Optimistic Online Learning: A Simpler Approach to Accelerated Stochastic Constrained Optimization

Artem Riabinin
KAUST
artem.riabinin@kaust.edu.sa

1 Introduction

Stochastic constrained optimization with stochastic first-order oracles (SCO) plays a pivotal role in machine learning. Indeed, the scalability of classical machine learning tasks, such as support vector machines (SVMs), linear/logistic regression and Lasso, rely on efficient stochastic optimization methods. Importantly, generalization guarantees for such tasks often rely on constraining the set of possible solutions. The latter induces simple solutions in the form of low norm or low entropy, which in trun enables to establish generalization guarantees.

In the SCO setting, the optimal convergence rates for the cases of non-smooth and smooth objectives are given by $\mathcal{O}(GD/\sqrt{T})$ and $\mathcal{O}\left(LD^2/T^2 + \sigma D/\sqrt{T}\right)$, where T is the total number of (noisy) gradient queries, L is the smoothness constant of the objective, σ^2 is the variance of the stochastic gradient estimates, and D is the effective diameter of the decision set.

The optimal rate for the non-smooth case may be obtained by the current state-of-the-art optimization algorithms, such as Stochastic Gradient Descent (SGD), AdaGrad (Duchi et al. [2011]), Adam (Kingma and Ba [2014], Reddi et al. [2018]). However, in order to obtain the optimal rate for the smooth case, one is required to use more involved accelerated methods such as C. Hu and Kwok [2009], Diakonikolas and Orecchia [2017], Cohen et al. [2018]. Unfortunately, all of these accelerated methods require a priori knowledge of the smoothness parameter L, as well as the variance of the gradients σ^2, creating a setup barrier for their use in practice. As a result, accelerated methods are not very popular in machine learning tasks.

In this work, we present a universal method for SCO (Joulani et al. [2020]), applying optimistic online learning algorithms and querying the gradient oracle at the online average of the intermediate optimization iterates. This method achieves optimal rates in the smooth case without requiring any prior knowledge regarding the smoothness of the problem L or the noise magnitude σ. Notably, this method addresses some unresolved inquiries related to another universal SCO method (Kavis et al. [2020]). For example, it extends scalar adaptive learning rates to per-coordinate matrix-like preconditioners. Moreover, it is possible to propose a proximal version of the algorithm outlined in Joulani et al. [2020], unlike the method proposed in Kavis et al. [2020] (however, we do not demonstrate this in our current work).

Such universal methods that implicitly adapt to the properties of the learning objective may be very beneficial in practical large-scale problems, where these properties are usually unknown.

2 Algorithms

2.1 Problem

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

Preprint. Under review.

$$\min_{x \in \mathcal{X}} \left\{ f(x) := \mathbb{E}_{\xi \sim \mathcal{D}} \left[f_\xi(x) \right] \right\}, \tag{1}$$

where $\mathcal{X}$ is a convex constraint set in the d-dimensional Euclidean space $\mathbb{R}^d$, ξ is a sample from some distribution $\mathcal{D}$, $f : \mathcal{X} \to \mathbb{R}$ and $f_\xi : \mathcal{X} \to \mathbb{R}$ are continuously differentiable, convex, and smooth functions.

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

We assume the availability of a first order stochastic oracle for f, where we have access to unbiased (noisy) gradient estimates. Concretely, we assume that by querying this oracle with a point $x \in \mathcal{X}$, we receive $\nabla f_\xi(x) \in \mathbb{R}^d$ such,

$$\mathbb{E}[\nabla f_\xi(x) \mid x] = \nabla f(x).$$

2.2 Online-to-Batch Conversion

One way to design and analyze iterative optimization methods is through online linear optimization (OLO) algorithms. One can convert an OLO algorithm to an iterative optimization algorithm. 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 to the regret analysis of the underlying OLO algorithm. In particular, by Jensen's inequality, we can get the following upper bound on $\mathbb{E}\left[f(\bar{x}_T) - f^\star\right]$:

$$\begin{aligned}
\mathbb{E}\left[f(\bar{x}_T)\right] - f^\star &\leq \sum_{t=1}^{T} \mathbb{E}\left[\frac{\alpha_t \left\langle g_t, x_t - x^\star \right\rangle}{\sum_{s=1}^{T} \alpha_s}\right] \\
&= \mathbb{E}\left[\frac{\sum_{t=1}^{T} \alpha_t \left\langle g_t, x_t - x^\star \right\rangle}{\sum_{s=1}^{T} \alpha_s}\right] \\
&\leq \mathbb{E}\left[\frac{\sum_{t=1}^{T} \alpha_t \mathcal{R}_t(x^\star)}{\sum_{s=1}^{T} \alpha_s}\right],
\end{aligned} \tag{2}$$

where $\mathcal{R}_T\left(x^\star\right)$ is an upper-bound for the regret of the OLO algorithm.

Algorithm 1 Vanilla Online-to-Batch

Require: Stochastic gradient oracle, non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, online linear optimization algorithm $\mathcal{A}$

1: Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$
2: **for** $t = 1$ to $T - 1$ **do**
3: Get stochastic gradient g_t at the current iterate x_t
4: Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
5: Let x_{t+1} be the next iterate from $\mathcal{A}$
6: **end for**
7: **return** the average iterate $\frac{\sum_{t=1}^{T} \alpha_t x_t}{\sum_{s=1}^{T} \alpha_s}$.

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

2.3 AO-FTRL

Next, we recall the regret bound for a general family of OLO algorithms known as "adaptive optimistic follow the regularized leader" or AO-FTRL (Flaspohler et al. [2021]). At time t, AO-FTRL makes its

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

Algorithm 2 Anytime Online-to-Batch (Cutkosky [2019])

Require: 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 $\bar{x}_{t+1} \leftarrow \frac{\sum_{s=1}^{t+1} \alpha_s x_s}{\sum_{s=1}^{t+1} \alpha_s}$
end for
return the average iterate $\bar{x}_T$

t-th prediction as

$$x_t = \operatorname*{argmin}_{x \in \mathcal{X}} \left\langle \sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t, x \right\rangle + \sum_{s=0}^{t-1} r_s(x), \tag{3}$$

where $g_t(\cdot) := \nabla f_{\xi_t}(\cdot)$ is a stochastic gradient of function f, and ξ_t is a sample from $\mathcal{D}$ at time t, $r_t : \mathcal{X} \to \mathbb{R}$ 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. For example, the dual-averaging algorithm corresponds to the case when $\sum_{s=0}^{t-1} r_s(x) = \frac{\eta_t}{2} \|\cdot\|_2^2$ for $\eta_t > 0$, in which case it is easy to verify that

$$x_t = \Pi_{\mathcal{X}} \left(- \frac{\sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t}{\eta_t} \right), \tag{4}$$

where $\Pi_{\mathcal{X}}$ denotes Euclidean projection onto set $\mathcal{X}$.

More generally, allowing coordinate wise step sizes $\eta_t \in [0, \infty)^d$ and setting η_t based on the past gradient estimates $g_s, \tilde{g}_s$ (for $s < t$) we can recover AdaGrad-style updates.

If $r_t \geq 0$, the cumulative regularizer $\sum_{s=0}^{t-1} r_s$ 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 $\left\langle \sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t, x_t \right\rangle + \sum_{s=0}^{t-1} r_s(x_t)$ is finite, then Theorem 6 from Joulani et al. [2017] gives the following regret bound:

$$\mathcal{R}_T(x^\star) = \sum_{s=0}^{T-1} r_s(x^\star) + \sum_{t=1}^{T} \frac{1}{2} \alpha_t^2 \|g_t - \tilde{g}_t\|_{(t)*}^2, \tag{5}$$

where $\mathcal{R}_T(x^\star)$ is an upper-bound for the regret of the OLO algorithm.

Now we present Lemma 1 from Joulani et al. [2020] that generalizes the regret decomposition of Joulani et al. [2017] to work with the averaging scheme of Cutkosky [2019]. The lemma gives the generic error bound, which improves the bound of Cutkosky [2019] by keeping around the aforementioned $-B_t$ and $-\bar{B}_t^f$ terms. 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 (Generic Error Bound)**.** *For $t = 1, 2, \ldots T$, let $\alpha_t > 0$ and $x_t \in \mathbb{R}^d$, and define $\bar{x}_t := \left(\sum_{s=1}^{t} \alpha_s x_s \right) / \sum_{s=1}^{t} \alpha_s$, $B_t := \alpha_t B_f(x^\star, \bar{x}_t)$, and $\bar{B}_t^f := \sum_{s=1}^{t-1} \alpha_s B_f(\bar{x}_{t-1}, \bar{x}_t), t > 1$, $g_t \in \mathbb{R}^d$ satisfies $\mathbb{E}[g_t \mid \bar{x}_t] = \nabla f(\bar{x}_t)$ and we have*

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

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

$$\mathbb{E}[f(\bar{x}_T) - f(x^\star)] \leq \mathbb{E}\left[\frac{\mathcal{R}_T(x^\star) - \left(\sum_{t=1}^{T} B_t + \sum_{t=2}^{T} \bar{B}_t^f \right)}{\sum_{s=1}^{T} \alpha_s} \right]. \tag{6}$$

2.4 Acceleration

The main idea behind deriving accelerated rates is combining (5) with (6), and selecting α_t and $\tilde{g}_t$ appropriately so that the negative terms $-\bar{B}_t^f$ in (6) offset the contribution of the terms $\frac{\alpha_t^2}{2}\left\|g_t-\tilde{g}_t\right\|_{(t)*}^2$ in (5) to the final error bound of $\bar{x}_T$. The given idea is reflected in Theorem 3 of convergence from Joulani et al. [2020]. Next, we apply this theorem to obtain accelerated convergence rates for various methods (Theorem 2 and 3).

Theorem 2 (AccelPDA, Accelerated Proximal Dual-Averaging)**.** *Consider the optimization setting in Problem (1), where f is convex and L-smooth over $\mathbb{R}^d$ or otherwise $\|\nabla f(x)-\nabla f(y)\|_2^2 \leq (2L)B_f(x,y)$ holds for all $x,y\in\mathcal{X}$. Consider the online-averaged (stochastic) proximal dual averaging algorithm, given by Algorithm 2 with update (4) 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, $\mathbb{E}\left[g_t\mid\bar{x}_t\right]=\nabla f\left(\bar{x}_t\right)$. Let $\sigma_\star^2=\max_{t=1}^T\mathbb{E}\left[\left\|\sigma_t\right\|_2^2\right]$, where $\sigma_t=g_t-\nabla f\left(\bar{x}_t\right)$, and let $D=\max\left\{\left\|x^\star\right\|_2,\left\|x_1-x^\star\right\|_2\right\}$. If $\eta_t=4L+\eta\alpha_t\sqrt{t}$ for some $\eta>0$ and $\alpha_t=t$, then*

$$\begin{aligned}\mathbb{E}\left[f\left(\bar{x}_T\right)-f^\star\right] &\leq \frac{\left(4L+\frac{L}{4}+\eta T\sqrt{T}\right)D^2+\frac{4\sigma_\star^2}{\eta}T\sqrt{T}}{T(T+1)}\\ &=\mathcal{O}\left(\frac{LD^2}{T^2}+\frac{\eta D^2+\eta^{-1}\sigma_\star^2}{\sqrt{T}}\right).\end{aligned}$$

Theorem 3 (UniAdaGrad, Adaptive Universal Algorithm with AdaGrad-style step sizes)**.** *Consider the optimization setting in Problem (1), where f is convex and L-smooth over $\mathbb{R}^d$ or otherwise $\|\nabla f(x)-\nabla f(y)\|_2^2\leq(2L)B_f(x,y)$ holds for all $x,y\in\mathcal{X}$. Suppose that the iterates x_t are given by Algorithm 2 with update AO-FTRL with AdaGrad step sizes, i.e., using (3) with $r_0=0$,*

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

where $\gamma>0,\eta_{t,j}=\sqrt{\sum_{s=1}^t\alpha_s^2\left(g_{s,j}-\tilde{g}_{s,j}\right)^2},t>0$ and $\eta_0=0$. Further suppose that g_t are unbiased estimates of $\nabla f\left(\bar{x}_t\right)$, and we use $\tilde{g}_t=g_{t-1},t>1$ and $\tilde{g}_1=0$. Let R be an upper-bound on $\left|x_j^\star-x_{t,j}\right|^2$. If $\mathbb{E}\left[\sigma_{t,j}^2\right]\leq\sigma_j^2$ for all $t\in[T]$, where $\sigma_t=g_t-\nabla f\left(\bar{x}_t\right)$, then

$$\begin{aligned}\mathbb{E}\left[f\left(\bar{x}_T\right)-f^\star\right] &\leq \frac{1}{\sum_{s=1}^T\alpha_s}\sum_{j=1}^d 6L\left(\frac{\gamma R^2}{2}+\frac{2}{\gamma}\right)^2\\ &+\frac{1}{\sum_{s=1}^T\alpha_s}\left(\frac{\gamma R^2}{2}+\frac{2}{\gamma}\right)\left(\Delta+\sum_{j=1}^d\sqrt{\sum_{t=1}^T 6\alpha_t^2\sigma_j^2}\right)\\ &=\mathcal{O}\left(\frac{LdR^2+\Delta R}{T^2}+\frac{\max_j\sigma_j dR}{\sqrt{T}}\right)\end{aligned}$$

for $\gamma=2/R$, where $\Delta=\sum_{j=1}^d\sqrt{2\mathbb{E}\left[\left|\nabla f_j\left(\bar{x}_1\right)\right|^2\right]}$.

According to Theorem 2 and 3 AccelPDA and UniAdaGrad have optimal convergence guarantees $\mathcal{O}\left(1/T^2+\sigma/\sqrt{T}\right)$ in the SCO setting for smooth functions. When there is no noise ($\sigma_t=0$), these methods have optimal convergence rates of $\mathcal{O}\left(1/T^2\right)$. It is worth noting that neither of these methods requires a priori knowledge of σ, but AccelPDA necessitates knowledge of L, whereas UniAdaGrad achieves optimal convergence rates without requiring further knowledge about the values of L.

2.5 Another Approach

To enhance the comprehensiveness of the analysis, we provide a brief description of another accelerated algorithm from Kavis et al. [2020] called UniXGrad, which is inspired by the Mirror-Prox (MP)

algorithm and the Optimistic Mirror Descent (OMD) algorithm. It also ensures optimal convergence rates of $\mathcal{O}\left(1/T^2 + \sigma/\sqrt{T}\right)$ for SCO and $\mathcal{O}\left(1/T^2\right)$ for the noise-free case ($\sigma = 0$). Importantly, UniXGrad does not require any prior knowledge regarding the smoothness of the problem L, nor the noise magnitude σ. This method is outlined in Algorithm 3, where

$$\eta_t = \frac{2D}{\sqrt{1 + \sum_{i=1}^{t-1} \alpha_i^2 \left\| g_i - M_i \right\|_*^2}}, \tag{7}$$

$D^2 = \sup_{x,y \in \mathcal{X}} B_{\mathcal{R}}(x, y)$, $\mathcal{R} : \mathcal{X} \to \mathbb{R}$ is a 1-strongly convex differentiable function.

Algorithm 3 UniXGrad (Kavis et al. [2020])

Require: # of iterations T, $y_0 \in \mathcal{X}$, diameter D, weight $\alpha_t = t$, learning rate $\{\eta_t\}_{t \in [T]}$

1: **for** $t = 1, \ldots, T$ **do**
2: sample ξ_t from $\mathcal{D}$
3: $x_t = \arg\min_{x \in \mathcal{X}} \alpha_t \langle x, M_t \rangle + \frac{1}{\eta_t} B_{\mathcal{R}}(x, y_{t-1}) \quad \left(M_t = \nabla f_{\xi_t}(\tilde{z}_t),\ \tilde{z}_t = \frac{\alpha_t y_{t-1} + \sum_{i=1}^{t-1} \alpha_i x_i}{\sum_{i=1}^{t} \alpha_i} \right)$
4: $y_t = \arg\min_{y \in \mathcal{X}} \alpha_t \langle y, g_t \rangle + \frac{1}{\eta_t} B_{\mathcal{R}}(y, y_{t-1}) \quad \left(g_t = \nabla f_{\xi_t}(\bar{x}_t),\ \bar{x}_t = \frac{\alpha_t x_t + \sum_{i=1}^{t-1} \alpha_i x_i}{\sum_{i=1}^{t} \alpha_i} \right)$
5: **end for**
6: **return** $\bar{x}_T$

According to Kavis et al. [2020] UniXGrad has the following rate of convergence presented in Theorem 4.

Theorem 4. *Consider the optimization setting in Problem (1), where f is convex and L-smooth over $\mathbb{R}^d$ or otherwise $\|\nabla f(x) - \nabla f(y)\|_2^2 \leq (2L) B_f(x, y)$ holds for all $x, y \in \mathcal{X}$. Let $\{x_t\}_{t=1,..,T}$ be a sequence generated by Algorithm 3 and $\mathbb{E}\left[\|\nabla f(x) - \nabla f_\xi(x)\|_*^2 \mid x\right] \leq \sigma^2$, $\forall x \in \mathcal{X}$. With $\alpha_t = t$ and learning rate as in (7), it holds that*

$$\mathbb{E}\left[f(\bar{x}_T) - f^\star\right] \leq \frac{224\sqrt{14} D^2 L}{T^2} + \frac{14\sqrt{2}\sigma D}{\sqrt{T}}.$$

In the next section, we will conduct numerical experiments to compare the performance of the proposed algorithms.

3 Numerical Experiments

We compare the performance of AccelPDA and UniAdaGrad for two different tasks against some other well-known adaptive methods, such as AdaGrad and UniXGrad. We consider a synthetic setting where we analyze the convergence behavior, as well as a logistic regression problem on the "breast-cancer" dataset taken from LIBSVM. In order to compare the adaptive methods on equal grounds, AdaGrad is implemented with a scalar step size based on the template given by Levy [2017] (Algorithm 4) and combined with Algorithm 2. The code for the numerical experiments is available on the GitHub repository[2].

3.1 Least Squares Problem

We take the least squares problem with l_2-norm ball constraint for this setting, i.e.,

$$\min_{x \in \mathcal{X}} \left\{ f(x) := \mathbb{E}_{\xi \sim \mathcal{D}}\left[f_\xi(x)\right] = \frac{1}{n} \sum_{i=1}^{n} f_i(x) = \frac{1}{2n} \|Ax - b\|_2^2 \right\},$$

where $\xi := \{\xi_1, \ldots, \xi_m\}$ is a random subset of $\{1, \ldots, n\}$ of size m chosen uniformly at random, $f_i(x) := \frac{1}{2}(a_i^T x - b_i)^2$, $a_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, $\mathcal{X} := \{x \in \mathbb{R}^d : \|x\|_2 \leq r, \quad r > 0\}$, $A =$

[2] See `https://github.com/artem-riabinin/OL-project-Accelerated-SCO-via-OLO-with-Optimism.git`

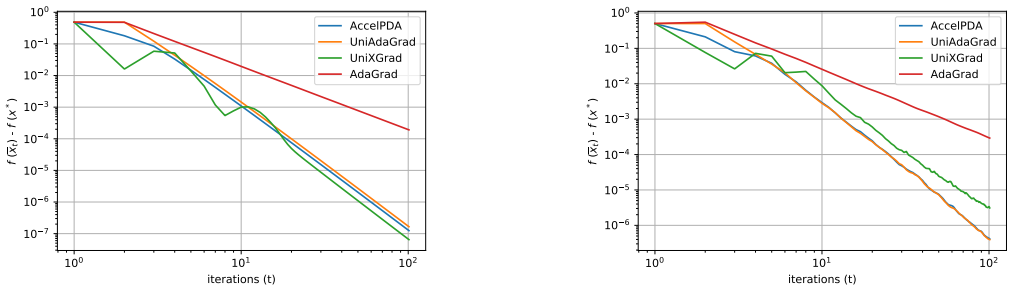
Algorithm 4 AdaGrad (Levy [2017])

Require: Iterations $T, x_1 \in \mathbb{R}^d$, set $\mathcal{X}$ with diameter D
1: Set $Q_0 = 0$
2: **for** $t = 1$ to T **do**
3: Calculate stochastic gradient g_t at x_t
4: $Q_t = Q_{t-1} + \|g_t\|^2$
5: Set $\eta_t = D/\sqrt{2Q_t}$
6: $x_{t+1} = \Pi_{\mathcal{X}}(x_t - \eta_t g_t)$
7: **end for**

$\{a_i, \ldots, a_n\}^T \in \mathbb{R}^{n\times d}$, $A \sim \mathcal{N}(0, I)$, $b = Ax^\star \in \mathbb{R}^n$, $x^\star = \frac{r}{\sqrt{d}}\{1, \ldots, 1\}^T \in \mathbb{R}^d$. We pick $n = 500$, $d = 10$, $r = 1$. Note that $x^\star \in \text{Boundary}(\mathcal{X})$. For the rest of this section, we refer to the exact solution of some problem as $x^\star$.

The gradient of each f_i is given by the following formula: $\nabla f_i(x) = a_i(a_i^T x - b)$. For this problem it is easy to check that f is convex and smooth with a smoothness constant $L = \frac{1}{n}\lambda_{\max}\left(A^\top A\right)$, where $\lambda_{\max}$ denotes the largest eigenvalue.

In this example, we use the following parameters: for AccelPDA: $D_{\text{AccelPDA}} = 2r$, $\eta = 0$ (for deterministic case) and $\eta = \|\nabla f_\xi(z) - \nabla f(z)\|^2 / D^2_{\text{AccelPDA}}$ (for stochastic case), z is a random point from $\mathcal{X}$, $\alpha_t = t$; for UniAdaGrad: $R = 4r^2/d$, $\alpha_t = t$; for UniXGrad: $D_{\text{UniXGrad}} = \sqrt{2}r$ (for $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$), $\alpha_t = t$; for AdaGrad: $D_{\text{AdaGrad}} = 2r$, $D^2_{\text{AdaGrad}} = \sup_{x,y\in\mathcal{X}} \|x - y\|^2$.



(a) $n = m$ (b) $n = m/10$

Figure 1: Convergence rates for the least squares problem in (a) deterministic and (b) stochastic oracle settings.

3.2 Logistic Regression Problem

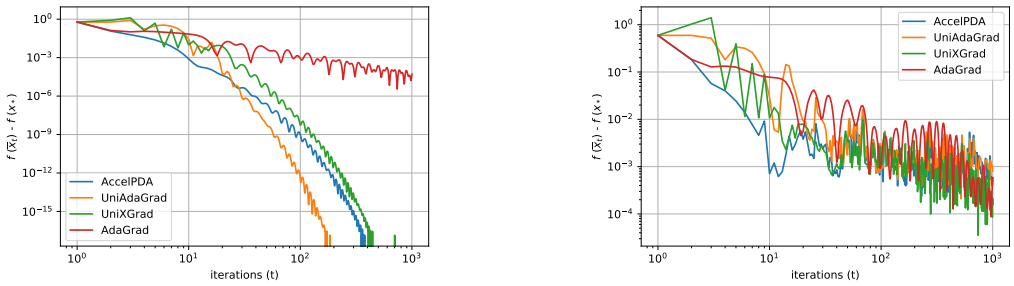
In this section, we will tackle the logistic regression problem on "breast-cancer" dataset taken from LIBSVM. We try to minimize the following logistic regression problem:

$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \mathbb{E}_{\xi\sim\mathbb{R}^d}\left[f_\xi(x)\right] = \frac{1}{n}\sum_{i=1}^n f_i(x) = \frac{1}{n}\sum_{i=1}^n\left(\log\left(1 + \exp(-y_i a_i^\top x)\right) + \frac{\lambda}{2}\|x\|^2\right)\right\},$$

where $\xi := \{\xi_1, \ldots, \xi_m\}$ is a random subset of $\{1, \ldots, n\}$ of size m chosen uniformly at random, $f_i(x) = \log\left(1 + \exp(-y_i a_i^\top x)\right) + \frac{\lambda}{2}\|x\|^2$, $a_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, $A = \{a_i, \ldots, a_n\}^T \in \mathbb{R}^{n\times d}$, $y_i \in \{-1, 1\}$, $\lambda = 1/n$ is a regularization parameter, the exact solution $x^\star$ is calculated numerically using the gradient descent method with a sufficient number of iterations. For the entries of matrix A we choose the "breast-cancer" dataset (scaled to $[-1, 1]$) from LIBSVM ($n = 683,\ d = 10$).

The gradient of each $f_i(x)$ is given by $\nabla f_i(x) = \frac{-y_i}{\exp\left(y_i a_i^\top x\right)+1}a_i + \lambda x$. For this problem it is easy to check that f is strongly convex and smooth with a smoothness constant $L = \frac{1}{4m}\lambda_{\max}\left(A^\top A\right) + \lambda = \frac{1}{4m}\|A\|^2 + \lambda$, where $\lambda_{\max}$ denotes the largest eigenvalue.

It is worth to note that for AccelPDA, UniAdaGrad, UniXGrad, and AdaGrad the bounds D_{AccelPDA}, R, D_{UniXGrad}, and D_{AdaGrad} respectively required by the theory are enforced, e.g., when $\mathcal{X}$ is compact. This implies that in the unconstrained optimization setting, we assume that we are still given a compact set $\mathcal{X}$ containing $x^\star$. In this particular problem, we select the bounds D_{AccelPDA}, R, D_{UniXGrad}, and D_{AdaGrad} for AccelPDA, UniAdaGrad, UniXGrad, and AdaGrad respectively, as in the previous example, choosing $r = 3$ and verifying that $x^\star \in \mathcal{X} := \{x \in \mathbb{R}^d : \|x\|_2 \leq r, \quad r > 0\}$. Other parameters for methods are also chosen in the same way as in the previous example.



(a) $n = m$ (b) $n = m/10$

Figure 2: Convergence rates for the logistic regression problem in (a) deterministic and (b) stochastic oracle settings.

From Figures 1 and 2, it is clear that AccelPDA, UniAdaGrad, and UniXGrad perform almost equally well and outperform AdaGrad in terms of convergence speed across all cases (only in Figure 2 (b) all the methods give almost the same result). In fact, it was expected from the theory since AccelPDA, UniAdaGrad, and UniXGrad guarantee optimal convergence bounds.

Moreover, it can be noticed that AccelPDA and UniAdaGrad either perform slightly better or nearly as well as UniXGrad. Importantly, UniAdaGrad and UniXGrad do not require knowledge of the smoothness constant L of the function f, while AccelPDA depends on it.

4 Conclusions

We demonstrated that online iterate averaging, combined with optimistic online learning, can lead to accelerated rates in several scenarios. The resulting algorithms (AccelPDA and UniAdaGrad) yield optimal rates both in deterministic and stochastic cases. Exploring the methods in practice, we found that AccelPDA and UniAdaGrad give faster convergence than AdaGrad and either slightly outperform or perform nearly as well as another method with optimal convergence bounds (UniXGrad).

References

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

M. B. Cohen, J. Diakonikolas, and L. Orecchia. On acceleration with noise-corrupted gradients. *arXiv:1805.12591*, 2018.

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*, 2019.

J. Diakonikolas and L. Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. *arXiv:1706.04680*, 2017.

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

Genevieve Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online Learning with Optimism and Delay. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvari. A modular analysis of adaptive (non-) convex optimization: Optimism, composite objectives, and variational bounds. In *International Conference on Algorithmic Learning Theory*, 2017.

Pooria Joulani, Anant Raj, Andras Gyorgy, and Csaba Szepesvari. A Simpler Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

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 32*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *The 3rd International Conference on Learning Representations*, 2014.

K. Levy. Online to offline conversions, universality and adaptive minibatch sizes. In *Advances in Neural Information Processing Systems*, pages 1612–1621, 2017.

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.