Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Delay-Tolerant Online Convex Optimization: Unified Analysis and Adaptive-Gradient Algorithms

Pooria Joulani[1] **András György**[2] **Csaba Szepesvári**[1]
[1]Department of Computing Science, University of Alberta, Edmonton, AB, Canada
{pooria,szepesva}@ualberta.ca
[2]Department of Electrical and Electronic Engineering, Imperial College London, UK
a.gyorgy@imperial.ac.uk

Abstract

We present a unified, black-box-style method for developing and analyzing online convex optimization (OCO) algorithms for full-information online learning in delayed-feedback environments. Our new, simplified analysis enables us to substantially improve upon previous work and to solve a number of open problems from the literature. Specifically, we develop and analyze asynchronous AdaGrad-style algorithms from the Follow-the-Regularized-Leader (FTRL) and Mirror-Descent family that, unlike previous works, can handle projections and adapt both to the gradients and the delays, without relying on either strong convexity or smoothness of the objective function, or data sparsity. Our unified framework builds on a natural reduction from delayed-feedback to standard (non-delayed) online learning. This reduction, together with recent unification results for OCO algorithms, allows us to analyze the regret of generic FTRL and Mirror-Descent algorithms in the delayed-feedback setting in a unified manner using standard proof techniques. In addition, the reduction is exact and can be used to obtain both upper and lower bounds on the regret in the delayed-feedback setting.

1 Introduction

Online learning algorithms are at the heart of modern machine learning algorithms. The sequential nature of these algorithms makes them ideal for learning from data that is too large to be processed in a batch mode. However, their very sequential nature also predicts that they may be unfit to be used in parallel and distributed processing environments. To address this potential issue, several papers have studied asynchronous and distributed versions of online learning and stochastic optimization algorithms in recent years (see Section 3). These papers have shown that the ability to tolerate delays in receiving feedback is a key for obtaining asynchronous online learning algorithms.

Depending on the specifics of a machine learning task, a user can choose from several online learning algorithms (see, e.g., the book of Cesa-Bianchi and Lugosi, 2006). Previous work has typically focused on extending these algorithms to various delayed-feedback scenarios on a case-by-case basis, and usually under the stochastic optimization setting. However, many ideas and core challenges of delay-tolerant online learning are common across different domains and algorithms. In this paper, we take a different approach: we propose a unified theoretical framework for analyzing full-information online learning algorithms under delayed feedback. This unified approach enables us to simultaneously analyze various online convex optimization (OCO) methods (with linear losses or implicit updates) in the delayed-feedback setting, without the need to resort to ad-hoc analysis techniques.

The framework that we present is based on a natural reduction from the delayed-feedback online learning problem to standard, non-delayed online learning, as well as on recent unified analyses of OCO algorithms (McMahan 2014; Duchi, Hazan, and Singer 2011). In particular, our first main result gives an easy-to-interpret *identity* relating the regret of an algorithm operated in a delayed environment to the regret of the algorithm when operated in a non-delayed environment. All of our subsequent results are then derived from this general identity. We demonstrate the flexibility and power of our framework by solving several open problems from the literature. In particular, we analyze general delay-adaptive ADAGRAD-style algorithms, both with and without projection, without relying on either strong convexity or smoothness of the loss function, or data sparsity.

The rest of this paper is organized as follows. We start with the formal definition of the learning setting we consider (Section 2), followed by a summary of our results and their connection to the previous work (Section 3). We present our general reduction in Section 4 and the unified analysis of OCO algorithms in Section 5. Section 6 demonstrates the application of our framework in solving the aforementioned open problems. We conclude the paper in Section 7 and discuss some potential future work.

1.1 Notation and definitions

We will work with a closed, convex, non-empty subset $\mathcal{X}$ of a Hilbert space $\mathbb{X}$ over the reals. That is, $\mathbb{X}$ is a real vector space equipped with an inner product $\langle \cdot, \cdot \rangle$ that is complete with respect to (w.r.t.) the norm induced by $\langle \cdot, \cdot \rangle$. For example, we might have $\mathbb{X} = \mathbb{R}^d$ where $\langle \cdot, \cdot \rangle$ is the dot-product, or $\mathbb{X} = \mathbb{R}^{m \times n}$, the set of $m \times n$ real matrices, where $\langle A, B \rangle = \operatorname{tr}\left(A^\top B\right)$. Let $\mathcal{R} : S \to \mathbb{R}$ be a strictly convex, differentiable function over a convex closed domain $S \subset \mathbb{X}$ with a non-empty interior S°. Then, the $\mathcal{R}$-induced

Bregman divergence between the points $x \in S$, $y \in S^\circ$ is defined as $\mathcal{B}_\mathcal{R}(x,y) = \mathcal{R}(x) - \mathcal{R}(y) - \langle \nabla \mathcal{R}(y), x - y \rangle$. The function $\mathcal{R}$ is α-strongly convex with respect to a norm $\|\cdot\|$ on S if $\mathcal{B}_\mathcal{R}(x,y) \geq (\alpha/2)\|x-y\|^2$ for all $x \in S, y \in S^\circ$. The indicator of an event $\mathcal{E}$ is denoted by $\mathbb{I}\{\mathcal{E}\}$. For any sequence $c_i, c_{i+1}, \ldots, c_j$, we use $c_{i:j}$ to denote its sum, and we define $c_{i:j} = 0$ when $i > j$. For any function f, we denote the set of all sub-gradients of f at x by $\partial f(x)$, and use $f'(x)$ to denote any member of $\partial f(x)$.

2 Problem setting

We consider prediction under delayed feedback in an online convex optimization setting, building on the delayed-feedback online learning framework of Joulani, György, and Szepesvári (2013).[1] Let $\mathcal{F} \subset \{f : \mathcal{X} \to \mathbb{R}\}$ be a set of convex functions.The pair $(\mathcal{X}, \mathcal{F})$ defines the sequential prediction game shown in Figure 1. The game consists of a forecaster making predictions against a fixed (but unknown) sequence of loss function $f_1, f_2, \ldots, f_n \in \mathcal{F}$, possibly chosen in an adversarial manner before the game starts. In every round $t = 1, 2, \ldots, n$ of the game, the forecaster makes a prediction $x_t \in \mathcal{X}$ based on the feedback (specified below) that it has observed in rounds $1, \ldots, t-1$, and suffers the loss $f_t(x_t)$. The goal of the forecaster is to minimize its total loss compared to the loss of the best constant prediction $x^* \in \mathcal{X}$. More precisely, with the *regret* against an arbitrary prediction $x \in \mathcal{X}$ defined as

$$R_n(x) = \sum_{t=1}^{n} f_t(x_t) - f_t(x)\,,$$

the goal of the forecaster is to minimize its regret $R_n(x^*)$, where $x^* = \operatorname{argmin}_{x \in \mathcal{X}} \sum_{t=1}^{n} f_t(x)$ is the best prediction in hindsight.

The feedback based on which the forecaster can make prediction x_t is a subset of the loss functions from the previous rounds, $f_1, f_2, \ldots, f_{t-1}$. In particular, in a *non-delayed* setting, in each round $s = 1, \ldots, n$, the forecaster always observes f_s before the end of the round; thus, the forecaster can make the prediction x_t based on $f_1, f_2, \ldots, f_{t-1}$ (we will call an algorithm non-delayed if it is designed for this setting). On the other hand, in the *delayed-feedback* setting that we consider in this paper, the forecaster observes f_t only after a delay of (say) τ_t time steps, at the end of round $t + \tau_t$, where we assume that the delays $\tau_1, \tau_2, \ldots, \tau_n$ are fixed (but unknown) non-negative integers. Hence, after predicting x_t in time step t, the forecaster in a delayed-feedback setting observes $H_t = \{f_s : 1 \leq s \leq t, s + \tau_s = t\}$, the multi-set of loss functions from rounds $1, 2, \ldots, t$ that arrive at the end of time step t. As such, the prediction x_t can be based only on the observed loss functions $\cup_{s=1}^{t-1} H_s$, *i.e.*, based on the subset $\{f_s : 1 \leq s \leq t-1, s + \tau_s < t\} \subset \{f_1, f_2, \ldots, f_{t-1}\}$ of the loss functions from rounds $1, 2, \ldots, t-1$. Note that the non-delayed setting corresponds to the special case when $\tau_t = 0$ and $H_t = \{f_t\}$ for all $t = 1, 2, \ldots, n$. Finally, note that the delays can reorder the feedbacks, so that the feedback f_t of the interaction at time step $t < t'$ might arrive after feedback $f_{t'}$; this can happen when $t + \tau_t \geq t' + \tau_{t'}$. Note that the feedback does not include the index of the round the loss function corresponds to, *i.e.*, the feedback is not time-stamped.

The environment chooses a sequence of convex loss functions $f_1, \ldots, f_n \in \mathcal{F}$.
Repeat: for each time step $t = 1, 2, \ldots, n$:

1. The forecaster makes a prediction $x_t \in \mathcal{X}$.
2. The forecaster incurs loss $f_t(x_t)$ and receives the set of feedbacks $H_t = \{f_s : s + \tau_s = t\}$.

Goal: minimize $\sup_{x \in \mathcal{X}} R_n(x)$.

Figure 1: Delayed-feedback Online Convex Optimization

3 Contributions and related work

Our first contribution is providing a unified framework for analyzing the regret of OCO algorithms under delayed feedback. Our proof-technique has two main steps:
1- Black-box reduction: First, we show (Theorem 1) that when any deterministic non-delayed full-information online learning algorithm is used (without modification) in a delayed-feedback environment, the additional regret the algorithm suffers compared to running in a non-delayed environment depends on its "prediction drift", a quantity that roughly captures how fast the algorithm changes its predictions (Definition 1).
2- Unified bounds on the prediction drift: Next, we derive upper bounds (Propositions 1 and 2) on the prediction drift of two generic non-delayed OCO algorithms, which we call FTRL-PROX and ADA-MD, and obtain general delayed-feedback regret bounds (Theorem 4) for these algorithms by combining their drift bounds with the reduction result of Theorem 1. These two algorithms generalize, respectively, the Follow-The-Regularized-Leader and the Mirror-Descent classes of OCO algorithms and include various ADAGRAD-style algorithms as special cases.

Our second contribution is to develop, using the new framework mentioned above, FTRL- and Mirror-Descent-based ADAGRAD-style OCO methods (Section 6) that can adapt both to the observed gradients and the observed delays and can handle projection. These contributions solve a number of open problems in the literature, as follows:
Problem posed by McMahan and Streeter (2014): In a recent paper, McMahan and Streeter (2014) provide a delay-adaptive ADAGRAD-style extension of unconstrained single-coordinate Online Gradient Descent (OGD) for linear losses, through an indirect method called `AdaptiveRevision`. Their analysis is specific to OGD, relies crucially on the absence of projection, and assumes that the delays do not change the order the feedback is received (i.e., f_s is received before f_t for all $s < t$; the so-called `InOrder` assumption). The authors pose the question whether there exits a general analysis for algorithms of this type that is less indirect, avoids the `InOrder` assumption, and allows to analyze algorithms with projection and

[1] Note, however, that the reduction in Section 4 applies more generally to full-information online learning, not just OCO.

Dual Averaging methods. With the exception of Dual Averaging , the current paper solves this open problem, and obtains simpler algorithms even in the special case considered by McMahan and Streeter (2014).

Some problems posed by Mania et al. (2015): The recent paper by Mania et al. (2015) casts the effect of delays as noise on the gradients, and uses this "perturbed iterate" framework to analyze the convergence rate of the sequence of iterates generated by asynchronous unconstrained Stochastic Gradient Descent (SGD). Their analysis relies on strong convexity of the objective function. If the objective function is also smooth and the gradients are sparse and/or the delays satisfy specific bounds, they show that the effect of delays on the rate of convergence of these iterates is asymptotically negligible. The authors pose the question whether it is possible to obtain tight bounds for the function values (rather than the iterates), and whether their framework can be generalized beyond the strong convexity assumption to other learning settings, or to the analysis of AdaGrad-style algorithms. The current paper answers these questions: our framework applies to online convex optimization with linear losses or implicit updates and to function values rather than iterates, and our main reduction result provides an *identity*, not just an upper bound, for the delayed-feedback regret. Furthermore our framework applies to algorithms with projection, does not rely on strong convexity or smoothness of the loss functions,[2] and, as mentioned above, allows us to analyze delay-adaptive AdaGrad-style algorithms.

To our knowledge, Mesterharm (2007) was the first to observe, in a special, restricted adversarial classification setting, that the additional regret due to delays depends on how frequently an algorithm changes its predictions. The reduction we present can be considered as a refined and generalized version of his reduction (Mesterharm 2007, Chapter 8, Algorithm ODB-2). An advantage of this type of reduction to those in previous works (Weinberger and Ordentlich 2002; Joulani, György, and Szepesvári 2013) is its resource-efficiency: we use only one instance of a non-delayed online learning algorithm, while previous work created multiple instances, potentially wasting storage and computational resources.[3]

Several recent papers have studied delay-tolerant stochastic optimization (Recht et al. 2011; Agarwal and Duchi 2011; Nesterov 2012; Liu et al. 2013; Liu and Wright 2015); see also the references in the paper of Mania et al. (2015). These works typically show that for a specific non-delayed algorithm, for separable objective functions and under data sparsity (and usually assuming smoothness and strong convexity of the loss function), the effect of delays on the excess risk is asymptotically negligible, i.e., the rate of convergence is nearly the same as for the corresponding non-delayed algorithms, and hence linear speed ups are possible in parallel processing. We are instead interested in a more basic, unified analysis of the full-information online learning setting to uncover the exact regret penalty due to delays. In addition, assumptions such as data sparsity can be applied to our generic regret bounds to recover some of these results, without the need for smoothness or strong convexity.[4]

An interesting work is the recent paper of Sra et al. (2015), who consider adapting a 2-norm-regularized Mirror-Descent algorithm to the observed delays in the stochastic optimization setting with specific delay distributions. Compared to their work, we consider a more general version of Mirror-Descent and support FTRL algorithms as well, in the more general online learning setting. However, we would like to emphasize that currently our framework does not contain their work as a special case, since their algorithm does not maintain a non-decreasing regularizer.

Finally, the effect of delayed feedback has also been analyzed beyond the full-information model we consider here. For this, we refer the readers to Joulani, György, and Szepesvári (2013) and the references therein.

4 Single-instance black-box reduction

Consider any deterministic non-delayed online learning algorithm (call it Base). Suppose that we use Base, without modification, in a delayed-feedback environment: we feed Base with only the feedback that has arrived, and at each time step, we use the prediction that Base has made after receiving the most recent feedback. This scheme, which we call Solid (for "Single-instance Online Learning In Delayed environments"), is shown in Algorithm 1. In this section, we analyze the regret of Solid in the delayed-feedback setting.

Algorithm 1 Single-instance Online Learning In Delayed environments (Solid)

```
Set $x \leftarrow$ first prediction of Base.
for each time step $t = 1, 2, \ldots$ do
    Set $x_t \leftarrow x$ as the prediction for the current time step.
    Receive the set of feedbacks $H_t$ that arrive at the end of time step $t$.
    for each $f_s \in H_t$ do
        Update Base with $f_s$.
        $x \leftarrow$ the next prediction of Base.
    end for
end for
```

Solid reduces delayed-feedback online learning back to the standard (non-delayed) online learning problem. As we show below, we can express the regret of Solid under delayed feedback in terms of the regret of Base in a non-delayed setting and what we call the *prediction drift* of Base. We start with the definition of the latter.

Definition 1 (Prediction drift)**.** *Consider a non-delayed algorithm* Base *that is run with a sequence of loss functions* $f_1, f_2, \ldots, f_n$ *in a non-delayed setting, and let* $x_s, s =$

[2] Note, however, that our reduction is for full-information online learning, and to be applied with gradient-only information, we have to first linearize the losses. Hence, without full information, our regret bounds apply to the linearized loss, which might not give a regret bound as tight as the original smooth or strongly convex loss.

[3] Joulani, György, and Szepesvári (2013) also provide another reduction using only a single instance under stochastic feedback.

[4] See, e.g., the comparison with AsyncAdaGrad (Duchi, Jordan, and McMahan 2013) made by McMahan and Streeter (2014).

$1, 2, \dots, n$, *denote the* s*-th prediction of* BASE*. For every* $s = 1, 2, \dots, n$ *and* $\tau = 1, 2, \dots, s-1$*, the* prediction drift *of* BASE *on* f_s *from the previous* τ *time steps is defined as*

$$D_{s,\tau} = f_s(x_{s-\tau}) - f_s(x_s)\,,$$

the difference of the loss f_s *of predictions* $x_{s-\tau}$ *and* x_s*.*

Next, we introduce some further notation that is needed for our regret bound. For $1 \le s \le n$, let $\rho(s)$ denote the time step whose feedback $f_{\rho(s)}$ is the s-th feedback that SOLID gives to BASE, and let $\tilde{f}_s = f_{\rho(s)}$. Let $\tilde{x}_1$ be the first prediction of BASE and $\tilde{x}_{s+1}, s = 1, 2, \dots, n$, denote the prediction that BASE makes after receving the s-th feedback $\tilde{f}_s$. Note that not all predictions of BASE become predictions of SOLID. Also note that BASE makes predictions against the losses $\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_n$ sequentially without delays, that is, $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$ are the predictions of BASE in a non-delayed environment. For $t = 1, 2, \dots, n$, let $S(t) = \sum_{i=1}^{t-1} \mathbb{I}\{i + \tau_i < t\}$ denote the number of feedbacks that SOLID has observed (and has given to BASE) before making its t-th prediction x_t. Let $\tilde{\tau}_s = s-1-S(\rho(s))$ be the number of feedbacks that SOLID gives to BASE while the s-th feedback $\tilde{f}_s$ is outstanding, *i.e.*, the number of feedbacks that BASE receives between the time $\rho(s)$ when SOLID makes the prediction $x_{\rho(s)}$ and the time when the loss function $\tilde{f}_s = f_{\rho(s)}$ is given to BASE. For the analysis below, without loss of generality, we will assume that for any $1 \le t \le n$, $t + \tau_t \le n$, *i.e.*, all feedbacks are received by the end of round n. This does not restrict generality because the feedbacks that arrive in round n are not used to make any predictions and hence cannot influence the regret of SOLID. Note that under this assumption $\sum_{s=1}^{n} \tilde{\tau}_s = \sum_{t=1}^{n} \tau_t$ (both count over time the total number of outstanding feedbacks), and $(\rho(s))_{1 \le s \le n}$ is a permutation of the integers $\{1, \dots, n\}$.

Theorem 1. *Let* BASE *be any deterministic non-delayed forecaster. For every* $x \in \mathcal{X}$*, the regret of* SOLID *using* BASE *satisfies*

$$R_n(x) = \tilde{R}_n^{\text{BASE}}(x) + \sum_{s=1}^{n} \tilde{D}_{s,\tilde{\tau}_s}\,, \tag{1}$$

where $\tilde{R}_n^{\text{BASE}}(x) = \sum_{s=1}^{n} \tilde{f}_s(\tilde{x}_s) - \sum_{s=1}^{n} \tilde{f}_s(x)$ *is the (non-delayed) regret of* BASE *relative to any* $x \in \mathcal{X}$ *for the sequence of losses* $\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_n$*, and* $\tilde{D}_{s,\tilde{\tau}_s} = \tilde{f}_s(\tilde{x}_{s-\tilde{\tau}_s}) - \tilde{f}_s(\tilde{x}_s)$ *is the prediction drift of* BASE *while feedback* $\tilde{f}_s$ *is outstanding.*

Proof. By construction, for all time steps $t = 1, 2, \dots, n$, the prediction x_t of SOLID is the latest prediction of BASE, so we have $x_t = \tilde{x}_{S(t)+1}$, or, equivalently, $x_{\rho(s)} = \tilde{x}_{s-\tilde{\tau}_s}$ for $s = 1, 2, \dots, n$. Furthermore, by definition, $\tilde{f}_s = f_{\rho(s)}$, so we have $f_{\rho(s)}(x_{\rho(s)}) - \tilde{f}_s(\tilde{x}_s) = \tilde{f}_s(\tilde{x}_{s-\tilde{\tau}_s}) - \tilde{f}_s(\tilde{x}_s) = \tilde{D}_{s,\tilde{\tau}_s}$. Hence,

$$\begin{aligned}
R_n(x) &= \sum_{t=1}^{n} f_t(x_t) - \sum_{t=1}^{n} f_t(x) \\
&= \sum_{s=1}^{n} f_{\rho(s)}(x_{\rho(s)}) - \sum_{s=1}^{n} f_{\rho(s)}(x) \\
&= \sum_{s=1}^{n} \tilde{D}_{s,\tilde{\tau}_s} + \sum_{s=1}^{n} \tilde{f}_s(\tilde{x}_s) - \sum_{s=1}^{n} \tilde{f}_s(x) \\
&= \sum_{s=1}^{n} \tilde{D}_{s,\tilde{\tau}_s} + \tilde{R}_n^{\text{BASE}}(x). \qquad \square
\end{aligned}$$

Note that this result is an *identity*: upper and lower bounds on the (worst-case) regret and prediction drift of any algorithm BASE can be used to obtain upper and lower bounds on the delayed-feedback regret of SOLID.

Theorem 1 shows, in particular, that *stable* algorithms, *i.e.*, algorithms with small prediction drifts, are likely to suffer a small additional regret in delayed environments. While in general changing the predictions too slowly might reduce adaptivity and result in a larger regret, Theorem 1 shows that in delayed environments the extra regret might be worth the trade-off against the extra penalty from the prediction drift. This formalizes the intuition that in delayed environments one should reduce the learning rate of the algorithms, and helps us characterize the amount by which the learning rate should be decreased.

5 Stability of OCO algorithms

In this section, we prove upper-bounds on the prediction drift of a family of online convex optimization algorithms including Follow-The-Regularized-Leader (FTRL), Mirror Descent, and their adaptive variants such as the ADA-GRAD-style algorithms of McMahan and Streeter (2010) and Duchi, Hazan, and Singer (2011), both with and without projection. In particular, we study FTRL-PROX and ADA-MD,[5] which are defined as follows. Both of these algorithms use a sequence of "regularizer functions" $r_0, \dots, r_n : S \to \mathbb{R}$, chosen by the algorithm sequentially (possibly based on previous observations). We assume that $S \subset \mathbb{X}$ is convex and $\mathcal{X} \subset S^\circ$.[6] The first prediction of both algorithms is

$$x_1 = \operatorname{argmin}_{x \in \mathcal{X}}\ r_0(x). \tag{2}$$

Then, for $s > 1$, FTRL-PROX predicts

$$x_s = \operatorname{argmin}_{x \in \mathcal{X}}\ f_{1:s-1}(x) + r_{0:s-1}(x), \tag{3}$$

while the predictions of ADA-MD are given by

$$x_s = \operatorname{argmin}_{x \in \mathcal{X}}\ f_{s-1}(x) + \mathcal{B}_{r_{0:s-1}}(x, x_{s-1}), \tag{4}$$

where $\mathcal{B}_{r_{0:s-1}}(.,.)$ is the Bregman-divergence induced by $r_{0:s-1}$. For FTRL-PROX, we assume that the regularizers r_s are selected such that x_s minimizes r_s on $\mathcal{X}$.

Note that these algorithms have been studied previously in the literature, e.g., by McMahan and Streeter (2010), McMahan (2014), and Duchi, Hazan, and Singer (2011). To put our

[5] The nomenclature here is somewhat inconsistent. Here we use ADA-MD to mean adaptive mirror descent with implicit update (Kulis and Bartlett 2010) which contains the normal, linear mirror descent as a special case.

[6] By making more assumptions on r_i, *i.e.*, assuming that they are Legendre functions (see, e.g., Cesa-Bianchi and Lugosi 2006), this assumption on the domain of r_i could be relaxed.

analysis into context, first we state the existing non-delayed regret bounds for these algorithms. In what follows, $f'_s(x_s)$ denotes any sub-gradient of f_s at x_s.

Assumption 1. *The loss functions $f_s, s = 1, 2, \dots, n$, are convex, and for all $s = 0, 1, 2, \dots, n$, the regularizer r_s is convex and non-negative. Furthermore, for* FTRL-Prox *we assume that x_s minimizes r_s on $\mathcal{X}$.*

For FTRL-Prox, we have the following regret bound in the non-delayed setting (Theorem 1 of McMahan 2014).

Theorem 2 (Regret of FTRL-Prox)**.** *Suppose that Assumption 1 holds and that $f_{1:s} + r_{0:s}$ is 1-strongly convex on $\mathcal{X}$ w.r.t. some norm $\|.\|_{(s)}$ for all $s = 0, \dots, n$. Then the regret of* FTRL-Prox *is upper-bounded as*

$$R_n^{\text{FTRL-Prox}}(x^*) \leq r_{0:n}(x^*) + \frac{1}{2}\sum_{s=1}^{n} \|f'_s(x_s)\|^2_{(s),*}, \quad (5)$$

where $\|.\|_{(s),*}$ is the dual norm of $\|.\|_{(s)}$.

We also have the next regret bound for Ada-MD (following Proposition 3 of Duchi, Hazan, and Singer 2011):

Theorem 3 (Regret of Ada-MD)**.** *Suppose that Assumption 1 holds, and for all $s = 0, \dots, n$, $r_{0:s}$ is differentiable and 1-strongly convex on $\mathcal{X}$ w.r.t. some norm $\|.\|_{(s)}$. Then, the regret of* Ada-MD *is upper-bounded as*

$$R_n^{\text{Ada-MD}}(x^*) \leq \sum_{s=1}^{n} \mathcal{B}_{r_s}(x^*, x_s) + \frac{1}{2}\sum_{s=1}^{n} \|f'_s(x_s)\|^2_{(s),*} . \quad (6)$$

The next propositions bound the prediction drifts of FTRL-Prox and Ada-MD. The proofs are short and use standard FTRL and Mirror-Descent techniques. Note that since f_s is convex for $s = 1, 2, \dots, n$, for any sequence of norms $\|.\|_{(j)}, j = 1, 2, \dots, n$, and any $1 \leq \tau < s \leq n$,

$$\begin{aligned} D_{s,\tau} &= \sum_{j=s-\tau}^{s-1} f_s(x_j) - f_s(x_{j+1}) \\ &\leq \sum_{j=s-\tau}^{s-1} \langle f'_s(x_j), x_j - x_{j+1} \rangle \\ &\leq \sum_{j=s-\tau}^{s-1} \|f'_s(x_j)\|_{(j),*} \, \|x_j - x_{j+1}\|_{(j)} \, , \end{aligned} \quad (7)$$

where the last step follows by Hölder's inequality. We will use this inequality in our proofs below.

Proposition 1 (Prediction drift of FTRL-Prox)**.** *Under the conditions of Theorem 2, for every $1 \leq \tau < s \leq n$,*

$$D_{s,\tau} \leq \sum_{j=s-\tau}^{s-1} \|f'_s(x_j)\|_{(j),*} \|f'_j(x_j)\|_{(j),*} . \quad (8)$$

Proof. Starting from (7), it remains to bound $\|x_j - x_{j+1}\|_{(j)}$. Define $h_0 = r_0$ and $h_s = f_s + r_s$ for $s = 1, 2, \dots, n$. Then, by our assumptions, x_s minimizes $h_{0:s-1}$ over $\mathcal{X}$, and $h_{0:j}$ is 1-strongly convex w.r.t. $\|.\|_{(j)}$. Note that since x_j minimizes r_j over $\mathcal{X}$, it also minimizes $\phi_1 = h_{0:j-1} + r_j$. Then, since x_{j+1} minimizes $h_{0:j} = h_{0:j-1} + r_j + f_j$, Lemma **??** (see the extend version (Joulani, György, and Szepesvári 2015)) with ϕ_1 above and $\delta = f_j$ gives

$$\|x_j - x_{j+1}\|_{(j)} \leq \|f'_j(x_j)\|_{(j),*}. \qquad \square$$

Proposition 2 (Prediction drift of Ada-MD)**.** *Under the conditions of Theorem 3, for every $1 \leq \tau < s \leq n$,*

$$D_{s,\tau} \leq \sum_{j=s-\tau}^{s-1} \|f'_s(x_j)\|_{(j),*} \|f'_j(x_{j+1})\|_{(j),*} . \quad (9)$$

Proof. As above, we start from (7) and bound $\|x_j - x_{j+1}\|_{(j)}$. Recall that $r_{0:j}$ is differentiable by assumption. By the strong convexity of $r_{0:j}$,

$$\begin{aligned} \|x_j - x_{j+1}\|^2_{(j)} &\leq \mathcal{B}_{r_{0:j}}(x_{j+1}, x_j) + \mathcal{B}_{r_{0:j}}(x_j, x_{j+1}) \\ &= \langle r'_{0:j}(x_{j+1}) - r'_{0:j}(x_j), x_{j+1} - x_j \rangle. \end{aligned}$$

Also, by the first-order optimality condition on x_{j+1},

$$\langle f'_j(x_{j+1}) + r'_{0:j}(x_{j+1}) - r'_{0:j}(x_j), x_j - x_{j+1} \rangle \geq 0.$$

Combining the above,

$$\begin{aligned} \|x_j - x_{j+1}\|^2_{(j)} &\leq \langle f'_j(x_{j+1}), x_j - x_{j+1} \rangle \\ &\leq \|f'_j(x_{j+1})\|_{(j),*} \|x_j - x_{j+1}\|_{(j)}. \end{aligned}$$

The proposition follows by the non-negativity of norms. $\square$

Note that the proofs use only the standard FTRL-Prox and Ada-MD analysis techniques. Combining the above bounds with Theorem 1, it is straightforward to obtain regret guarantees for FTRL-Prox and Ada-MD in delayed-feedback environments. Consider an algorithm Base which is used inside Solid in a delayed-feedback game. Recall that Base receives the sequence of loss functions $\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_n$ and makes predictions $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$. Also recall that $\tilde{\tau}_n$ denoted the update delay, *i.e.*, the number of updates that Base performs from the time Solid selects x_t in time step t until the time when Base receives the corresponding loss function f_t.

Theorem 4. *Suppose Assumption 1 holds, and we run* Solid *in a delayed-feedback environment. Let $\tilde{r}_s, s = 0, 1, \dots, n$, denote the regularizers that* Base *uses in its simulated non-delayed run inside* Solid*, and let $\|.\|_{(s)}$ denote the associated strong-convexity norms. Let R_n denote the regret of* Solid *in its delayed-feedback environment.*

(i) If Base *is an* FTRL-Prox *algorithm and the conditions of Theorem 2 hold, then*

$$R_n \leq \tilde{r}_{0:n}(x^*) + \frac{1}{2}\sum_{s=1}^{n} \|\tilde{f}'_s(\tilde{x}_s)\|^2_{(s),*} + \sum_{s=1}^{n} \sum_{j=s-\tilde{\tau}_s}^{s-1} \|\tilde{f}'_s(\tilde{x}_j)\|_{(j),*} \|\tilde{f}'_j(\tilde{x}_j)\|_{(j),*} .$$

(ii) If Base *is an* Ada-MD *algorithm and the conditions in Theorem 3 hold, then*

$$R_n \leq \sum_{s=1}^{n} \mathcal{B}_{\tilde{r}_s}(x^*, \tilde{x}_s) + \frac{1}{2}\sum_{s=1}^{n} \|\tilde{f}'_s(\tilde{x}_s)\|^2_{(s),*}$$

$$\sum_{s=1}^{n}\sum_{j=s-\tilde{\tau}_s}^{s-1}\|\tilde{f}_s'(\tilde{x}_j)\|_{(j),*}\|\tilde{f}_j'(\tilde{x}_{j+1})\|_{(j),*}.$$

These bounds are still somewhat unwieldy. To get a more indicative result, suppose that there exists a norm $\|.\|$ such that for all $s = 1, 2, \dots, n$, we have $\|.\|_{(s)} = \frac{1}{\sqrt{\eta_s}}\|.\|$ for some non-negative constant η_s (e.g., consider a single-coordinate ADAGRAD algorithm). Note that by the non-negativity of Bregman divergences, this condition implies that the sequence (η_s) is non-increasing. Further suppose that there exists a constant G such that $\|f_s'(x)\| \le G$ for all $s = 1, 2, \dots, n$ and $x \in \mathcal{X}$, and a constant R such that $\eta_n \tilde{r}_{0:n}(x^*) \le 2R^2$ for FTRL-PROX or $\tilde{\eta}_n \sum_{s=1}^{n} \mathcal{B}_{\tilde{r}_s}(x^*, x_s) \le 2\mathbb{R}^2$ for ADA-MD. Let $\tau^* = \max_{1\le s\le n}\tilde{\tau}_s$ be the maximum delay. Theorems 2 and 3 give

$$R_n \le \frac{2R^2}{\eta_n} + \frac{G^2}{2}\sum_{s=1}^{n}\eta_s,$$

for FTRL-PROX and ADA-MD in the non-delayed setting, whereas Theorem 4 gives

$$R_n \le \frac{2R^2}{\tilde{\eta}_n} + \frac{G^2}{2}\sum_{s=1}^{n}\tilde{\eta}_s(1 + 2\tilde{\tau}_s),$$

for SOLID in the delayed-feedback setting, where $\tilde{\eta}_s, s = 1, 2, \dots, n$, denote the learning rates used by BASE inside SOLID. Using a constant learning rate η_s set as a function of the cumulative delay $\mathcal{T} = \sum_{s=1}^{n}\tilde{\tau}_s = \sum_{s=1}^{n}\tau_s$ (if available in advance), the regret becomes $O(\sqrt{T + 2\mathcal{T}})$, while scaling down the non-delayed learning rates η_s by $\sqrt{1 + 2\tau^*}$, we get a multiplicative regret penalty of $\sqrt{1 + 2\tau^*}$ compared to the non-delayed case.

While the above regret penalty matches (up to a constant factor) the worst-case lower bound for delayed environments (Weinberger and Ordentlich 2002; Mesterharm 2007), McMahan and Streeter (2014) show that one can tune the learning rates to adapt to the actual observed delays and past gradients. In the next section, we considerably generalize their results and solve the open problems they have posed.

6 Adaptive learning-rate tuning for linear loss functions

In this section we restrict our attention to linear loss functions $f_t = \langle g_t, \cdot\rangle$. Consider the setting of Theorem 4, and let $\tilde{g}_s$ denote the gradient of $\tilde{f}_s$. Consider a norm $\|.\|$, and for $s = 1, 2, \dots, n$, define $\hat{g}_s = \|\tilde{g}_s\|_*$. The following result is a corollary of Theorem 4 that generalizes the bound in Lemma 1 of McMahan and Streeter (2014) to FTRL-PROX and ADA-MD, either with or without projection. The proof is given in the extend version (Joulani, György, and Szepesvári 2015).

Corollary 1. *Consider the case of linear losses and suppose that the conditions of Theorem 4 hold. Suppose that $\tilde{r}_{0:s}, s = 0, 1, \dots, n$, is $(1/\tilde{\eta}_s)$-strongly convex w.r.t. the norm $\|.\|$. Then, the regret of* SOLID *satisfies*

$$R_n \le \frac{2R^2}{\tilde{\eta}_n} + \frac{1}{2}\sum_{j=1}^{n}\tilde{\eta}_j\hat{G}_j^{\text{fwd}}, \tag{10}$$

where for $j = 1, 2, \dots, n$,

$$\hat{G}_j^{\text{fwd}} = \hat{g}_j^2 + 2\hat{g}_j\sum_{s=j+1}^{n}\hat{g}_s\mathbb{I}\{s - \tilde{\tau}_s \le j\},$$

and $R > 0$ is such that $\tilde{\eta}_n\tilde{r}_{0:n}(x^*) \le 2R^2$ for FTRL-PROX, *or $\tilde{\eta}_n\sum_{t=1}^{n}\mathcal{B}_{r_t}(x^*, x_t) \le 2R^2$ for* ADA-MD.

Note that in a non-delayed setting, the regret bound of these adaptive algorithms is of the form

$$R_n \le \frac{2R^2}{\eta_n} + \frac{1}{2}\sum_{s=1}^{n}\eta_s\|g_s\|_*^2.$$

In such a case we would let $\eta_s = O(1/\sqrt{\sum_{j=1}^{s}\hat{g}_j^2})$ and then use Lemma **??** (see the extended version (Joulani, György, and Szepesvári 2015)) to get a regret bound of the form

$$R_n \le 2\sqrt{2}R^2\sqrt{\sum_{s=1}^{n}\|g_s\|_*^2}\,.$$

Similarly, with $\tilde{\eta}_s = O(1/\sqrt{\hat{G}_{1:s}^{\text{fwd}}})$, in a delayed-feedback setting we would achieve a regret of the form

$$R_n \le 2\sqrt{2}R^2\sqrt{\hat{G}_{1:n}^{\text{fwd}}}\,. \tag{11}$$

Unfortunately, this is not possible since $\hat{G}_s^{\text{fwd}}$ depends on future, unobserved gradients, and hence the $\tilde{\eta}_s$ given above cannot be computed at time step s.

To work around this problem, McMahan and Streeter (2014) define a quantity $\hat{G}_s^{\text{bck}}$ that depends only on the observed gradients. The goal is to bound $\hat{G}_{1:s}^{\text{fwd}}$ from above and below by a function of $\hat{G}_j^{\text{bck}}, j = 1, 2, \dots, s-1$, plus an additive term independent of s; then, setting the learning rate based on that quantity results in a regret bound that is only an additive term (independent of n) larger than the bound of (11). Similarly, in our setting we define

$$\hat{G}_s^{\text{bck}} = \hat{g}_s^2 + 2\hat{g}_s\sum_{j=s-\tilde{\tau}_s}^{s-1}\hat{g}_j.$$

The next lemma bounds $\hat{G}_{1:s}^{\text{fwd}}$ from above and below using $\hat{G}_{1:s}^{\text{bck}}$.

Lemma 1. *Let $G_* = \max_{1\le j\le n}\hat{g}_j$ and $\tau_* = \max_{1\le s\le n}\tilde{\tau}_s$. For all $t = 1, 2, \dots, n$,*

$$\hat{G}_{1:t}^{\text{bck}} \le \hat{G}_{1:t}^{\text{fwd}} \le \hat{G}_{1:t}^{\text{bck}} + (\tau_*^2 + \tau_*)G_*^2. \tag{12}$$

In addition,

$$\hat{G}_{1:n}^{\text{fwd}} = \hat{G}_{1:n}^{\text{bck}}. \tag{13}$$

Instead of using $\hat{G}_{1:s}^{\text{bck}}$ directly as in our Lemma 1, McMahan and Streeter (2014) use $\hat{G}_{1:o(s)}^{\text{bck}}$ for their bounds, where $o(s)$ is the index of the largest outstanding gradient at the time of update s. Their bounds need an extra `In-Order` assumption on the delays, i.e, that the delays do not change the order of the updates. In addition, $\hat{G}^{\mathrm{bck}}_{1:o(s)}$ is not efficiently computable in an online fashion (it requires keeping track of the outstanding updates), and they use an indirect algorithm (called `AdaptiveRevision`) on top of this learning rate schedule that "revises" the previous gradients and can be implemented in practice. We do not require this indirect approach, since $\hat{G}^{\mathrm{bck}}_{s}$ can be efficiently computed in an online fashion (in fact, this is the quantity z that the `AdaptiveRevision` algorithm of McMahan and Streeter (2014) also needs and maintains, using the network as storage).

Based on Lemma 1, we can show that setting the learning rate, for some $\alpha > 0$, as

$$\tilde{\eta}_j = \alpha \left(\sqrt{\hat{G}^{\mathrm{bck}}_{1:j} + (\tau_*^2 + \tau_*) G_*^2} \right)^{-1} , \qquad (14)$$

results in only a constant additional regret compared to using the learning rate $\tilde{\eta}_j = O\left(1/\sqrt{\hat{G}^{\mathrm{fwd}}_{1:j}}\right)$. We prove this in the following theorem.

Theorem 5. *Consider the conditions of Corollary 1. If* $\alpha = \sqrt{2}R$ *and* $\tilde{\eta}_t$ *is given by* (14)*, then the regret of* SOLID *with* FTRL-PROX *or* ADA-MD *can be bounded as*

$$R_n(x^*) \le 2\sqrt{2}R\sqrt{\hat{G}^{\mathrm{fwd}}_{1:n}} + \sqrt{2(\tau_*^2 + \tau_*)}RG_* \,.$$

This generalizes the bound

$$R_n(x^*) \le 2\sqrt{2}R\sqrt{\max_{1 \le s \le n} \hat{G}^{\mathrm{fwd}}_{1:s}} + O(\tau^* R G_*) \qquad (15)$$

obtained in Theorem 3 of McMahan (2014) for `AdaptiveRevision`. Note, however, that in the case of `AdaptiveRevision`, the algorithm is applied to a single coordinate, and the $\hat{g}$ values are the actual (possibly negative) gradients, not their norms. To refine our bound to the one-dimensional setting, one can define the step-size $\tilde{\eta}_j$ based on the maximum $\hat{G}^{\mathrm{bck}}_{1:i}$ for $1 \le i \le j$ (this is still efficiently computable at time j, and corresponds to the quantity z' in `AdaptiveRevision`). Then, a modified Lemma 1 together with Corollary 10 of McMahan and Streeter (2014) gives a regret bound similar to (15).

7 Conclusion and future work

We provided a unified framework for developing and analyzing online convex optimization algorithms under delayed feedback. Based on a general reduction, we extended two generic adaptive online learning algorithms (an adaptive FTRL and an adaptive Mirror-Descent algorithm) to the delayed feedback setting. Our analysis resulted in generalized delay-tolerant ADAGRAD-style algorithms that adapt both to the gradients and the delays, solving a number of open problems posed by McMahan and Streeter (2014) and Mania et al. (2015).

An interesting problem for future research is analyzing delay-tolerant adaptive Dual Averaging algorithms using this framework. Deriving lower bounds for asynchronous optimization using Theorem 1 is also of natural interest. Finally, it seems to be possible to extend our framework to use gradients or higher-order information only, using a shifting argument; this is also left for future work.

Acknowledgments This work was supported by the Alberta Innovates Technology Futures through the Alberta Ingenuity Centre for Machine Learning and NSERC.

References

Agarwal, A., and Duchi, J. 2011. Distributed delayed stochastic optimization. In Shawe-Taylor, J.; Zemel, R.; Bartlett, P.; Pereira, F.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 24 (NIPS)*, 873–881.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.

Duchi, J.; Jordan, M. I.; and McMahan, B. 2013. Estimation, optimization, and parallelism when data is sparse. In *Advances in Neural Information Processing Systems*, 2832–2840.

Joulani, P.; György, A.; and Szepesvári, C. 2013. Online learning under delayed feedback. In *Proceedings of The 30th International Conference on Machine Learning (ICML)*. (extended arXiv version : http://arxiv.org/abs/1306.0686).

Joulani, P.; György, A.; and Szepesvári, C. 2015. Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms. Url: http://ualberta.ca/˜joulani/publications/AAAI16-Extended.pdf.

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

Liu, J., and Wright, S. J. 2015. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization* 25(1):351–376.

Liu, J.; Wright, S. J.; Ré, C.; Bittorf, V.; and Sridhar, S. 2013. An asynchronous parallel stochastic coordinate descent algorithm. *arXiv preprint arXiv:1311.1873*.

Mania, H.; Pan, X.; Papailiopoulos, D.; Recht, B.; Ramchandran, K.; and Jordan, M. I. 2015. Perturbed Iterate Analysis for Asynchronous Stochastic Optimization. *ArXiv e-prints*.

McMahan, H. B., and Streeter, M. 2010. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Conference on Learning Theory*.

McMahan, B., and Streeter, M. 2014. Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems*, 2915–2923.

McMahan, H. B. 2014. Analysis techniques for adaptive online learning. *CoRR* abs/1403.3465.

Mesterharm, C. J. 2007. *Improving on-line learning*. Ph.D. Dissertation, Department of Computer Science, Rutgers University, New Brunswick, NJ.

Nesterov, Y. 2012. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* 22(2):341–362.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 693–701.

Sra, S.; Yu, A. W.; Li, M.; and Smola, A. J. 2015. Adadelay: Delay adaptive distributed stochastic convex optimization. *arXiv preprint arXiv:1508.05003*.

Weinberger, M. J., and Ordentlich, E. 2002. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory* 48(7):1959–1976.