a convex function such that $0 \in \partial\Phi_a(x_1)$, and let $Q_a \in S^n_{++}$. Define*

$$h_1(x) = \frac{1}{2}\left\|Q_a^{\frac{1}{2}}(x - x_1)\right\|^2 + \Phi_a(x),$$

so $x_1 = \arg\min_x h_1(x)$. Let f and Φ_b be convex functions, let $Q_b \in S^n_{+}$, and define

$$h_2(x) = h_1(x) + f(x) + \frac{1}{2}\left\|Q_b^{\frac{1}{2}}(x - y)\right\|^2 + \Phi_b(x).$$

Let $x_2 = \arg\min_x h_2(x)$, let $g \in \partial f(x_1)$, let $d = y - x_1$, and let $Q_{a:b} = Q_a + Q_b$. Then, there exists a certain subgradient $\tilde{g}$ of f such that

$$h_2(x_1) - h_2(x_2) \le \left(g - \frac{1}{2}\tilde{g}\right)^\top Q_{a:b}^{-1}\tilde{g} + \frac{1}{2}\left\|Q_{a:b}^{-\frac{1}{2}}(Q_b d)\right\|^2 - g^\top Q_{a:b}^{-1}Q_b d + \Phi_b(x_1) - \Phi_b(x_2) \tag{27}$$

Further,

$$\left(g - \frac{1}{2}\tilde{g}\right)^\top Q_{a:b}^{-1}\tilde{g} \le \frac{1}{2}g^\top Q_{a:b}^{-1}g - \delta \tag{28}$$

where $\delta \ge 0$.

As we will see in the proof, $\delta > 0$ when the implicit update is non-trivial.

Proof. To obtain these bounds, we first analyze the problem without the Φ terms. For this purpose, we define

$$\tilde{h}_2(x) = \frac{1}{2}\big\|Q_a^{\frac{1}{2}}(x - x_1)\big\|^2 + \frac{1}{2}\big\|Q_b^{\frac{1}{2}}(x - y)\big\|^2 + f(x),$$

and let $\tilde{x}_2 = \arg\min_x \tilde{h}_2(x)$. We can re-write

$$\begin{aligned}\tilde{h}_2(x) &= f(x) + \frac{1}{2}\big\|Q_a^{\frac{1}{2}}(x - x_1)\big\|^2 + \frac{1}{2}\big\|Q_b^{\frac{1}{2}}(x - x_1 - d)\big\|^2 \\ &= f(x) + \frac{1}{2}\big\|Q_{a:b}^{\frac{1}{2}}(x - x_1)\big\|^2 - d^\top Q_b(x - x_1) + \frac{1}{2}\big\|Q_b^{\frac{1}{2}}d\big\|^2.\end{aligned}$$

Then, using Theorem 3 on the last expression, there exists a $\tilde{g} \in \partial f(\tilde{x}_2)$ such that $\tilde{g} + Q_{a:b}(\tilde{x}_2 - x_1) - Q_b d = 0$, and so in particular

$$\tilde{x}_2 - x_1 = Q_{a:b}^{-1}(Q_b d - \tilde{g}). \tag{29}$$

Then,

$$\begin{aligned}&\tilde{h}_2(x_1) - \tilde{h}_2(\tilde{x}_2) \\ &\qquad = f(x_1) + \frac{1}{2}\big\|Q_b^{\frac{1}{2}}d\big\|^2 - f(\tilde{x}_2) - \frac{1}{2}\big\|Q_{a:b}^{\frac{1}{2}}(\tilde{x}_2 - x_1)\big\|^2 + d^\top Q_b(\tilde{x}_2 - x_1) - \frac{1}{2}\big\|Q_b^{\frac{1}{2}}d\big\|^2 \\ &\qquad = f(x_1) - f(\tilde{x}_2) - \frac{1}{2}\big\|Q_{a:b}^{\frac{1}{2}}(\tilde{x}_2 - x_1)\big\|^2 + d^\top Q_b(\tilde{x}_2 - x_1),\end{aligned}$$

and since $f(\tilde{x}_2) \geq f(x_1) + g(\tilde{x}_2 - x_1)$ implies $f(x_1) - f(\tilde{x}_2) \leq -g(\tilde{x}_2 - x_1)$,

$$\leq -\frac{1}{2}\big\|Q_{a:b}^{\frac{1}{2}}(\tilde{x}_2 - x_1)\big\|^2 + (Q_b d - g)^\top(\tilde{x}_2 - x_1)$$

and applying Eq. (29),

$$\begin{aligned}&= -\frac{1}{2}\big\|Q_{a:b}^{-\frac{1}{2}}(Q_b d - \tilde{g})\big\|^2 + (Q_b d - g)^\top(Q_{a:b}^{-1}(Q_b d - \tilde{g})) \\ &= g^\top Q_{a:b}^{-1}\tilde{g} - \frac{1}{2}\big\|Q_{a:b}^{-\frac{1}{2}}\tilde{g}\big\|^2 + \frac{1}{2}\big\|Q_{a:b}^{-\frac{1}{2}}(Q_b d)\big\|^2 - g^\top Q_{a:b}^{-1}Q_b d,\end{aligned}$$

and so we conclude

$$\tilde{h}_2(x_1) - \tilde{h}_2(\tilde{x}_2) \leq \big(g - \frac{1}{2}\tilde{g}\big)^\top Q_{a:b}^{-1}\tilde{g} + \frac{1}{2}\big\|Q_{a:b}^{-\frac{1}{2}}(Q_b d)\big\|^2 - g^\top Q_{a:b}^{-1}Q_b d. \tag{30}$$

Next, we quantify the advantage offered by implicit updates. Suppose we choose $\tilde{x}_2$ by optimizing a version of $\tilde{h}_2$ where f is linearized at x_1:

$$\bar{h}_2(x) = \frac{1}{2}\big\|Q_a^{\frac{1}{2}}(x - x_1)\big\|^2 + \frac{1}{2}\big\|Q_b^{\frac{1}{2}}(x - y)\big\|^2 + g \cdot x.$$

Let $\bar{x}_2 = \arg\min_x \bar{h}_2(x)$. We say the implicit update is non-trivial when $\tilde{h}_2(\tilde{x}_2) < \tilde{h}_2(\bar{x}_2)$, that is, the implicit update provides a better solution to the optimization problem defined by $\tilde{h}_2$. By definition $\tilde{h}_2(\tilde{x}_2) \leq \tilde{h}_2(\bar{x}_2)$, and we can write $\tilde{h}_2(\tilde{x}_2) = \tilde{h}_2(\bar{x}_2) - 2\delta$ with $\delta \geq 0$. Let $R_1(x) = \frac{1}{2}\left\|Q_a^{\frac{1}{2}}(x - x_1)\right\|^2$ and $R_2(x) = \frac{1}{2}\left\|Q_b^{\frac{1}{2}}(x - y)\right\|^2$. Then, by the definition of $\tilde{x}_2$ and $\bar{x}_2$ we have

$$\begin{aligned}
R_{1:2}(\bar{x}_2) + g \cdot \bar{x}_2 &\leq R_{1:2}(\tilde{x}_2) + g \cdot \tilde{x}_2 \\
R_{1:2}(\tilde{x}_2) + \tilde{g} \cdot \tilde{x}_2 &= R_{1:2}(\bar{x}_2) + \tilde{g} \cdot \bar{x}_2 - 2\delta
\end{aligned}$$

and adding and canceling terms common to both sides gives

$$g \cdot \bar{x}_2 + \tilde{g} \cdot \tilde{x}_2 \leq g \cdot \tilde{x}_2 + \tilde{g} \cdot \bar{x}_2 - 2\delta. \tag{31}$$

Following Equation (29) $\tilde{x}_2 = Q_{a:b}^{-1}(Q_b d - \tilde{g}) + x_1$ or $\tilde{x}_2 = -Q_{a:b}^{-1}\tilde{g} + \kappa$ where $\kappa = Q_{a:b}^{-1}Q_b d + x_1$. Similarly, $\bar{x}_2 = -Q_{a:b}^{-1}g + \kappa$. Plugging into Equation (31), and noting the κ terms cancel, we have

$$-g^\top Q_{a:b}^{-1}g - \tilde{g}^\top Q_{a:b}^{-1}\tilde{g} \leq -g^\top Q_{a:b}^{-1}\tilde{g} - \tilde{g}^\top Q_{a:b}^{-1}g - 2\delta,$$

or re-arranging and dividing by one-half,

$$\frac{1}{2}g^\top Q_{a:b}^{-1}g - \delta \geq (g - \frac{1}{2}\tilde{g})^\top Q_{a:b}^{-1}\tilde{g}, \tag{32}$$

We now consider the functions that include the Φ terms. Note

$$h_2(x_2) = \tilde{h}_2(x_2) + \Phi_a(x_2) + \Phi_b(x_2) \geq \tilde{h}_2(\tilde{x}_2) + \Phi_a(x_1) + \Phi_b(x_2).$$

Then,

$$\begin{aligned}
h_2(x_1) - h_2(x_2) &= \tilde{h}_2(x_1) + \Phi_a(x_1) + \Phi_b(x_1) - h_2(x_2) \\
&\leq \tilde{h}_2(x_1) + \Phi_a(x_1) + \Phi_b(x_1) - \tilde{h}_2(\tilde{x}_2) - \Phi_a(x_1) - \Phi_b(x_2) \\
&= \tilde{h}_2(x_1) - \tilde{h}_2(\tilde{x}_2) + \Phi_b(x_1) - \Phi_b(x_2).
\end{aligned}$$

Combining this fact with Equations (30) and (32) proves the theorem. □

5 Experiments with L_1 Regularization

We compare FOBOS, FTRL-Proximal, and RDA on a variety of datasets to illustrate the key differences between the algorithms, from the point of view of introducing sparsity with L_1 regularization. In all experiments we optimize log-loss (see Section 2). Since our goal here is to show the impact of the different choices of regularization and the handling of the L_1 penalty, for simplicity we use first-order updates rather than implicit updates for the log-loss term.

For an experimental evaluation of implicit updates, we refer the reader to Karampatziakis and Langford (2010), which provides a convincing demonstration of the advantages of implicit updates on both importance weighted and standard learning problems.

Binary Classification We compare FTRL-Proximal, RDA, and FOBOS on several public datasets. We used four sentiment classification data sets (Books, Dvd, Electronics, and

Table 2: AUC (area under the ROC curve) for online predictions and sparsity in parentheses. The best value for each dataset is shown in bold. For these experiments, λ was fixed at $0.05/T$.

Data	FTRL-Proximal	RDA	FOBOS
books	0.874 (0.081)	**0.878** (**0.079**)	0.877 (0.382)
dvd	0.884 (0.078)	0.886 (**0.075**)	**0.887** (0.354)
electronics	0.916 (0.114)	**0.919** (**0.113**)	0.918 (0.399)
kitchen	0.931 (**0.129**)	**0.934** (0.130)	0.933 (0.414)
news	0.989 (**0.052**)	**0.991** (0.054)	0.990 (0.194)
rcv1	0.991 (**0.319**)	**0.991** (0.360)	0.991 (0.488)
web search ads	**0.832** (**0.615**)	0.831 (0.632)	0.832 (0.849)

Kitchen), available from (Dredze, 2010), each with 1000 positive examples and 1000 negative examples,[5] as well as the scaled versions of the rcv1.binary (20,242 examples) and news20.binary (19,996 examples) data sets from LIBSVM (Chang and Lin, 2010).

All our algorithms use a learning rate scaling parameter γ (see Section 2). The optimal choice of this parameter can vary somewhat from dataset to dataset, and for different settings of the L_1 regularization strength λ. For these experiments, we first selected the best γ for each (dataset, algorithm, λ) combination on a random shuffling of the dataset. We did this by training a model using each possible setting of γ from a reasonable grid (12 points in the range $[0.3, 1.9]$), and choosing the γ with the highest online AUC. We then fixed this value, and report the average AUC over 5 different shufflings of each dataset. We chose the area under the ROC curve (AUC) as our accuracy metric as we found it to be more stable and have less variance than the mistake fraction. However, results for classification accuracy were qualitatively very similar.

Ranking Search Ads by Click-Through-Rate We collected a dataset of about 1,000,000 search ad impressions from a large search engine,[6] corresponding to ads shown on a small set of search queries. We formed examples with a feature vector θ_t for each ad impression, using features based on the text of the ad and the query, as well as where on the page the ad showed. The target label y_t is 1 if the ad was clicked, and -1 otherwise.

Smaller learning-rates worked better on this dataset; for each (algorithm, λ) combination we chose the best γ from 9 points in the range $[0.03, 0.20]$. Rather than shuffling, we report results for a single pass over the data using the best γ, processing the events in the order the queries actually occurred. We also set a lower bound for the stabilizing terms $\bar{\sigma}_t$ of 20.0, (corresponding to a maximum learning rate of 0.05), as we found this improved accuracy somewhat. Again, qualitative results did not depend on this choice.

Results Table 2 reports AUC accuracy (larger numbers are better), followed by the density of the final predictor x_T (number of non-zeros divided by the total number of features present in the training data). We measured accuracy online, recording a prediction for each example before training on it, and then computing the AUC for this set of predictions. For these experiments, we fixed $\lambda = 0.05/T$ (where T is the number of examples in the dataset),

[5] We used the features provided in processed_acl.tar.gz, and scaled each vector of counts to unit length.

[6] While we report results on a single dataset, we repeated the experiments on two others, producing qualitatively the same results. No user-specific data was used in these experiments.

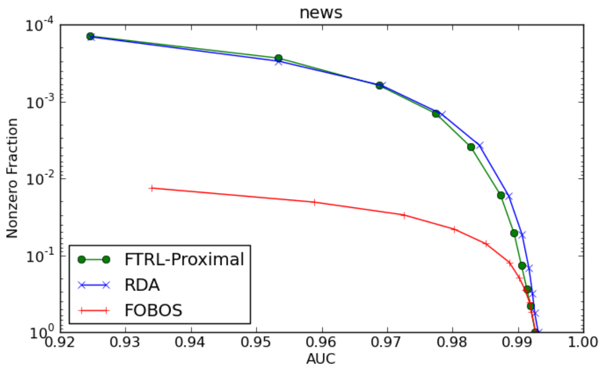

Figure 1: Sparsity versus accuracy tradeoffs on the 20 newsgroups dataset. Sparsity increases on the y-axis, and AUC increases on the x-axis, so the top right corner gets the best of both worlds. FOBOS is pareto-dominated by FTRL-Proximal and RDA.

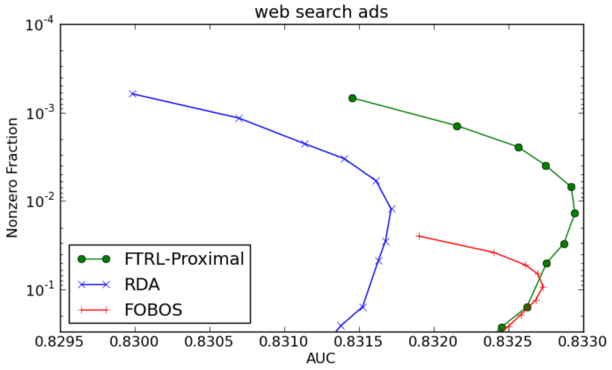

Figure 2: The same comparison as the previous figure, but on a large search ads ranking dataset. On this dataset, FTRL-Proximal significantly outperforms both other algorithms.

which was sufficient to introduce non-trivial sparsity. Overall, there is very little difference between the algorithms in terms of accuracy, with RDA having a slight edge for these choices for λ. Our main point concerns the sparsity numbers. It has been shown before that RDA outperforms FOBOS in terms of sparsity. The question then is how does FTRL-Proximal perform, as it is a hybrid of the two, selecting additional stabilization R_t in the manner of FOBOS, but handling the L_1 regularization in the manner of RDA. These results make it very clear: it is the treatment of L_1 regularization that makes the key difference for sparsity, as FTRL-Proximal behaves very comparably to RDA in this regard.

Fixing a particular value of λ, however, does not tell the whole story. For all these algorithms, one can trade off accuracy to get more sparsity by increasing the λ parameter. The best choice of this parameter depends on the application as well as the dataset. For example, if storing the model on an embedded device with expensive memory, sparsity might be relatively more important. To show how these algorithms allow different tradeoffs, we plot sparsity versus AUC for the different algorithms over a range of λ values. Figure 1 shows the tradeoffs for the 20 newsgroups dataset, and Figure 2 shows the tradeoffs for web search ads.

In all cases, FOBOS is pareto-dominated by RDA and FTRL-Proximal. These two algorithms are almost indistinguishable in the their tradeoff curves on the newsgroups dataset, but on the ads dataset FTRL-Proximal significantly outperforms RDA as well.[7]

6 Conclusions and Open Questions

The goal of this work has been to extend the theoretical understanding of several families of algorithms that have shown significant applied success for large-scale learning problems. We have shown that the most commonly used versions of mirror descent, FTRL-Proximal and RDA are closely related, and provided evidence that the non-smooth regularization Ψ is best handled globally, via RDA or FTRL-Proximal. Our analysis also extends these algorithms to implicit updates, which can offer significantly improved performance for some problems, including applications in active learning and importance-weighted learning.

Significant open questions remain. The observation that FOBOS is using a subgradient approximation for much of the cumulative L_1 penalty while RDA and FTRL-Proximal handle it exactly provides a compelling explanation for the improved sparsity produced by the latter two algorithms. Nevertheless, this is not a proof that these two algorithms always produce more sparsity. Quantitative bounds on sparsity have proved theoretically very challenging, and any additional results in this direction would be of great interest.

Similar challenges exist with quantifying the advantage offered by implicit updates. Our bounds demonstrate, essentially, a one-step advantage for implicit updates: on any given update, the implicit update will increase the regret bound by no more than the explicit linearized update, and the inequality will be strict whenever the implicit update is non-trivial. However, this is insufficient to say that for any given learning problem implicit updates will offer a better bound. After one update, the explicit and implicit algorithms will be at different feasible points x_{t+1}, which means that they will suffer different losses under f_{t+1} and (more importantly) compute and store different gradients for that function.

This issue is not unique to implicit updates: anytime the real loss functions f_t are non-linear, but the algorithm approximates them by computing $g_t = \triangledown f_t(x_t)$, two different first-order algorithms may see a different sequence of g_t's; since tight regret bounds depend on this sequence, the bounds will not be directly comparable. Generally we assume the gradients are bounded, $\|g_t\| \leq G$, which leads to bounds like $\mathcal{O}(G\sqrt{T})$, but since a large number of algorithms obtain this bound, it cannot be used to discriminate between them. Developing finer-grained techniques that can accurately compare the performance of different first-order online algorithms on non-linear functions could be of great practical interest to the learning community since the loss functions used are almost never linear.

Acknowledgments

The author wishes to thank Matt Streeter for numerous helpful discussions and comments, and Fernando Pereira for a conversation that helped focus this work on the choice $\Psi(x) =$

[7] The improvement is more significant than it first appears. A simple model with only features based on where the ads were shown achieves an AUC of nearly 0.80, and the inherent uncertainty in the clicks means that even predicting perfect probabilities would produce an AUC significantly less than 1.0, perhaps 0.85.

$\|x\|_1$.

References

Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *COLT*, 2008.

Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *NIPS*, 2007.

Alina Beygelzimer, Daniel Hsu, John Langford, and Zhang Tong. Agnostic active learning without constraints. In *NIPS*, 2010.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM data sets. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 2010.

Chuong B. Do, Quoc V. Le, and Chuan-Sheng Foo. Proximal regularization for online and batch learning. In *ICML*, 2009.

Mark Dredze. Multi-domain sentiment dataset (v2.0). http://www.cs.jhu.edu/~mdredze/datasets/sentiment/, 2010.

John Duchi and Yoram Singer. Efficient learning using forward-backward splitting. In *NIPS*. 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.

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

Sham M. Kakade, Shai Shalev-shwartz, and Ambuj Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. 2009.

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

Nikos Karampatziakis and John Langford. Importance weight aware gradient updates. `http://arxiv.org/abs/1011.1576`, 2010.

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

Jyrki Kivinen, Manfred Warmuth, and Babak Hassibi. The p-norm generalization of the lms algorithm for adaptive filtering. *IEEE Transactions on Signal Processing*, 54(5), 2006.

Brian Kulis and Peter Bartlett. Implicit online learning. In *ICML*, 2010.

Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient l1 regularized logistic regression. In *AAAI*, 2006.

H. Brendan McMahan. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization. Submitted, 2010.

H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, 2010.

Shai Shalev-Shwartz and Sham M. Kakade. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *NIPS*, pages 1457–1464, 2008.

Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and fenchel duality. In *NIPS*, 2006.

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

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4), 2008.

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *NIPS*, 2009.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11, 2010.

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

Martin Zinkevich. *Theoretical guarantees for algorithms in multi-agent settings*. PhD thesis, Pittsburgh, PA, USA, 2004.