A Simpler Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism

Pooria Joulani [*1] **Anant Raj** [*2] **András György** [1] **Csaba Szepesvári** [1 3]

Abstract

Recently there have been several attempts to extend Nesterov's accelerated algorithm to smooth stochastic and variance-reduced optimization. In this paper, we show that there is a simpler approach to acceleration: applying *optimistic* online learning algorithms and querying the gradient oracle at the *online average* of the intermediate optimization iterates. In particular, we tighten a recent result of Cutkosky (2019) to demonstrate theoretically that online iterate averaging results in a reduced optimization gap, independently of the algorithm involved. We show that carefully combining this technique with existing generic optimistic online learning algorithms yields the optimal accelerated rates for optimizing strongly-convex and non-strongly-convex, possibly composite objectives, with deterministic as well as stochastic first-order oracles. We further extend this idea to variance-reduced optimization. Finally, we also provide "universal" algorithms that achieve the optimal rate for smooth and non-smooth composite objectives simultaneously without further tuning, generalizing the results of Kavis et al. (2019) and solving a number of their open problems.

1. Introduction

Our goal in this paper is to obtain algorithms with optimal convergence rates for the following problem:

$$\text{find } x^\star = \operatorname*{argmin}_{x \in \mathcal{X}} \ell(x) = f(x) + \phi(x)\,, \qquad (1)$$

where $\mathcal{X}$ is a convex constraint set in the d-dimensional Euclidean space, f is convex and smooth, and ϕ is a (possibly non-smooth) convex function. When $\phi = 0$, and given access to (noise-free) gradients of f, Nesterov's accelerated gradient algorithms (Nesterov, 2018) achieve optimal rates of convergence for Problem (1). Several recent papers, summarized in Table 1, have attempted to obtain similarly accelerated rates that improve upon the sub-optimal rates of Stochastic Gradient Descent (SGD) when the gradients of f are corrupted by noise and/or when $\phi \neq 0$.

Despite the major effort to obtain these extensions, existing results suffer from several limitations such as: (a) inhibiting noise in the gradient (Allen-Zhu and Orecchia, 2017; Wang and Abernethy, 2018); (b) potentially querying the gradient oracle outside the constraint set (Levy et al., 2018; Cutkosky, 2019) (c) not providing optimal rates for strongly-convex objectives (Cutkosky, 2019); (d) extra logarithmic terms appearing in the error bounds (Levy et al., 2018; Cutkosky, 2019); (e) not handling proximal updates when $\phi \neq 0$ (Levy et al., 2018; Kavis et al., 2019; Cutkosky, 2019) or (f) relying on prior knowledge of problem parameters (Tseng, 2008; Beck and Teboulle, 2009; Hu et al., 2009; Xiao, 2010; Lan, 2012; Chen et al., 2012).

In this paper, we demonstrate a simple direct approach to deriving accelerated rates: following Cutkosky (2019), we propose running an online learning algorithm and feeding it with (possibly noisy) first-order information obtained at the *weighted average* of its iterates. Then, building on the recent simple, tight modular analysis techniques of generic optimistic online learning algorithms (Joulani et al., 2017; 2020), we are able to alleviate all the aforementioned limitations, design new accelerated algorithms with straightforward convergence analyses, and solve a number of problems left open in previous work.

1.1. Contributions and Related Work

Our main contributions can be summarized as follows:

- We provide a direct, simple template for deriving and analyzing accelerated algorithms for stochastic and deterministic convex optimization with composite objectives. We further extend the above framework to variance-reduced stochastic non-strongly-convex optimization.

[*]Equal contribution [1]DeepMind, London, UK [2]Max-Planck Institute for Intelligent Systems, Tübingen, Germany; work done during an internship at Deepmind [3]University of Alberta, Edmonton, AB, Canada. Correspondence to: Pooria Joulani <pjoulani@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

	$\mathcal{X}$	f	ϕ	Oracle	Universal	Notes
Tseng (2008)	Any	Non-SC	✓	D	-	
Beck and Teboulle (2009)	$\mathbb{R}^d$	Non-SC	✓	D	-	
Hu et al. (2009)	$\mathbb{R}^d$	SC / Non-SC	✓	S+D	-	Assumes bounded trajectory
Xiao (2010)	Any	Non-SC	✓	S + D	-	Dual-Averaging
Lan (2012)	Any	Non-SC	✓	S+D	-	Not utilizing prox-map of ϕ
Chen et al. (2012)	Any	SC / Non-SC	✓	S+D	-	
	Any	SC	✓	S+D	-	Exponential noise-free rate
Allen-Zhu and Orecchia (2017)	Any	Non-SC	-	D	-	Linear-coupling
	Any	SC	-	D	-	Exponential rate
Wang and Abernethy (2018)	Any	Non-SC	✓	D	-	Primal-dual view
	Any	SC	-	D	-	Exponential rate
This paper (Corollary 4)	**Any**	**SC / Non-SC**	✓	**S + D**	**-**	
	Any	**SC**	✓	**D**	**-**	**Exponential rate**
Cutkosky (2019)	Compact	Non-SC	-	S+D	✓	Accessing f outside $\mathcal{X}$
Levy et al. (2018)	Compact	Non-SC	-	S+D	✓	Accessing f outside $\mathcal{X}$
Kavis et al. (2019)	Compact	Non-SC	-	S + D	✓	
This paper (Theorem 5)	**Compact**	**Non-SC**	✓	**S + D**	✓	

Table 1. Summary of previous work obtaining accelerated rates of convergence. Cutkosky (2019) analyses strongly-convex optimization as well, but the rates are sub-optimal (i.e., non-accelerated). Here, "Non-SC" means non-strongly convex (that is, strong-convexity of f is not required), "SC" means strongly convex, "D" and "S" stand for deterministic and stochastic oracles, respectively. Universality means the algorithm achieves the smooth and non-smooth rates simultaneously without requiring the knowledge of the problem's smoothness and noise level. The "bounded-trajectory" assumption means that the error bound scales with the maximum distance of the iterates from the optimum $x^\star$, but the algorithm does not enforce this to be bounded (e.g., through projection to a compact set). See also the survey by Bubeck (2015) and the recent book of Nesterov (2018).

- For composite non-strongly-convex objectives, we provide a new *universal* algorithm (in the sense of Nesterov, 2015): given only access to the proximal projection oracle of ϕ onto the constraint set, without prior knowledge of the smoothness or noise level, the new algorithm simultaneously achieves the optimal rate of convergence for smooth and non-smooth f. This, together with the fact that the algorithm uses coordinate-wise adaptive step-sizes, resolves two problems left open by Kavis et al. (2019).

In particular, in Lemma 1 and Corollary 2, we tighten the recent analysis of online iterate averaging by Cutkosky (2019). Compared to their Theorem 1, Corollary 2 exposes additional terms that reduce the optimization gap. These terms, whose absence prevented Cutkosky (2019, Theorem 3) from getting the optimal accelerated rates, are similar to what Wang and Abernethy (2018) obtained through an indirect formulation of acceleration as a two-player game.

Next, we show how to utilize the aforementioned reduction to obtain accelerated rates. This is achieved by using properly-tuned *optimistic online learning* algorithms (Rakhlin and Sridharan, 2013a;b; Mohri and Yang, 2016) as the underlying optimization machinery. Importantly, this tuning can be done somewhat independently of the assumptions on the objective (such as the presence of noise, strong convexity, or a non-zero ϕ) or the algorithmic techniques (such as proximal updates or adaptive learning rates), thanks to the recent modular analyses of online learning algorithms by Joulani et al. (2017; 2020). This results in a simple, straightforward acceleration framework.

Furthermore, we extend the analysis to variance-reduced optimization for smooth non-strongly-convex functions. We show that incorporating negative momentum (common in the accelerated SVRG literature, see, e.g., Allen-Zhu, 2017; Lan et al., 2019) in our framework introduces an additional reduction in the optimization gap, enabling us to obtain the optimal convergence rate. We also analyze a simpler version of the variance reduced algorithm without negative momentum, which enjoys a variance-reduced, though still sub-optimal rate of convergence for the last iterate.

Finally, we provide a universal algorithm for non-strongly-convex composite optimization, extending the works of

Levy et al. (2018); Cutkosky (2019); Kavis et al. (2019) to the case when $\phi \neq 0$. The new algorithm features proximal updates and coordinate-wise adaptive step-sizes, thus solving two problems left open by Kavis et al. (2019). Unlike Levy et al. (2018) and Cutkosky (2019), the new algorithm does not query the optimization oracle outside the constraint set $\mathcal{X}$, and does not suffer from extra $\log$ terms in the bound. Unlike the algorithm of Kavis et al. (2019) (which is based on mirror-descent) our algorithm is based on dual-averaging, which is better suited to sparse learning with a proximal ℓ_1 penalty (Xiao, 2009; McMahan, 2011).

Notation. $\mathbb{R}$ denotes the set of real numbers. For any positive integer n, $[n] = \{1, \ldots, n\}$. Let $h : \mathcal{D} \to \mathbb{R}$ where $\mathcal{D} \subset \mathbb{R}^d$ for some positive integer d. The gradient or a subgradient of h is denoted by h'. When h is convex, the Bregman-divergence $B_h : \mathcal{D} \times \mathcal{D}^o \to \mathbb{R}$ is defined as $B_h(x, y) = h(x) - h(y) - \langle h'(y), x - y\rangle$, where $\mathcal{D}^o$ denotes the interior of $\mathcal{D}$. We say that h is μ-strongly convex with respect to (w.r.t.) a norm $\|\cdot\|$ if for all $x \in \mathcal{D}, y \in \mathcal{D}^o$, $\frac{\mu}{2}\|x - y\|^2 \leq B_h(x, y)$, and it is μ-strongly convex w.r.t. a function $n : \mathcal{D} \times \mathcal{D}^o \to [0, \infty)$ if $\mu \cdot n(x, y) \leq B_h(x, y)$ for all $x \in \mathcal{D}, y \in \mathcal{D}^o$ (note that h is μ-strongly convex w.r.t. a norm $\|\cdot\|$ if it is μ-strongly convex w.r.t. the function $\|\cdot\|^2/2$). For non-negative integers a, b and a sequence of numbers or vectors $x_0, x_1, \ldots$, we let $x_{a:b} = \sum_{s=a}^{b} x_s$ if $a \leq b$ and 0 otherwise. With a slight abuse of notation, for a vector $x \in \mathbb{R}^d$, we denote its coordinates as $x = (x_1, \ldots, x_d)$; whether the subscript refers to a coordinate or a time index is usually clear from the context (to reduce the possible ambiguity, we normally use x_i and x_j to index coordinates of x, and x_t and x_s to indicate a quantity corresponding to time steps t or s). For an event E, $\mathbb{I}\{E\}$ denotes its indicator function, that is $\mathbb{I}\{E\} = 1$ if E is true, otherwise $\mathbb{I}\{E\} = 0$. The base-2 logarithm of $x \in (0, +\infty)$ is denoted by $\log(x)$.

2. Preliminaries

For simplicity, we assume that an optimizer $x^\star \in \mathcal{X}$ of Problem (1) exists, i.e., $\ell^\star := \ell(x^\star) \leq \ell(x)$ for all $x \in \mathcal{X}$.[1]

Smoothness of functions. When f is differentiable over $\mathbb{R}^d$, given a norm $\|\cdot\|$, the following are equivalent definitions of smoothness of f (Nesterov, 2018, Theorem 2.1.5): f is L-smooth if

(i) for all $x, y \in \mathbb{R}^d$, $B_f(x, y) \leq \frac{L}{2}\|x - y\|^2$;

(ii) for all $x, y \in \mathbb{R}^d$, $\|f'(x) - f'(y)\|_* \leq L\|x - y\|$;

(iii) for all $x, y \in \mathbb{R}^d$,

$$\|f'(x) - f'(y)\|_*^2 \leq (2L)B_f(x, y)\,. \tag{2}$$

[1] We do not require $\mathcal{X}$ to be closed or compact, which are normally assumed to ensure $x^\star$ exists.

Throughout the paper, we only require[2] that f is differentiable over $\mathcal{X}$, and use (2), holding for all $x, y \in \mathcal{X}$, as the notion of smoothness under which accelerated rates are obtained. Alternatively, assuming (ii) holds only for $x, y \in \mathcal{X}$, one can still obtain the same rates with a very similar analysis as we provide, at the expense of an additional gradient oracle call per step of the algorithms; we leave the details for an extended version of the paper.[3]

Iterative optimization. We consider first-order sequential optimization procedures with access to a stochastic gradient oracle that returns unbiased estimates of f'. A sequential optimization method then, in iteration t, queries the oracle at a point $y_t \in \mathcal{X}$, receives a gradient estimate g_t such that $\mathbb{E}\left[g_t | \mathcal{H}_t\right] = f'(y_t)$ where $\mathcal{H}_t = \sigma\left((g_s)_{s=1}^{t-1}, (y_s)_{s=1}^{t}\right)$ is the sigma-algebra generated by all the information used by the algorithm before making the query at y_t to the gradient oracle. In case $\phi \neq 0$, we also assume that the optimization method has access to the prox-function of ϕ (cf. Eq. 6). After T iterations, the algorithm produces an estimate $\bar{x}_T$ of $x^\star$, based on all the information it has seen, where the quality of the estimate is measured by the error $\mathbb{E}\left[\ell(\bar{x}_T)\right] - \ell^\star$.

Online linear optimization. One way to design and analyze iterative optimization methods is through *online linear optimization* (OLO) algorithms. An OLO algorithm sequentially comes up, at each time step $t \in [T]$, with a prediction x_t, then receives a linear loss function $\langle \alpha_t u_t, \cdot\rangle$, with the aim of maintaining a small cumulative composite loss $\sum_{t=1}^{T} \alpha_t \left(\langle u_t, x_t - x\rangle + \phi(x_t) - \phi(x)\right)$, a.k.a. its *regret* compared to a competitor point x. Here $u_t \in \mathbb{R}^d$ is unknown to the algorithm before selecting x_t, but the non-negative weights α_t are known ahead of time. One can convert an OLO algorithm to an iterative optimization algorithm by using $y_t = x_t$ to query the oracle, using $u_t = g_t$ in the linear loss to the OLO algorithm, and employing the average $\bar{x}_T = \sum_{t=1}^{T} \frac{\alpha_t}{\alpha_{1:T}} x_t$ as the final estimate of $x^\star$.

The appeal of this "vanilla online-to-batch" approach (Algorithm 1), is that it reduces the convergence analysis of $\bar{x}_T$ for convex f and ϕ to the regret analysis of the underlying OLO algorithm. In particular, by Jensen's inequality,

$$\begin{aligned}
&\mathbb{E}\left[\ell(\bar{x}_T)\right] - \ell^\star \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\frac{\alpha_t \left(\langle f'(x_t), x_t - x^\star\rangle + \phi(x_t) - \phi(x^\star)\right)}{\alpha_{1:T}}\right] \\
&= \mathbb{E}\left[\frac{\sum_{t=1}^{T} \alpha_t \left(\langle g_t, x_t - x^\star\rangle + \phi(x_t) - \phi(x^\star)\right)}{\alpha_{1:T}}\right]
\end{aligned}$$

[2] Extensions when f is non-differentiable at boundary points are straightforward.

[3] Assuming only that (i) holds for all $x, y \in \mathcal{X}$ does not imply (ii) or (2) in general (even for $x, y \in \mathcal{X}$). E.g., $f'(x)$ can grow arbitrarily in the directions orthogonal to $\mathcal{X}$ while (i) holds.

Algorithm 1 Vanilla Online-to-Batch

Input: Stochastic gradient oracle, non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, online linear optimization algorithm $\mathcal{A}$
Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$
for $t = 1$ **to** $T-1$ **do**
  Get stochastic gradient g_t at the *current* iterate x_t
  Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
  Let x_{t+1} be the next iterate from $\mathcal{A}$
end for
return the average iterate $\frac{\sum_{t=1}^T \alpha_t x_t}{\alpha_{1:T}}$.

Algorithm 2 Anytime Online-to-Batch (Cutkosky, 2019)

Input: Stochastic gradient oracle, non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, online linear optimization algorithm $\mathcal{A}$
Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$ and let $\bar{x}_1 \leftarrow x_1$
for $t = 1$ **to** $T-1$ **do**
  Get stochastic gradient g_t at the *average* iterate $\bar{x}_t$
  Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
  Let x_{t+1} be the next iterate from $\mathcal{A}$
  Let $\bar{x}_{t+1} \leftarrow \frac{\sum_{s=1}^{t+1} \alpha_s x_s}{\alpha_{1:t+1}}$
end for
return the average iterate $\bar{x}_T$

$$\leq \mathbb{E}\left[\frac{\mathcal{R}_T(x^\star)}{\alpha_{1:T}}\right], \tag{3}$$

where $\mathcal{R}_T(x^\star)$ is an upper-bound for the regret of the OLO algorithm. Thus, to analyze the convergence of $\bar{x}_T$, one can simply plug-in an off-the-shelf regret bound (reviewed at the end of this section) for the underlying OLO algorithm.

Anytime online-to-batch. An alternative, elegant online-to-batch conversion (Algorithm 2) was recently proposed by Cutkosky (2019), which uses the "online" average $\bar{x}_t = \sum_{s=1}^t \frac{\alpha_s}{\alpha_{1:t}} x_s$ as the query point, i.e., $y_t = \bar{x}_t$. Cutkosky (2019, Theorem 1) showed (with $\phi = 0$) that (3) holds under this conversion scheme as well. In the next section, we show that in fact Algorithm 2 enjoys a tighter version of (3) that enables us to prove accelerated rates.

Generic regret bound. Next, we recall the regret bound for a general family of OLO algorithms known as "adaptive optimistic follow the regularized leader" or AO-FTRL (Rakhlin and Sridharan, 2013a;b; Mohri and Yang, 2016). At time t, AO-FTRL makes its t-th prediction as

$$x_t = \operatorname*{argmin}_{x \in \mathcal{X}} \left\langle \sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t, x \right\rangle + \alpha_{1:t}\phi(x) + r_{0:t-1}(x)\,, \tag{4}$$

where, the $r_t : \mathcal{X} \to \mathbb{R}$ are convex *regularizer* functions, and for every t, $\tilde{g}_t$, the *optimistic* part of the update, is interpreted as a prediction of g_t before it is received.

It is straightforward to see that AO-FTRL captures a wide range of algorithms used in optimization (Xiao, 2009; McMahan, 2017). For example, the dual-averaging algorithm of Xiao (2009) corresponds to the case when $\phi = 0$ and $r_{0:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$ for $\eta_t > 0$, in which case it is easy to verify that

$$x_t = \Pi_{\mathcal{X}}\left(-\frac{\sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t}{\eta_t}\right), \tag{5}$$

where $\Pi_{\mathcal{X}}$ denotes Euclidean projection onto set $\mathcal{X}$. More generally, allowing coordinatewise step sizes $\eta_t \in [0,\infty)^d$ and a possibly non-zero ϕ, with $r_{0:t-1}(x) = \frac{1}{2}\sum_{j=1}^d \eta_{t,j} x_j^2$ we recover the proximal (a.k.a. "composite-objective" or "regularized") dual-averaging update (Xiao, 2009):

$$\begin{aligned} x_t &= \mathbf{prox}_{\alpha_{1:t}\phi, \eta_t}\left(-\sum_{s=1}^{t-1} \alpha_s g_s - \alpha_t \tilde{g}_t\right) \\ &= \operatorname*{argmin}_{x \in \mathcal{X}} \alpha_{1:t}\phi(x) + \frac{1}{2}\sum_{j=1}^d \eta_{t,j}\left(x_j - \frac{z_{t-1,j}}{\eta_{t,j}}\right)^2, \end{aligned} \tag{6}$$

where $\mathbf{prox}_{\alpha_{1:t}\phi,\eta_t}$ is the prox-function of $\alpha_{1:t}\phi$ with coordinatewise step sizes $\eta_{t,j}$ and $z_{t-1} = -\left(\sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t\right)$. Note that AdaGrad-style updates (Duchi et al., 2011) can be recovered by setting η_t based on the past gradient estimates $g_s, \tilde{g}_s$ (for $s < t$).

If $r_t \geq 0$, the cumulative regularizer $\alpha_{1:t}\phi + r_{0:t-1}$ is 1-strongly convex w.r.t. a norm $\|\cdot\|_{(t)}$, and the AO-FTRL update is well-defined, that is, the minimizer $x_t \in \mathcal{X}$ exists and $\left\langle \sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t, x_t \right\rangle + \alpha_{1:t}\phi(x_t) + r_{0:t-1}(x_t)$ is finite, then Theorem 6 of Joulani et al. (2020) gives the following regret bound (see Appendix E):

$$\mathcal{R}_T(x^\star) = r_{0:T-1}(x^\star) + \sum_{t=1}^T \frac{1}{2}\alpha_t^2 \|g_t - \tilde{g}_t\|_{(t)*}^2. \tag{7}$$

3. Acceleration with Anytime Online-to-Batch

First, we present a lemma that generalizes the regret decomposition of Joulani et al. (2017) to work with the averaging scheme of Cutkosky (2019). Crucially, the decomposition keeps track of some negative Bregman-divergence terms, which are instrumental in reducing the contribution of the OLO regret to the error of $\bar{x}_T$.

Lemma 1. *For $t = 1, 2, \dots T$, let $\alpha_t > 0$ and $x_t \in \mathbb{R}^d$, and define $\bar{x}_t = (\sum_{s=1}^t \alpha_s x_s)/\alpha_{1:t}$, $B_t = \alpha_t B_f(x^\star, \bar{x}_t)$, and $\bar{B}_t^f = \alpha_{1:t-1} B_f(\bar{x}_{t-1}, \bar{x}_t), t > 1$. Then, if ϕ is convex,*

$$\alpha_{1:T}\left(\ell(\bar{x}_T) - \ell^\star\right)$$

$$\leq \sum_{t=1}^{T} \alpha_t \left(\langle f'(\bar{x}_t), x_t - x^\star \rangle + \phi(x_t) - \phi(x^\star) \right) - B_{1:T} - \bar{B}^f_{2:T} \,. \quad (8)$$

The lemma immediately gives rise to the following generic error bound, which improves upon Theorem 1 of Cutkosky (2019) by keeping around the aforementioned $-\bar{B}^f_t$ and $-B_t$ terms. While the B_t are the usual Bregman-divergence terms (also appearing in the vanilla online-to-batch) that are utilized to get fast rates for strongly convex functions (and can be dropped in general as long as the function is star-convex; see Joulani et al., 2020), the important new terms here are the $-\bar{B}^f_t$ terms, which allow us to prove accelerated rates for online averaging.

Corollary 2 (Generic Error Bound)**.** *Under the assumptions of Lemma 1, if for all $t = 1, 2, \ldots, T$, $g_t \in \mathbb{R}^d$ satisfies $\mathbb{E}\left[g_t | \bar{x}_t\right] = f'(\bar{x}_t)$ and we have*

$$\sum_{t=1}^{T} \alpha_t \left(\langle g_t, x_t - x^\star \rangle + \phi(x_t) - \phi(x^\star) \right) \leq \mathcal{R}_T(x^\star) \quad (9)$$

for some upper-bound $\mathcal{R}_T(x^\star)$, then

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell(x^\star)\right] \leq \mathbb{E}\left[\frac{\mathcal{R}_T(x^\star) - B_{1:T} - \bar{B}^f_{2:T}}{\alpha_{1:T}}\right]. \quad (10)$$

The corollary follows since g_t is a conditionally unbiased estimate of $f'(\bar{x}_t)$, so the first term on the r.h.s. of (8) is, in expectation, equal to the term on the l.h.s. of (9), and hence upper-bounded by $\mathbb{E}\left[\mathcal{R}_T(x^\star)\right]$. Next, we prove the lemma.

Proof of Lemma 1. Writing $f(\bar{x}_T)$ as a telescoping sum,

$$\begin{aligned}
&f(\bar{x}_T) - f(x^\star) \\
&= -f(x^\star) + \frac{\alpha_1 f(\bar{x}_1)}{\alpha_{1:T}} + \sum_{t=2}^{T} \frac{\alpha_{1:t} f(\bar{x}_t) - \alpha_{1:t-1} f(\bar{x}_{t-1})}{\alpha_{1:T}} \\
&= \sum_{t=1}^{T} \frac{\alpha_t \left(f(\bar{x}_t) - f(x^\star)\right)}{\alpha_{1:T}} + \sum_{t=2}^{T} \frac{\alpha_{1:t-1}\left(f(\bar{x}_t) - f(\bar{x}_{t-1})\right)}{\alpha_{1:T}} \\
&= \sum_{t=1}^{T} \frac{\alpha_t \left\langle f'(\bar{x}_t), \bar{x}_t - x^\star \right\rangle - B_t}{\alpha_{1:T}} \\
&\qquad + \sum_{t=2}^{T} \frac{\alpha_{1:t-1} \left\langle f'(\bar{x}_t), \bar{x}_t - \bar{x}_{t-1} \right\rangle - \bar{B}^f_t}{\alpha_{1:T}} \\
&= \sum_{t=1}^{T} \frac{\alpha_t \left\langle f'(\bar{x}_t), \bar{x}_t - x^\star \right\rangle - B_t}{\alpha_{1:T}} \\
&\qquad + \sum_{t=2}^{T} \frac{\alpha_t \left\langle f'(\bar{x}_t), x_t - \bar{x}_t \right\rangle - \bar{B}^f_t}{\alpha_{1:T}} \\
&= \frac{\sum_{t=1}^{T} \alpha_t \left\langle f'(\bar{x}_t), x_t - x^\star \right\rangle - B_{1:T} - \bar{B}^f_{2:T}}{\alpha_{1:T}} \,,
\end{aligned}$$

where the third step follows since by the definition of Bregman divergence, $f(z) - f(y) = \langle f'(z), z - y \rangle - B_f(y, z)$, the fourth step follows since by the definition of $\bar{x}_t$, for $t = 2, 3, \ldots, T$ we have $\alpha_t(\bar{x}_t - x_t) = \alpha_{1:t-1}(\bar{x}_{t-1} - \bar{x}_t)$, and the last step uses $\bar{x}_1 = x_1$. The proof is completed by $\phi(\bar{x}_T) - \phi(x^\star) \leq \sum_{t=1}^{T} \frac{\alpha_t}{\alpha_{1:T}} \left(\phi(x_t) - \phi(x^\star)\right)$, which holds by Jensen's inequality. □

Acceleration. The main idea behind deriving accelerated rates is combining (7) with (10), and selecting α_t and $\tilde{g}_t$ appropriately so that the negative terms $-\bar{B}^f_t$ in (10) offset the contribution of the terms $\frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*}$ in (7) to the final error bound of $\bar{x}_T$. For example, let f be L-smooth over $\mathbb{R}^d$ or assume otherwise that (2) holds with the norm $\|\cdot\| = \|\cdot\|_2$. Suppose the optimization algorithm uses the dual averaging update (5) with $\alpha_t = t$, $\eta_t = \eta = 2L$, deterministic gradients $g_t = f'(\bar{x}_t)$, and $\tilde{g}_t = g_{t-1}$. Then, $r_{0:t-1}$ is 1-strongly convex w.r.t. the norm $L\|\cdot\|_2^2$, and the norm terms $\frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*}$ in (7) can be bounded as

$$\begin{aligned}
\frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*} &= \alpha_t^2 \frac{1}{4L}\|f'(\bar{x}_t) - f'(\bar{x}_{t-1})\|_2^2 \\
&\leq \frac{\alpha_t^2}{2\alpha_{1:t-1}} \bar{B}^f_t = \frac{t^2}{t(t-1)} \bar{B}^f_t \leq \bar{B}^f_t \,,
\end{aligned}$$

where the first inequality follows using (2). Hence, $\mathcal{R}_T(x^\star) - \bar{B}^f_{2:T} \leq L\|x^\star\|_2^2 + \frac{1}{4L}\|f'(x_1)\|_2^2$. Noticing that $\alpha_{1:T} = \Omega(T^2)$ gives the well-known accelerated $\mathcal{O}(1/T^2)$ rate for the error of $\bar{x}_T$. The next theorem, proved in Appendix A, makes this argument precise for the general setting with noise, non-zero ϕ and generic AO-FTRL.

Theorem 3. *In Algorithm 2, let the base method $\mathcal{A}$ generate its iterates by the AO-FTRL update* (4)*, using $\tilde{g}_t = g_{t-1}$ as the optimistic prediction of g_t for $t > 1$ and arbitrary $\tilde{g}_1$. Suppose that f and ϕ are convex, and there exists a norm $\|\cdot\|$ such that either f is 1-smooth w.r.t. $\|\cdot\|$ over $\mathbb{R}^d$ or otherwise* (2) *holds with $L = 1$ for all $x, y \in \mathcal{X}$. Further suppose that for all $t \in [T]$, $r_{t-1} \geq 0$ is convex, the AO-FTRL update* (4) *is well-defined with finite value at the optimum x_t, and there exist $\beta_t > 0$ and a norm $\|\cdot\|_{(t)}$ such that $\alpha_{1:t}\phi + r_{0:t-1}$ is 1-strongly-convex w.r.t. $\frac{\beta_t}{2}\|\cdot\|^2 + \frac{1}{2}\|\cdot\|^2_{(t)}$. Then, if $\alpha_t^2 \beta_t^{-1} \leq \alpha_{1:t-1}$ for all $t > 1$, we have*

$$\begin{aligned}
\mathbb{E}\left[\ell(\bar{x}_T) - \ell^\star\right] \leq &\sum_{t=1}^{T} \mathbb{E}\left[\frac{r_{t-1}(x^\star) - r_{t-1}(x_t) - B_t}{\alpha_{1:T}}\right] \\
&+ \sum_{t=1}^{T} \mathbb{E}\left[\frac{\alpha_t^2 \|\sigma_t - \sigma_{t-1}\|^2_{(t)*}}{2\alpha_{1:T}}\right] \\
&+ \mathbb{E}\left[\frac{\alpha_1^2 \|f'(\bar{x}_1) - \tilde{g}_1\|^2_*}{2\beta_1 \alpha_{1:T}}\right], \quad (11)
\end{aligned}$$

where $\sigma_t = g_t - f'(\bar{x}_t), t \in [T]$, *and* $\sigma_0 = 0$.

4. Applications

In this section we use the framework of the previous section, and Theorem 3 in particular, to obtain accelerated convergence rates with proximal updates, noisy gradients, and universal algorithms.

4.1. Accelerated Proximal Dual-Averaging

First, we show that with appropriately setting α_t and η_t, one can obtain the optimal accelerated rates for the proximal dual averaging update (6). In particular, we consider the case of a single step size for all coordinates (with a slight abuse of notation, $\eta_{t,i} = \eta_t$ for all i) and $r_{0:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$. Then, under the conditions of Theorem 3, we have

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell^\star\right] \le \sum_{t=1}^{T} \mathbb{E}\left[\frac{\alpha_t^2 \|\sigma_t - \sigma_{t-1}\|_{(t)*}^2}{2\alpha_{1:T}}\right] + \mathbb{E}\left[\frac{\eta_T \|x^\star\|_2^2 - \sum_{t=1}^{T}(\eta_t - \eta_{t-1})\|x_t\|_2^2 - 2B_t}{2\alpha_{1:T}}\right] + \mathbb{E}\left[\frac{\alpha_1^2}{2\beta_1\alpha_{1:T}}\|f'(\bar{x}_1) - \tilde{g}_1\|_*^2\right]. \quad (12)$$

Thus, the optimal rates follow immediately by properly setting η_t and α_t, as captured by the following corollary.

Corollary 4 (Accelerated Proximal Dual-Averaging)**.** *Let* f *and* ϕ *be convex and assume that either* f *is* L*-smooth over* $\mathbb{R}^d$ *or otherwise* (2) *holds for all* $x, y \in \mathcal{X}$*. Consider the online-averaged (stochastic) proximal dual averaging algorithm, given by Algorithm 2 with update* (6) *using* $\tilde{g}_t = g_{t-1}$ *as the optimistic prediction of* g_t *for* $t > 1$*, and* $\tilde{g}_1 = 0$*, where the gradient estimates* g_t *are unbiased, that is,* $\mathbb{E}\left[g_t|\bar{x}_t\right] = f'(\bar{x}_t)$*. Let* $\sigma_*^2 = \max_{t=1}^{T}\mathbb{E}\left[\|\sigma_t\|_2^2\right]$*, where* $\sigma_t = g_t - f'(\bar{x}_t)$*, and let* $D = \max\{\|x^\star\|_2, \|x_1 - x_f^*\|_2\}$*, where* x_f^* *is the minimizer of* f *over* $\mathbb{R}^d$*. Then we have the following error bounds:*

(i) If $\eta_t = 4L + \eta\alpha_t\sqrt{t}$ *for some* $\eta > 0$ *and* $\alpha_t = t$*, we have*

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell^\star\right] \le \frac{\left(4L + \frac{L}{4} + \eta T\sqrt{T}\right)D^2 + \frac{4\sigma_*^2}{\eta}T\sqrt{T}}{T(T+1)} = \mathcal{O}\left(\frac{LD^2}{T^2} + \frac{\eta D^2 + \eta^{-1}\sigma_*^2}{\sqrt{T}}\right).$$

(ii) If ϕ_t *is* μ*-strongly-convex then using* $\eta_t = 4L$ *and* $\alpha_t = t$*, we have*

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell^\star\right] \le \frac{\left(4L + \frac{L}{4}\right)D^2 + \frac{8\sigma_*^2 T}{\mu}}{T(T+1)} = \mathcal{O}\left(\frac{LD^2}{T^2} + \frac{\sigma_*^2}{\mu T}\right).$$

(iii) If $g_t = f'(x_t)$ *(i.e., the noiseless case) and* ϕ *is* μ*-strongly-convex, then for* $\eta_t = 0$ *and any sequence of* $\alpha_t > 0, t \in [T]$ *satisfying*

$$\sqrt{c\kappa} \ge \frac{\alpha_{1:t}}{\alpha_t} \ge \sqrt{2\kappa} \qquad t > 1\,, \quad (13)$$

for some $c \ge 2$ *where* $\kappa = (L+\mu)/\mu$ *denotes the condition number, we have*

$$\ell(\bar{x}_T) - \ell^\star \le \frac{\|f'(x_1)\|^2\left(1 - \frac{1}{\sqrt{c\kappa}}\right)^{T-1}}{2\mu}. \quad (14)$$

Remark 1. *The above rates of* $\mathcal{O}\left(1/T^2\right)$ *for a non-strongly-convex* f *are optimal in* T *when there is no noise (*$\sigma_t = 0$*), and the bound* (14) *also almost matches the optimal* $\mathcal{O}\left((1 - 1/\sqrt{\kappa})^T\right)$ *rate for the noiseless strongly-convex case. When there is noise, the worst-case rate of* $\mathcal{O}\left(1/\sqrt{T}\right)$ *(for non-strongly-convex* f*) and* $\mathcal{O}\left(1/T\right)$ *(for strongly-convex* f*) are unavoidable, according to the lower-bounds of Nemirovsky and Yudin (1983): when the noise dominates, there is no hope of exploiting the smoothness in the signal (i.e., the gradient). Therefore, similarly to our paper, all previous work obtain only a lower-order improvement, e.g., from* $1/T + \sigma/\sqrt{T}$ *(of smooth non-strongly-convex SGD) to* $1/T^2 + \sigma/\sqrt{T}$*. If the noise is small, the latter rate is closer to the noise-free optimal rate of* $1/T^2$*, and determines the convergence speed of the algorithm in the initial stages of optimization. In contrast, the former bound (for SGD) is sub-optimal in the noise-free case. The possible improvements are lower-order in case of noisy strongly-convex optimization as well.*

Proof of Corollary 4. First, notice that with any step size $\eta_t = 4L + \gamma_t$, the algorithm is equivalent to Algorithm 2 with AO-FTRL as the base algorithm, using regularizers $r_{0:t-1} = \frac{4L+\gamma_t}{2}\|\cdot\|_2^2$, which satisfy the conditions of Theorem 3 with $\beta_t = 4$, $\|\cdot\|^2 = L\|\cdot\|_2^2$, and $\|\cdot\|_{(t)}^2 = (\gamma_t + \alpha_{1:t}\mu)\|\cdot\|_2^2$, where μ is the strong-convexity parameter of ϕ (i.e, $\mu = 0$ in part (i), and $\mu > 0$ in parts (ii) and (iii)). Hence, starting from (12), with $\alpha_t = t$ we have

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell(x^\star)\right] \le \sum_{t=1}^{T}\frac{t^2\mathbb{E}\left[\|\sigma_t - \sigma_{t-1}\|_2^2\right]}{(\gamma_t + \alpha_{1:t}\mu)T(T+1)} + \frac{(4L+\gamma_T)\|x^\star\|_2^2}{T(T+1)} - \frac{\sum_{t=1}^{T}(\gamma_t - \gamma_{t-1})\mathbb{E}\left[\|x_t\|_2^2\right]}{T(T+1)} + \frac{\mathbb{E}\left[\|f'(\bar{x}_1)\|_2^2\right]}{4T(T+1)L}.$$

In the above, $\mathbb{E}\left[\|\sigma_t - \sigma_{t-1}\|_2^2\right] \le 4\sigma_*^2$. In addition, since f is convex and satisfies (2), we have $\frac{1}{2L}\|f'(\bar{x}_1)\|_2^2 \le$

$B_f(x_1, x_f^*) \leq \frac{L}{2}\|x_1 - x_f^*\|^2$ where x_f^* is the minimizer of f over $\mathbb{R}^d$. Then, plugging in $\gamma_t = \eta\alpha_t\sqrt{t}$ (respectively, $\gamma_t = 0$) and dropping the non-positive terms $-(\gamma_t - \gamma_{t-1})\|x_t\|_2^2$ immediately gives part (i) (respectively, part (ii)).

To prove part (iii), first recall that for $\eta_{t,j} > 0$, (6) is equivalent to the AO-FTRL update (4) with $r_{0:t-1}(x) = \frac{1}{2}\sum_{j=1}^d \eta_{t,j}x_j^2$. For $\eta_t = 0$, we define the update to be AO-FTRL with $r_{0:t-1} = 0$, hence the update will be of the form $x_t = \operatorname{argmin}_{x\in\mathcal{X}} \langle z_{t-1}, x\rangle + \alpha_{1:t}\phi(x)$ (recall that $z_{t-1} = -\sum_{s=1}^{t-1}\alpha_s g_s - \alpha_t\tilde{g}_t$). Then, since ϕ is strongly-convex, despite having $r_s = 0$ for all s, we have that $\alpha_{1:t}\phi + r_{0:t-1}$ is strongly-convex w.r.t. $\beta_t\frac{L}{2}\|\cdot\|_2^2$ with $\beta_t = \alpha_{1:t}\frac{\mu}{L}$. Hence, by Theorem 3,[4] we have

$$\alpha_{1:T}\left(\ell(\bar{x}_T) - \ell(x^\star)\right) \leq \frac{\alpha_1}{2\mu}\|f'(x_1) - \tilde{g}_1\|^2\,, \qquad (15)$$

as long as for all $t > 1$, the assumption $\alpha_t^2\beta_t^{-1} \leq \alpha_{1:t-1}$ of Theorem 3 is satisfied: that is, we have

$$\frac{\alpha_t^2}{\alpha_{1:t}\alpha_{1:t-1}} \leq \frac{\mu}{L} = \frac{1}{\kappa - 1}\,. \qquad (16)$$

It remains to show that (16) is satisfied, and simplify the bound (15). To that end, note that on the one hand, by (13) we have $\alpha_{1:t-1}/\alpha_t \geq \sqrt{2\kappa} - 1$, which in turn implies $\frac{\alpha_{1:t}\alpha_{1:t-1}}{\alpha_t^2} \geq 2\kappa - \sqrt{2\kappa} \geq \kappa - 1$, proving (16). On the other hand, (13) implies $\alpha_{1:t-1} \leq (1 - \frac{1}{\sqrt{c\kappa}})\alpha_{1:t}$ for all $t > 1$; therefore $\alpha_1 \leq \alpha_{1:T}\left(1 - \frac{1}{\sqrt{c\kappa}}\right)^{T-1}$. Putting this back into (15) finishes the proof. □

4.2. A Proximal Adaptive Universal Algorithm

Next, we present the universal convergence of Algorithm 2 with AdaGrad-style step sizes, proved in Appendix B.

Theorem 5. *Suppose that the iterates x_t are given by AO-FTRL with AdaGrad step sizes, i.e., using* (4) *with $r_0 = 0$,*

$$r_t(x) = \gamma\sum_{j=1}^d \frac{\eta_{t,j} - \eta_{t-1,j}}{2}(x_j - x_{t,j})^2, \quad t \geq 1\,,$$

where $\gamma > 0$, $\eta_{t,j} = \sqrt{\sum_{s=1}^t \alpha_s^2(g_{s,j} - \tilde{g}_{s,j})^2}, t > 0$ and $\eta_0 = 0$. Further suppose that g_t are unbiased estimates of $f'(\bar{x}_t)$, and we use $\tilde{g}_t = g_{t-1}, t > 1$ and $\tilde{g}_1 = 0$. Let R be an upper-bound on $|x_j^\star - x_{t,j}|^2$. Then the following hold:

(i) If $\mathbb{E}\left[g_{t,j}^2\right] \leq G_j^2$ for all $t \in [T]$, then

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell^\star\right] \leq \sum_{j=1}^d \mathbb{E}\left[\frac{\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)}{\alpha_{1:T}}\sqrt{\sum_{t=1}^T \alpha_t^2 G_{t,j}^2}\right]$$

$$= \mathcal{O}\left(\frac{R\sum_{j=1}^d G_j}{\sqrt{T}}\right),$$

for $\gamma = 2/R$, where $G_{t,j} := (g_{t,j} - \tilde{g}_{t,j})$.

(ii) If f is L-smooth over $\mathbb{R}^d$ or otherwise (2) *holds for all $x, y \in \mathcal{X}$, and $\mathbb{E}\left[\sigma_{t,j}^2\right] \leq \sigma_j^2$ for all $t \in [T]$ (recall that $\sigma_t = g_t - f'(\bar{x}_t)$), then*

$$\mathbb{E}\left[\ell(\bar{x}_T) - \ell^\star\right] \leq \frac{1}{\alpha_{1:T}}\sum_{j=1}^d 6L\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)^2$$

$$+ \frac{1}{\alpha_{1:T}}\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)\left(\Delta + \sum_{j=1}^d\sqrt{\sum_{t=1}^T 6\alpha_t^2\sigma_j^2}\right)$$

$$= \mathcal{O}\left(\frac{LdR^2 + \Delta R}{T^2} + \frac{\max_j \sigma_j dR}{\sqrt{T}}\right),$$

for $\gamma = 2/R$, where $\Delta = \sum_{j=1}^d \sqrt{2\mathbb{E}\left[|f'(x_{1,j})|^2\right]}$.

Remark 2. *Both bounds above are achieved by the same algorithm, without further prior knowledge about f or the values of L and σ. The first bound is a data-adaptive bound that holds even if f is non-smooth, and is optimal when the data is sparse (Duchi et al., 2013). The second bound is of the optimal rate $\mathcal{O}(1/T^2 + \sigma/\sqrt{T})$ when f is smooth.*

Remark 3. *The bound R required by the theorem is enforced, e.g., when $\mathcal{X}$ is compact. This implies that in the unconstrained optimization setting, similarly to Levy et al. (2018), we assume that we are still given a compact set $\mathcal{X}$ containing $x^\star$ and project to that set in the algorithm.*

5. Accelerated Variance-Reduced Methods

In this section, we apply our framework to the variance reduced setting. In this setting, we assume $f = \mathbb{E}\left[F(\cdot, \xi)\right]$ is the expected value of functions $F : \mathbb{R}^d \times \Xi \to \mathbb{R}$, where ξ is a random variable from some set Ξ, with distribution P_Ξ. At time step t, the algorithm receives a realization $\zeta_t \sim P_\Xi$, and can query the gradient oracle $F'(\cdot, \zeta_t)$ at (potentially multiple) points in $\mathcal{X}$. In addition, the algorithm can query the exact (non-stochastic) gradient oracle f' from time to time. Then, the gradient estimate g_t at $\bar{x}_t$ is calculated as

$$g_t = F'(\bar{x}_t, \zeta_t) - F'(\tilde{x}_t, \zeta_t) + f'(\tilde{x}_t)\,, \qquad (17)$$

where $\tilde{x}_t$ is the *snapshot* point at time t, i.e., the most recent point at which f' has been queried prior to time t.

The underlying operational assumption in computing g_t is that calls to F' are computationally cheaper than calls to f', and hence the latter is queried less frequently. This is in particular the case in finite sum minimization problems, where $f = \frac{1}{n}\sum_{i=1}^n f_i$ for some functions $f_i : \mathcal{X} \to \mathbb{R}$, $F(x, i) = f_i(x)$ for all $i \in [n]$, and ζ_t has a uniform distribution over $[n]$. In this case, the computational complexity

[4] Note that instead of using a norm, here we set $\|\cdot\|_{(t)}$ in Theorem 3 to be zero. While this is not a valid choice, an inspection of the proof of the theorem verifies that the theorem still holds in this case if the dual norm is set to zero and $\sigma_t = 0$ for all t.

of an algorithm is measured by the number of times the gradient of any f_i is computed, so a single access to the full gradient oracle f' has a computation cost of $\mathcal{O}(n)$.

Let $\mathcal{H}_1 = \emptyset$ and $\mathcal{H}_t = \{\zeta_1, \zeta_2, \dots, \zeta_{t-1}\}$ for $t > 1$, i.e., $\mathcal{H}_t$ is the history of random realizations up to time t. To ensure g_t is an unbiased estimate of $f'(\bar{x}_t)$, i.e., $\mathbb{E}[g_t|\mathcal{H}_t] = f'(\bar{x}_t)$, we assume that for any t and any $\mathcal{H}_t$-measurable x,

$$\begin{aligned} \mathbb{E}[F'(x, \zeta_t)|\mathcal{H}_t] &= f'(x)\,, \text{and,} \\ \mathbb{E}[F(x, \zeta_t)|\mathcal{H}_t] &= f(x)\,. \end{aligned} \tag{18}$$

This is ensured, e.g., if $\zeta_t, t = 1, 2, \dots$ is an i.i.d. sequence.

Algorithm. In this setting, instead of defining the query point $\bar{x}_t$ as the average of the previous outputs of the underlying online optimization algorithm $\mathcal{A}$, we define it as

$$\bar{x}_t = \frac{\alpha_{1:t-1}\bar{x}_{t-1} + \alpha_t x_t + p_t \tilde{x}_t}{\alpha_{1:t} + p_t} \tag{19}$$

where the $\alpha_t > 0$ are the averaging weights as before, and $p_t \geq 0$ incorporates a *negative momentum* (first introduced by Allen-Zhu (2017)) towards the current snapshot point $\tilde{x}_t$. If $p_t = 0$, (19) reduces back to $\bar{x}_t = \frac{1}{\alpha_{1:t}} \sum_{s=1}^{t} \alpha_s x_s$.

The resulting algorithm, presented in Algorithm 3, extends Algorithm 1 of Joulani et al. (2020) to the anytime averaging scheme with negative momentum. Algorithm 3 operates in epochs (the outer loop in the algorithm goes over the epochs): At the beginning of epoch s, the gradient snapshot is calculated. Then, in the sth run of the inner loop, from time $T_{1:s-1} + 1$ to $T_{1:s}$, an optimization algorithm $\mathcal{A}$ is run for T_s steps with the variance reduced gradient estimates (17) and averaging (19). Finally, the snapshot point is updated at the end of the epoch; the exact form of the update is given later for the different variants we consider.

Algorithm 3 Variance-Reduced Anytime Online-to-Batch with Negative Momentum

1: **Input:** Gradient oracle F' and f', non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, epoch lengths $T_1, T_2, \dots, T_S$, online linear optimization algorithm $\mathcal{A}$
2: Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$
3: $\tilde{x} \leftarrow x_1$, $\bar{x}_1 \leftarrow x_1$
4: **for** $s = 1$ **to** S **do**
5: Compute and store the full gradient $f'(\tilde{x})$
6: **for** $t = T_{1:s-1} + 1$ **to** $T_{1:s}$ **do** ▷ Denote $\tilde{x}_t = \tilde{x}$
7: Get the gradient estimate g_t at $\bar{x}_t$ by (17)
8: Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
9: Let x_{t+1} be the next iterate from $\mathcal{A}$
10: Let $\bar{x}_{t+1} \leftarrow \frac{\alpha_{1:t}\bar{x}_t + \alpha_{t+1}x_{t+1} + p_{t+1}\tilde{x}_t}{\alpha_{1:t+1} + p_{t+1}}$
11: **end for**
12: Update the snapshot point $\tilde{x}$.
13: **end for**
14: **return** the average iterate $\bar{x}_T$ and the latest snapshot $\tilde{x}$.

5.1. Warm-Up: No Negative Momentum

First, we consider a version of our accelerated variance-reduced method without negative momentum ($p_t = 0$ for all t), using the first iterate of each epoch (i.e., the last iterate of the previous epoch $s - 1$ for $s > 1$) as the snapshot point: In line 12, we let $\tilde{x} = \bar{x}_{t+1}$, so that in every epoch $s \in [S]$, $\tilde{x}_t = \bar{x}_{T_{1:s-1}+1}$ for all $t \in [T_{1:s-1} + 1, T_{1:s}]$. We use AO-FTRL with regularizer $r_{1:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$ as the underlying algorithm $\mathcal{A}$, with the snapshot used as the optimistic gradient estimate: $\tilde{g}_t = f'(\tilde{x}_{t-1})$. Then, we have the following bound on the performance of the algorithm:

Theorem 6. *Suppose that f, as well as $F(\cdot, \zeta)$ for all $\zeta \in \Xi$, are a) convex; and, b) either L-smooth w.r.t. $\|\cdot\|_2$ over $\mathbb{R}^d$ or otherwise satisfying* (2) *for all $x, y \in \mathcal{X}$. Further suppose that* (18) *holds. Assume that Algorithm 3 is run with epoch lengths $T_s = \min\{\tau, 2^{s-1}\}$ for some maximum epoch length τ, snapshot update $\tilde{x} = \bar{x}_{t+1}$ in line 12, $\alpha_t = t$, and $\mathcal{A}$ selected as AO-FTRL with regularizer $r_{1:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$ for $\eta_t = 8L\tau^2$ and optimistic gradient estimates $\tilde{g}_1 = 0$ and $\tilde{g}_t = f'(\tilde{x}_{t-1}), t > 1$. Then, for any $T > \tau$,*

$$\mathbb{E}[\ell(\bar{x}_T)] - \ell(x^\star) \leq \frac{8L\tau^2\|x^\star\|_2^2 + \frac{\|f'(\bar{x}_1)\|_2^2}{8L\tau^2}}{T(T+1)}\,.$$

Proof. By Theorem 17 in Appendix E,

$$\mathcal{R}_T(x^\star) = \frac{\eta_T}{2}\|x^\star\|_2^2 + \sum_{t=1}^{T} \frac{\alpha_t^2}{2\eta_t}\|g_t - \tilde{g}_t\|_2^2$$

bounds the linearized composite-objective regret of $\mathcal{A}$. Combining with Corollary 2 and using $B_{1:T} \geq 0$,

$$\begin{aligned} &\mathbb{E}[\ell(\bar{x}_T)] - \ell(x^\star) \\ &\leq \frac{1}{\alpha_{1:T}}\mathbb{E}\left[\frac{\eta_T}{2}\|x^\star\|_2^2 + \sum_{t=1}^{T}\frac{\alpha_t^2}{2\eta_t}\|g_t - \tilde{g}_t\|_2^2 - \bar{B}_{2:T}^f\right]. \end{aligned}$$

Lemma 9 in Appendix C shows that

$$\sum_{t=2}^{T} \alpha_t^2\|g_t - \tilde{g}_t\|_2^2 \leq 16L\tau^2\bar{B}_{2:T}^f,$$

which then can be used to cancel all terms but $\|g_1 - \tilde{g}_1\|_2^2/(2\eta_1)$ from the summation above. Using $g_1 = f'(\bar{x}_1)$ and $\tilde{g}_1 = 0$, and substituting η_t finishes the proof. □

In the finite sum optimization setting, by selecting $\tau = n$, our algorithm achieves ε error after $\mathcal{O}\left(n\log n + n\sqrt{\frac{L}{\varepsilon}}\right)$ individual gradient evaluations, via a simple direct approach. More complicated methods, such as Catalyst (Lin et al., 2015), RPDG (Lan and Zhou, 2018), Katyusha (Allen-Zhu, 2017) and related papers achieve an iteration complexity of $\mathcal{O}\left(n\log\frac{1}{\varepsilon} + \sqrt{\frac{nL}{\varepsilon}}\right)$, which has a better dependence on n in the dominant second term. However, these

methods use an indirect approach (as termed by Allen-Zhu (2017)), where non-strongly-convex functions are optimized by adding strongly-convex perturbations, and yet do not achieve the near-optimal rate of Lan et al. (2019), which is obtained using negative momentum and epoch averaging. In the next section, we obtain this near-optimal bound.

5.2. Improved Variance-Reduced Acceleration

In this section, we use negative momentum to achieve a near-optimal accelerated variance-reduced rate: we set $p_t > 0$ in Algorithm 3. In addition, unlike Theorem 6, the snapshot point at the end of epoch s is now given by an average:

$$\tilde{x}_{s+1} = \frac{1}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}} p_t} \sum_{t=T_{1:s-1}+1}^{T_{1:s}} p_t \, \bar{x}_t \,. \tag{20}$$

For simplicity, we assume $\phi = 0$, so that $\ell = f$.

A consequence of computing $\bar{x}_t$ via (19) with $p_t > 0$ is that for all $t = 1, 2, \ldots, T$,

$$\alpha_t(\bar{x}_t - x_t) = \alpha_{1:t-1}(\bar{x}_{t-1} - \bar{x}_t) + p_t(\tilde{x}_t - \bar{x}_t) \,. \tag{21}$$

Then, we will have the following error decomposition.

Lemma 7 (Regret Decomposition)**.** *For $t \in [T]$, let $\alpha_t, p_t > 0$, $x_t, \tilde{x}_t \in \mathbb{R}^d$, and define $\bar{x}_t$ as in Equation* (19)*, $B_t = \alpha_t B_f(x^\star, \bar{x}_t)$ and $\bar{B}_t^f = \alpha_{1:t-1} B_f(\bar{x}_{t-1}, \bar{x}_t)$. Then, for all $x^\star \in \mathbb{R}^d$,*

$$f(\bar{x}_T) - f^\star = \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\bar{x}_t), x_t - x^\star \rangle - B_{1:T} \right.$$
$$\left. - \sum_{t=2}^{T} \bar{B}_t^f + \sum_{t=1}^{T} p_t (f(\tilde{x}_t) - f(\bar{x}_t) - B_f(\tilde{x}_t, \bar{x}_t)) \right] \tag{22}$$

The above error decomposition, proved in Appendix D, is similar to Lemma 1, but has an extra term due to the negative momentum, which will be helpful in further reducing the error. Then, the next theorem, proved in Appendix D, provides the improved convergence rate.

Theorem 8. *Consider the conditions of Theorem 6, but instead suppose that the snapshot update in Line 12 of Algorithm 3 is given by* (20)*, $\tilde{g}_t = g_{t-1}, t > 1$, we use p_t such that $0 < p_1 \leq 1$ and $p_t \geq \frac{15 L \alpha_t^2}{\eta_t}, t \geq 1$, and we set $\eta_t = 1860 L T_{s(t)} \log(2t)$, where $s(t)$ denotes the epoch containing iteration t. Then, for any $T \geq 1$,*

$$\mathbb{E}\left[\ell(\bar{x}_T)\right] - \ell(x^\star) \leq$$
$$\frac{3720 L T_{s(T)} \|x^\star\|_2^2 + \frac{\|f'(\bar{x}_1)\|_2^2}{930 L \log 2} + 4(f(\tilde{x}_1) - f^\star)}{T^2} \log(2T).$$

The rate provided in Theorem 8 is optimal up to a logarithmic factor. In particular, for the finite sum setting, with $\tau = n$, the algorithm needs $\tilde{\mathcal{O}}\left(n \log n + \sqrt{\frac{nL}{\varepsilon}}\right)$ individual gradient evaluations to reach ε error, matching the rate recently obtained by Lan et al. (2019). Unlike in previous work, our convergence guarantee holds for the last iterate instead of a snapshot point or the average of the last epoch.

6. Conclusions

We demonstrated that online iterate averaging combined with optimistic online learning can lead to accelerated rates in several scenarios. The resulting algorithms and their analyses are surprisingly simple and often yield the optimal rates. Exploring the full power of this method is left for future work. In particular, it would be interesting to extend this approach to obtain accelerated exponential rates for variance-reduced optimization of strongly-convex objectives, and remove the extra logarithmic terms in the non-strongly-convex variance-reduction case.

References

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67, pages 3:1–3:22, 2017.

Heinz H Bauschke and Patrick L Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2011.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

Xi Chen, Qihang Lin, and Javier Pena. Optimal regularized dual averaging methods for stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 395–403, 2012.

Ashok Cutkosky. Anytime online-to-batch, optimism and acceleration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1446–1454, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

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

John Duchi, Michael I Jordan, and Brendan McMahan. Estimation, optimization, and parallelism when data is sparse. In *Advances in Neural Information Processing Systems*, pages 2832–2840, 2013.

Chonghai Hu, Weike Pan, and James T Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pages 781–789, 2009.

Pooria Joulani, András György, and Csaba Szepesvári. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds. In *International Conference on Algorithmic Learning Theory, ALT*, pages 681–720, 2017.

Pooria Joulani, András György, and Csaba Szepesvári. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theoretical Computer Science*, 808: 108 – 138, 2020. Special Issue on Algorithmic Learning Theory.

Ali Kavis, Kfir Y Levy, Francis Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In *Advances in Neural Information Processing Systems*, pages 6257–6266, 2019.

Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical programming*, 171(1-2):167–215, 2018.

Guanghui Lan, Zhize Li, and Yi Zhou. A unified variance-reduced accelerated gradient method for convex optimization. In *Advances in Neural Information Processing Systems*, pages 10462–10472, 2019.

Kfir Y Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. In *Advances in Neural Information Processing Systems*, pages 6500–6509, 2018.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in neural information processing systems*, pages 3384–3392, 2015.

H. Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 525–533, 2011.

H. Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, 18(90):1–50, 2017.

Mehryar Mohri and Scott Yang. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics*, pages 848–856, 2016.

A.S. Nemirovsky and D.B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley, Chichester, New York, 1983.

Yu Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152 (1-2):381–404, 2015.

Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing, 2018.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019, 2013a.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013b.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, 2008.

Jun-Kun Wang and Jacob D Abernethy. Acceleration through optimistic no-regret dynamics. In *Advances in Neural Information Processing Systems*, pages 3824–3834, 2018.

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2009.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.