Adaptive Gradient Methods for Constrained Convex Optimization and Variational Inequalities

Alina Ene,[1] **Huy L. Nguyen,** [2] **Adrian Vladu** [3]
[1] Department of Computer Science, Boston University
[2] Khoury College of Computer and Information Science, Northeastern University
[3] CNRS & IRIF, Université de Paris
aene@bu.edu, hu.nguyen@northeastern.edu, vladu@irif.fr

Abstract

We provide new adaptive first-order methods for constrained convex optimization. Our main algorithms ADAACSA and ADAAGD+ are accelerated methods, which are universal in the sense that they achieve nearly-optimal convergence rates for both smooth and non-smooth functions, even when they only have access to stochastic gradients. In addition, they do not require any prior knowledge on how the objective function is parametrized, since they automatically adjust their per-coordinate learning rate. These can be seen as truly accelerated ADAGRAD methods for constrained optimization.

We complement them with a simpler algorithm ADAGRAD+ which enjoys the same features, and achieves the standard non-accelerated convergence rate. We also present a set of new results involving adaptive methods for unconstrained optimization and monotone operators.

Introduction

Gradient methods are a fundamental building block of modern machine learning. Their scalability and small memory footprint makes them exceptionally well suited to the massive volumes of data used for present-day learning tasks.

While such optimization methods perform very well in practice, one of their major limitations consists of their inability to converge faster by taking advantage of specific features of the input data. For example, the training data used for classification tasks may exhibit a few very informative features, while all the others have only marginal relevance. Having access to this information a priori would enable practitioners to appropriately tune first-order optimization methods, thus allowing them to train much faster. Lacking this knowledge, one may attempt to reach a similar performance by very carefully tuning hyper-parameters, which are all specific to the learning model and input data.

This limitation has motivated the development of adaptive methods, which in absence of prior knowledge concerning the importance of various features in the data, adapt their learning rates based on the information they acquired in previous iterations. The most notable example is ADAGRAD (Duchi, Hazan, and Singer 2011), which adaptively modifies the learning rate corresponding to each coordinate in the vector of weights. Following its success, a host of new adaptive methods appeared, including ADAM (Kingma and Ba 2014), AMSGRAD (Reddi, Kale, and Kumar 2018), and SHAMPOO (Gupta, Koren, and Singer 2018), which attained optimal rates for generic online learning tasks.

A significant series of recent works on adaptive methods addresses the regime of smooth convex functions. Notably, Levy (2017), Cutkosky (2019), Kavis et al. (2019), and Bach and Levy (2019) consider the case of minimizing smooth convex functions without having prior knowledge of the smoothness parameter. While a standard convergence rate of $1/T$ is fairly easily attainable in the case of unconstrained optimization, achieving the optimal $1/T^2$ rate becomes significantly more challenging. Even worse, for constrained minimization objectives, where the gradient is nonzero at the optimum, it is generally unclear how an adaptive method can pick the correct step sizes even when aiming for the weaker non-accelerated rate of $1/T$. These difficulties occur when one merely attempts to find the correct learning rate; taking advantage of non-uniform per-coordinate learning rates, as in the case of the original ADAGRAD method has remained largely open. In (Kavis et al. 2019), finding such a method with an accelerated $1/T^2$ convergence is posed as an open problem, since it would allow the development of robust algorithms that are applicable to non-convex problems such as training deep neural networks.

In this paper, we address this problem and present adaptive algorithms which achieve nearly-optimal convergence with per-coordinate learning rates, even in constrained domains. Our algorithms are *universal* in the sense that they achieve nearly-optimal convergence rate even when the objective function is non-smooth (Nesterov 2015). Furthermore, they automatically extend to the case of stochastic optimization, achieving up to logarithmic factors optimal dependence in the standard deviation of the stochastic gradient norm. We complement them with a simpler non-accelerated algorithm which enjoys the same features: it achieves the standard convergence rate on both smooth and non-smooth functions, and does not require prior knowledge of the smoothness parameters, or the variance of the stochastic gradients.

Previous Work: Work on adaptive methods has been extensive, and resulted in a broad range of algorithms (Duchi, Hazan, and Singer 2011; Kingma and Ba 2014; Reddi, Kale, and Kumar 2018; Tieleman and Hinton 2012; Dozat 2016;

Chen et al. 2018). A significant body of work is dedicated to non-convex optimization (Zou et al. 2018; Ward, Wu, and Bottou 2019; Zou et al. 2019; Li and Orabona 2019; Défossez et al. 2020). In a slightly different line of research, there has been recent progress on obtaining improved convergence bounds in the online non-smooth setting; these methods appear in the context of parameter-free optimization, whose main feature is that they adapt to the radius of the domain (Cutkosky and Sarlos 2019; Cutkosky 2020).

Here we discuss, for comparison, relevant previous results on adaptive first order methods for smooth convex optimization where the function $f : \mathbb{R}^d \to \mathbb{R}$ to be minimized is smooth with respect to some unknown norm $\|\cdot\|_{\mathcal{B}}$, where $\mathcal{B}$ is a non-negative diagonal matrix. The case where $\mathcal{B} = \beta I$ is a multiple of the identity corresponds to the standard assumption that f is β-smooth, and we refer to this as the *scalar* version of the problem. In the case when $\mathcal{B}$ is a non-negative diagonal matrix, we optimize using the *vector* version of the problem.

Notably, Levy (Levy 2017) presents an adaptive first order method, achieving an optimal convergence rate of $O\left(\beta R^2/T\right)$, without requiring prior knowledge of the smoothness β. While this method also applies to the case where the domain is constrained, it requires the strong condition that the global optimum lies within the domain. In (Levy, Yurtsever, and Cevher 2018a), this issue is discussed explicitly, and the line of work is pushed further in the unconstrained case to obtain an accelerated rate of $O\left(\beta R^2 \ln\left(\beta R/\left\|g_0\right\|\right)/T^2\right)$, where g_0 is the gradient evaluated at the initial point. In (Bach and Levy 2019), the authors consider constrained variational inequalities, which are more general, as they include both convex optimization and convex-concave saddle point problems. The rate they achieve is $O\left(\beta R^2/T\right)$, where β is the an upper bound on the unknown Lipschitz parameter of the monotone operator, generalizing the case of β-smooth convex functions. Based on this scheme, in (Kavis et al. 2019) the authors deliver an accelerated adaptive method with nearly optimal rate for the scalar version of the problem. There, they pose as an open problem the question of delivering an accelerated adaptive method for the vector case.

Our Contributions: We give the first adaptive algorithms with per-coordinate step sizes achieving nearly-optimal rates for both constrained convex minimization and variational inequalities arising from monotone operators. Variational inequalities are a very general framework that captures convex minimization, convex-concave saddle point problems, and many other problems of interest (Bach and Levy 2019; Nemirovski 2004). Our algorithms are universal, in the sense defined by Nesterov (2015). They automatically achieve optimal convergence rates (up to a $\sqrt{\ln T}$ factor) in the smooth and non-smooth setting, both in the deterministic setting as well as the stochastic setting where we have access to noisy gradient or operator evaluations. Our algorithms automatically adapt to problem parameters such as smoothness, gradient or operator norms, and the variance of the stochastic gradient or operator norms. Our results answer several open questions raised in previous work (Kavis et al. 2019; Bach and Levy 2019).

For constrained convex minimization, we present three algorithms: AdaGrad+, AdaACSA, and AdaAGD+. For β-smooth functions, AdaGrad+ convergences at the rate $O\left(R_\infty^2 d \cdot \beta \ln \beta / T\right)$, and AdaACSA and AdaAGD+ converge at the rate $O\left(R_\infty^2 d \cdot \beta \ln \beta / T^2\right)$. Since $R_\infty d^{1/2}$ is the ℓ_2 diameter of the region containing the ℓ_∞ ball of radius R_∞, these exactly match the rates of standard non-accelerated and accelerated gradient decent, when the domain is an ℓ_∞ ball (Nesterov 2013). Therefore these schemes can be interpreted as learning the optimal *diagonal preconditioner* for a smooth function f.

For variational inequalities, we present the Adaptive Mirror-Prox algorithm that couples the Universal Mirror-Prox scheme (Bach and Levy 2019; Nemirovski 2004) with novel per-coordinate step sizes. The Universal Mirror-Prox algorithm of (Bach and Levy 2019) sets a single step size for all coordinates that is initialized using an estimate for the gradient norms. In contrast, our algorithm uses per-coordinate step sizes that are initialized to an absolute constant. In addition to eliminating a hyperparameter that we would need to tune, this approach leads to larger stepsizes. Adaptive methods such as AdaGrad are also implemented and used in practice using step sizes initialized to a small constant, such as $\epsilon = 10^{-10}$. We show that the algorithm simultaneously achieves convergence guarantees that are optimal (up to a $\sqrt{\ln T}$ factor) for both smooth and non-smooth operators, as well as in the deterministic and stochastic settings.

Algorithmically, we provide a new rule for updating the diagonal preconditioner, which is better suited to constrained optimization. While the unconstrained AdaGrad algorithm updates the preconditioner based on the previously seen gradients, here we update based on the movement performed by the iterate (see Figure 1). In the unconstrained setting, our update rule matches the standard AdaGrad update. The works (Kavis et al. 2019; Joulani et al. 2020) tackled the difficulties introduced by constraining the domain by using a different update rule based on the change in gradients.

Contemporaneous work: Joulani et al. (2020) also obtain an accelerated algorithm with coordinate-wise adaptive rates, in constrained domains. The convergence guarantee is stronger than ours by a $O(\ln \beta)$ factor in the smooth setting, where β is the smoothness constant, and by a $O(\sqrt{\ln T})$ factor in the non-smooth and stochastic settings. On the other hand, we obtain adaptive schemes for a wide-range of settings, including a non-dual-averaging scheme (AdaACSA, based on the AC-SA algorithm (Lan 2012)), a dual-averaging scheme (AdaAGD+, based on the AGD+ algorithm (Cohen, Diakonikolas, and Orecchia 2018)), and an adaptive mirror-prox scheme (Bach and Levy 2019; Nemirovski 2004) for solving variational inequalities which generalizes both convex minimization and convex-concave zero-sum games. The latter answers an open question (Bach and Levy 2019). Joulani et al. (2020) propose a very different dual-averaging scheme for convex minimization based on the online-to-batch conversion (Cutkosky 2019; Kavis

method	non-smooth convergence	smooth convergence
ADAGRAD	$O\left(\frac{R_\infty\sqrt{d}G}{\sqrt{T}}+\frac{R_\infty\sqrt{d}\sigma}{\sqrt{T}}\right)$ Follows from (Duchi, Hazan, and Singer 2011)	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i}{T}+\frac{R_\infty\sqrt{d}\sigma}{\sqrt{T}}\right)$
ADAGRAD+	$O\left(\frac{R_\infty\sqrt{d}G\sqrt{\ln\left(\frac{GT}{R_\infty}\right)}}{\sqrt{T}}+\frac{R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}+\frac{R_\infty^2 d}{T}\right)$	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i\ln\beta_i}{T}+\frac{R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}\right)$
ADAACSA	$O\left(\frac{R_\infty\sqrt{d}G\sqrt{\ln\left(\frac{GT}{R_\infty}\right)}+R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}+\frac{R_\infty^2 d}{T^2}\right)$	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i\ln\beta_i}{T^2}+\frac{R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}\right)$
ADAAGD+	$O\left(\frac{R_\infty\sqrt{d}G\sqrt{\ln\left(\frac{GT}{R_\infty}\right)}+R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}+\frac{R_\infty^2 d}{T^2}\right)$	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i\ln\beta_i}{T^2}+\frac{R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}\right)$
Adaptive Mirror Prox	$O\left(\frac{R_\infty\sqrt{d}G\sqrt{\ln\left(\frac{GT}{R_\infty}\right)}+R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}+\frac{R_\infty^2 d}{T}\right)$	$O\left(\frac{R_\infty^2\sum_{i=1}^d\beta_i\ln\beta_i}{T}+\frac{R_\infty\sqrt{d}\sigma\sqrt{\ln\left(\frac{T\sigma}{R_\infty}\right)}}{\sqrt{T}}\right)$

Table 1: Convergence rates of adaptive methods in the vector setting. We assume that $f:\mathcal{K}\to\mathbb{R}$, with $\mathcal{K}\subseteq\mathbb{R}^d$, is either smooth with respect to an unknown norm $\|\cdot\|_{\mathcal{B}}$, where $\mathcal{B}=\operatorname{diag}(\beta_1,\ldots,\beta_d)$, or non-smooth and G-Lipschitz. We assume access to stochastic gradients $\widetilde{\nabla}f(x)$ which are unbiased estimators for the true gradient and have bounded variance $\mathbb{E}\left\|\widetilde{\nabla}f(x)-\nabla f(x)\right\|^2\le\sigma^2$. The Adaptive Mirror Prox algorithm is for the more general setting of variational inequalities.

et al. 2019) and the online learning with optimism framework (Mohri and Yang 2016). Our algorithms use the iterate movement to set the per-coordinate step sizes, whereas the algorithm presented in (Joulani et al. 2020) uses the change in gradients.

Preliminaries

Constrained Convex Optimization: We consider the problem $\min_{x\in\mathcal{K}}f(x)$, where $f\colon\mathbb{R}^d\to\mathbb{R}$ is a convex function and $\mathcal{K}\subseteq\mathbb{R}^d$ is an arbitrary convex set. For simplicity, we assume that f is continuously differentiable and we let $\nabla f(x)$ denote the gradient of f at x. We assume access to projections over $\mathcal{K}$ in the sense that we can efficiently solve problems of the form $\arg\min_{x\in\mathcal{K}}\langle g,x\rangle+\frac{1}{2}\|x\|_D^2$, where D is an arbitrary non-negative diagonal matrix and $\|x\|_D=\sqrt{x^\top Dx}$.

We say that f is smooth with respect to the norm $\|\cdot\|_{\mathcal{B}}$ if $\nabla^2 f(x)\preceq\mathcal{B}$, for all $x\in\mathcal{K}$. Equivalently, we have $f(y)\le f(x)+\langle\nabla f(x),y-x\rangle+\frac{1}{2}\|x-y\|_{\mathcal{B}}^2$, for all $x,y\in\mathcal{K}$. We say that f is strongly convex with respect to the norm $\|\cdot\|_{\mathcal{B}}$ if $\nabla^2 f(x)\succeq\mathcal{B}$, for all $x\in\mathcal{K}$.

Variational Inequalities: We also consider the more general problem setting of variational inequalities arising from monotone operators. Let $\mathcal{K}\subseteq\mathbb{R}^d$ be a convex set and let $F\colon\mathcal{K}\to\mathbb{R}^d$ be an operator. The operator F is monotone if it satisfies $\langle F(x)-F(y),x-y\rangle\ge 0$ for all $x,y\in K$ and it is smooth with respect to the norm $\|\cdot\|_{\mathcal{B}}$ if $\|F(x)-F(y)\|_{\mathcal{B}^{-1}}\le\|x-y\|_{\mathcal{B}}$ for all $x,y\in K$. The goal is to find a strong solution x^* for the variational inequality arising from F, i.e., a solution $x^*\in\mathcal{K}$ satisfying $\langle F(x^*),x^*-x\rangle\le 0$ for all $x\in\mathcal{K}$.

Variational inequalities are a very general framework that captures convex minimization, convex-concave saddle point problems, and many other problems of interest (Bach and Levy 2019; Nemirovski 2004). For convex minimization, the operator $F(x)$ is simply the gradient $\nabla f(x)$.

Let $x_0\in\mathcal{K}$, $D_0=I$, $R_\infty\ge\max_{x,y\in\mathcal{K}}\|x-y\|_\infty$.
For $t=0,\ldots,T-1$, update:

$$x_{t+1}=\arg\min_{x\in\mathcal{K}}\left\{\langle\nabla f(x_t),x\rangle+\frac{1}{2}\|x-x_t\|_{D_t}^2\right\},$$

$$D_{t+1,i}^2=D_{t,i}^2\left(1+\frac{(x_{t+1,i}-x_{t,i})^2}{R_\infty^2}\right),\text{ for all } i\in[d].$$

Return $\overline{x}_T=\frac{1}{T}\sum_{t=1}^T x_t$.

Figure 1: ADAGRAD+ algorithm.

Adaptive Schemes for Constrained Convex Optimization and Variational Inequalities

Constrained ADAGRAD Scheme: Figure 1 presents the basic ADAGRAD algorithm for constrained convex optimization. The algorithm can be viewed as a generalization of ADAGRAD to the constrained setting. To see the parallel with ADAGRAD, consider the gradient mapping:

$$g_t=-D_t(x_{t+1}-x_t)\Leftrightarrow x_{t+1}=x_t-D_t^{-1}g_t\,.$$

Letting $\eta=R_\infty$, the update is

$$x_{t+1,i}=x_{t,i}-\frac{\eta}{\sqrt{\eta^2+\sum_{s=1}^{t-1}g_{s,i}^2}}g_{t,i}\,,\qquad\forall i\in[d]\,.$$

In the unconstrained setting, we have $g_t=\nabla f(x_t)$ and our scheme almost coincides with ADAGRAD. We have chosen the initial scaling to be the identity, whereas the original

ADAGRAD scheme uses $D_0 = \epsilon I$. Our analysis extends to this choice and we incur an additional $O(\ln(1/\epsilon))$ factor in the convergence guarantee. In addition, the diagonal matrix D_t we use is off by one iterate, in the sense that it does not contain information about g_t. This is an essential feature of our method, since in the constrained setting computing the gradient mapping requires access to D_t.

Similarly to (Bach and Levy 2019), we can motivate the choice of updating D by the iterate movement as follows. The algorithm simultaneously addresses the unconstrained setting and the more challenging constrained setting. Since our goal is to design a universal method, intuitively we would like the step size to decay in the non-smooth setting and to remain constant in the smooth setting, similarly to the standard (non-adaptive) gradient descent schemes. In the unconstrained setting, the iterate movement coincides with the gradient. In the constrained setting, the gradient is non-zero at the optimum and we cannot hope that the gradient norm decreases as we approach the optimum. Instead, as the iterate converges to the optimum, the movement also goes to zero and thus our adaptive step size remains around the optimal value.

We show that our algorithm is universal and it obtains the smooth rate of $\frac{1}{T}$ if the function is smooth while retaining the optimal $\frac{1}{\sqrt{T}}$ rate if the function is non-smooth. Our algorithm and analysis extend to the stochastic setting. The algorithm automatically adapts to the smoothness parameters, the gradient norm, and the variance parameter.

Accelerated Schemes: We give two adaptive schemes for constrained convex optimization that achieve the optimal rate of $\frac{1}{T^2}$ for smooth functions without knowing the smoothness parameters. Our algorithms are adaptive versions of the AC-SA algorithm (Lan 2012), and the AGD+ algorithm (Cohen, Diakonikolas, and Orecchia 2018). For this reason, we coin the names ADAACSA (Figure 2) and ADAAGD+ (Figure 3). The AGD+ algorithm is a dual-averaging version of AC-SA. The algorithms and their adaptive versions have different iterates and they may be useful in different contexts.

We show that our algorithms simultaneously achieve convergence rates that are optimal (up to a $\sqrt{\ln T}$ factor) for both smooth and non-smooth functions, both in the deterministic and stochastic setting. The algorithms automatically adapt to the smoothness parameters, the gradient norm, and the variance parameter.

Variational Inequalities: Building on the work of Bach and Levy (2019), we give the first universal method with per-coordinate adaptive step sizes for variational inequalities arising from monotone operators, and answer the open question asked by them. The algorithm, shown in Figure 4, is the natural extension to the vector setting of the scheme of (Bach and Levy 2019). A notable difference is that the algorithm provided in (Bach and Levy 2019) uses an estimate for $G \geq \max_{x \in \mathcal{K}} \|F(x)\|$ as part of the step size. Our algorithm does not use the G parameter and it automatically adapts to it, as well as the smoothness and variance parameters.

We show that the algorithm simultaneously achieves convergence rates that are optimal (up to a $\sqrt{\ln T}$ factor) for both smooth and non-smooth operators, both in the deterministic and stochastic setting. We note that, in the stochastic setting, the analysis of (Bach and Levy 2019) makes the additional assumption that the stochastic estimates of the operator are bounded almost surely, which is stronger than our assumption of bounded variance. This assumption simplifies the analysis, as it allows one to directly upper bound D_T (equivalently, lower bound the step size $\eta_T = 1/D_T$), which is a key loss term in the convergence analysis. Our analysis removes this assumption by employing a more involved argument that does not upper bound $\mathrm{Tr}(D_T)$ directly.

Table 1 summarizes the convergence guarantees for all of the algorithms, and we give the complete analyses in the full version (Ene, Nguyen, and Vladu 2021).

Let $D_0 = I$, $z_0 \in \mathcal{K}$, $\alpha_t = \gamma_t = 1 + \frac{t}{3}$, $R_\infty^2 \geq \max_{x,y \in \mathcal{K}} \|x - y\|_\infty^2$.
For $t = 0, \ldots, T-1$, update:

$$x_t = \left(1 - \alpha_t^{-1}\right) y_t + \alpha_t^{-1} z_t \,,$$

$$z_{t+1} = \arg\min_{u \in \mathcal{K}} \left\{ \gamma_t \langle \nabla f(x_t), u \rangle + \frac{1}{2} \|u - z_t\|_{D_t}^2 \right\} \,,$$

$$y_{t+1} = \left(1 - \alpha_t^{-1}\right) y_t + \alpha_t^{-1} z_{t+1} \,,$$

$$D_{t+1,i}^2 = D_{t,i}^2 \left(1 + \frac{(z_{t+1,i} - z_{t,i})^2}{R_\infty^2}\right) \text{, for all } i \in [d].$$

Return y_T.

Figure 2: ADAACSA algorithm.

Let $D_1 = I$, $z_0 \in \mathcal{K}$, $a_t = t$, $A_t = \sum_{i=1}^t a_i = \frac{t(t+1)}{2}$, $R_\infty^2 \geq \max_{x,y \in \mathcal{K}} \|x - y\|_\infty^2$.
For $t = 1, \ldots, T$, update:

$$x_t = \frac{A_{t-1}}{A_t} y_{t-1} + \frac{a_t}{A_t} z_{t-1} \,,$$

$$z_t = \arg\min_{u \in \mathcal{K}} \left(\sum_{i=1}^t \langle a_i \nabla f(x_i), u \rangle + \frac{1}{2} \|u - z_0\|_{D_t}^2 \right)$$

$$y_t = \frac{A_{t-1}}{A_t} y_{t-1} + \frac{a_t}{A_t} z_t \,,$$

$$D_{t+1,i}^2 = D_{t,i}^2 \left(1 + \frac{(z_{t,i} - z_{t-1,i})^2}{R_\infty^2}\right) \text{, for all } i \in [d].$$

Return y_T.

Figure 3: ADAAGD+ algorithm.

Convergence Analysis

In this section, we provide a brief sketch of the convergence analysis in the setting where f is smooth with respect to the norm $\|\cdot\|_{\mathcal{B}}$, where $\mathcal{B} = \mathrm{diag}(\beta_1, \ldots, \beta_d)$, and we have access to exact gradients ($\sigma = 0$). Similarly, for variational

Let $y_0 \in \mathcal{K}$, $D_1 = I$, $R_\infty \geq \max_{x,y \in K} \|x - y\|_\infty$.
For $t = 1, \ldots, T$, update:

$$x_t = \arg\min_{x \in \mathcal{K}} \left\{ \langle F(y_{t-1}), x \rangle + \frac{1}{2} \|x - y_{t-1}\|_{D_t}^2 \right\} ,$$

$$y_t = \arg\min_{x \in \mathcal{K}} \left\{ \langle F(x_t), x \rangle + \frac{1}{2} \|x - y_{t-1}\|_{D_t}^2 \right\} ,$$

$$D_{t+1,i}^2 = D_{t,i}^2 \left(1 + \frac{(x_{t,i} - y_{t-1,i})^2 + (x_{t,i} - y_{t,i})^2}{2R_\infty^2} \right) .$$

Return $\overline{x}_T = \frac{1}{T} \sum_{t=1}^T x_t$.

Figure 4: Adaptive Mirror-Prox algorithm, extending (Bach and Levy 2019) to the vector setting.

inequalities, we consider the setting where F is smooth with respect to $\|\cdot\|_{\mathcal{B}}$ and we can evaluate F exactly. The complete analyses can be found in the full version (Ene, Nguyen, and Vladu 2021).

AdaGrad+ Algorithm: Using the standard analysis of gradient descent for smooth functions, we obtain

$$f(x_{t+1}) - f(x^*) \leq \frac{1}{2} \left(\|x_t - x^*\|_{D_t}^2 - \|x_{t+1} - x^*\|_{D_t}^2 - \|x_{t+1} - x_t\|_{D_t}^2 \right) + \frac{1}{2} \|x_{t+1} - x_t\|_{\mathcal{B}}^2 .$$

Summing up over all iterations and using the inequality $\|x - y\|_D^2 \leq \mathrm{Tr}(D) \|x - y\|_\infty^2 \leq \mathrm{Tr}(D) R_\infty^2$, we obtain

$$\sum_{t=0}^{T-1} (f(x_{t+1}) - f(x^*)) \leq \frac{1}{2} R_\infty^2 \mathrm{Tr}(D_T) - \frac{1}{2} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{D_t}^2 + \frac{1}{2} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{\mathcal{B}}^2 .$$

The heart of the analysis is to show that the right-hand side is bounded by a constant (independent of T). Our analysis can be viewed as a vector generalization of the scalar analyses presented in previous work (Levy, Yurtsever, and Cevher 2018b; Bach and Levy 2019; Kavis et al. 2019). Note that the above guarantee has two loss terms, $R_\infty^2 \mathrm{Tr}(D_T)$ and $\sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{\mathcal{B}}^2$, and the gain term $\sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{D_t}^2$. We will use the gain to absorb most of the loss. To this end, we split the guarantee into two terms as follows:

$$\underbrace{R_\infty^2 \mathrm{Tr}(D_T) - \frac{1}{2} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{D_t}^2}_{(\star)} + \underbrace{\sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{\mathcal{B}}^2 - \frac{1}{2} \sum_{t=0}^{T-1} \|x_{t+1} - x_t\|_{D_t}^2}_{(\star\star)} .$$

We bound each of the terms $(\star)$ and $(\star\star)$ with the aid of the following inequalities, which are standard and are equivalent to the inequalities used in previous work.

Lemma 0.1. *Let $d_0^2, d_1^2, d_2^2, \ldots, d_T^2$ and R^2 be scalars. Let $D_0 > 0$ and let $D_1, \ldots, D_T$ be defined according to the recurrence $D_{t+1}^2 = D_t^2 \left(1 + \frac{d_t^2}{R^2}\right)$. We have $\sum_{t=a}^{b-1} D_t \cdot d_t^2 \geq 2R^2 (D_b - D_a)$. If $d_t^2 \leq R^2$ for all t, we have $\sum_{t=a}^{b-1} D_t \cdot d_t^2 \leq \left(\sqrt{2} + 1\right) R^2 (D_b - D_a)$ and $\sum_{t=a}^{b-1} d_t^2 \leq 4R^2 \ln \left(\frac{D_b}{D_a}\right)$.*

To bound $(\star)$, we apply Lemma 0.1 for each coordinate i separately with $d_t^2 = (x_{t+1,i} - x_{t,i})^2$and $R^2 = R_\infty^2$. The first inequality in the lemma implies that

$$(\star) \leq R_\infty^2 \mathrm{Tr}(D_0) = R_\infty^2 d .$$

To bound $(\star\star)$, we note that, for each coordinate i, $D_{t,i}$ is increasing with t. We let $\tilde{T}_i$ be the last iteration t for which $D_{t,i} \leq 2\beta_i$. We have

$$(\star\star) = \sum_{i=1}^d \sum_{t=0}^{T-1} \left(\beta_i (x_{t+1,i} - x_{t,i})^2 - \frac{1}{2} D_{t,i} (x_{t+1,i} - x_{t,i})^2 \right) \leq \sum_{i=1}^d \sum_{t=0}^{\tilde{T}_i} \beta_i (x_{t+1,i} - x_{t,i})^2 .$$

For each coordinate $i \in [d]$ separately, we apply Lemma 0.1 with $d_t^2 = (x_{t+1,i} - x_{t,i})^2$ and $R^2 = R_\infty^2 \geq d_t^2$. The third inequality in the lemma implies

$$\sum_{t=0}^{\tilde{T}_i} (x_{t+1,i} - x_{t,i})^2 \leq R_\infty^2 + \sum_{t=0}^{\tilde{T}_i - 1} (x_{t+1,i} - x_{t,i})^2 \leq R_\infty^2 + 4R_\infty^2 \ln \left(\frac{D_{\tilde{T}_i,i}}{D_{0,i}} \right) = R_\infty^2 + 4R_\infty^2 \ln (2\beta_i) .$$

Therefore

$$(\star\star) \leq O \left(R_\infty^2 \sum_{i=1}^d \beta_i \ln (2\beta_i) \right) .$$

Putting everything together, we obtain

$$f(\overline{x}_T) - f(x^*) \leq \frac{1}{T} \sum_{t=1}^T (f(x_t) - f(x^*)) \leq O \left(\frac{R_\infty^2 \sum_{i=1}^d \beta_i \ln (2\beta_i)}{T} \right) .$$

Our analysis above did not directly upper bound $\mathrm{Tr}(D_T)$. Using our techniques, we can show that $\mathrm{Tr}(D_T) \leq \mathrm{Tr}(D_0) + O\left(\sum_{i=1}^d \beta_i \ln(2\beta_i)\right) + (f(x_0) - f(x^*))/(2R_\infty^2)$. Thus the ADAGRAD step sizes remain constant and very close to the ideal step sizes given by the smoothness parameters. The result theoretically confirms the intuition provided in previous work (e.g., (Bach and Levy 2019)) and it expands our understanding of ADAGRAD.

ADAACSA Algorithm: We assume that the parameters of the algorithm are chosen so that $a_0 = 1$, $0 < (\alpha_{t+1} - 1)\gamma_{t+1} \leq \alpha_t \gamma_t$, and $\gamma_t \leq \alpha_t$ for all t. By extending the analysis given in (Lan 2012), we obtain

$$\begin{aligned}
&(\alpha_T - 1)\gamma_T (f(y_T) - f(x^*)) \\
&\leq \frac{1}{2} R_\infty^2 \mathrm{Tr}(D_{T-1}) \\
&+ \sum_{t=0}^{T-1} \left(\frac{1}{2}\frac{\gamma_t}{\alpha_t} \|z_t - z_{t+1}\|_{\mathcal{B}}^2 - \frac{1}{2}\|z_t - z_{t+1}\|_{D_t}^2 \right) .
\end{aligned}$$

We analyze the right-hand side using an argument that is analogous to the one we used above for ADAGRAD+. We write

$$\underbrace{\left(\frac{1}{2} R_\infty^2 \mathrm{Tr}(D_{T-1}) - \frac{1}{2\sqrt{2}} \sum_{t=0}^{T-1} \|z_t - z_{t+1}\|_{D_t}^2 \right)}_{(\star)} +$$

$$\underbrace{\sum_{t=0}^{T-1} \left(\frac{1}{2}\frac{\gamma_t}{\alpha_t} \|z_t - z_{t+1}\|_{\mathcal{B}}^2 - \left(\frac{1}{2} - \frac{1}{2\sqrt{2}} \right) \|z_t - z_{t+1}\|_{D_t}^2 \right)}_{(\star\star)}$$

Using Lemma 0.1, we obtain $(\star) \leq \frac{1}{2} R_\infty^2 d$ and $(\star\star) \leq O\left(R_\infty^2 \sum_{i=1}^d \beta_i \ln(2\beta_i)\right)$. Thus

$$(\alpha_T - 1)\gamma_T (f(y_T) - f(x^*)) \leq O\left(R_\infty^2 \sum_{i=1}^d \beta_i \ln(2\beta_i) \right)$$

One choice is $\gamma_t = \alpha_t = \frac{t}{3} + 1$. Another choice is $\gamma_t = \alpha_t$, $\alpha_0 = 1$, $\alpha_{t+1} = \frac{1 + \sqrt{1 + 4\alpha_t^2}}{2}$. Both choices satisfy the above assumptions and they give

$$f(y_T) - f(x^*) = O\left(\frac{R_\infty^2 \sum_{i=1}^d \beta_i \ln(2\beta_i)}{T^2} \right) .$$

Adaptive Mirror-Prox Algorithm: Following (Bach and Levy 2019), we analyze the convergence of the algorithm in Figure 4 for variational inequalities via the regret $\sum_{t=1}^T \langle F(x_t), x_t - x^* \rangle$. In the following, we sketch the regret analysis and show that, if F is smooth, the regret is upper bounded by a constant. Bach and Levy (2019) showed that this implies the optimal $\frac{1}{T}$ convergence rate for smooth operators. By extending the analysis of (Bach and Levy 2019), we obtain

$$\begin{aligned}
&2\sum_{t=1}^T \langle F(x_t), x_t - x^* \rangle \leq \\
&\underbrace{R_\infty^2 \mathrm{Tr}(D_T) - \frac{1}{2}\sum_{t=1}^T \left(\|x_t - y_{t-1}\|_{D_t}^2 + \|x_t - y_t\|_{D_t}^2 \right)}_{(\star)} \\
&+ \underbrace{\sum_{t=1}^T \left(\|x_t - y_{t-1}\|_{\mathcal{B}}^2 - \frac{1}{2}\|x_t - y_{t-1}\|_{D_t}^2 \right)}_{(\star\star)} \\
&+ \underbrace{\sum_{t=1}^T \left(\|x_t - y_t\|_{\mathcal{B}}^2 - \frac{1}{2}\|x_t - y_t\|_{D_t}^2 \right)}_{(\star\star\star)} .
\end{aligned}$$

Using a similar argument to the one above, we obtain that $(\star) \leq 2R_\infty^2 d$ and $(\star\star) + (\star\star\star) \leq O\left(R_\infty^2 \sum_{i=1}^d \beta_i \ln(2\beta_i)\right)$. Thus the regret is at most $O\left(R_\infty^2 \sum_{i=1}^d \beta_i \ln(2\beta_i)\right)$.

	1e-1	1e-2	1e-3	1e-4	1e-5
SGD	16	400	>2000	>2000	>2000
ADAGRAD	101	104	>2000	>2000	>2000
ADAM	64	149	570	1055	1697
ADAACSA	**10**	**73**	**275**	**387**	**431**
ADAAGD+	30	154	525	934	1633
JRGS	18	136	278	495	880

Table 2: Experimental results for the synthetic experiment using Nesterov's "worst function in the world." For each method, we display the number of iterations before the first iterate with target error is encountered.

Experimental Evaluation

To empirically validate the ADAACSA and ADAAGD+ algorithms, we test them on a series of standard models encountered in machine learning. While the analyses we provided are specifically crafted for convex objectives, we see that these methods exhibit good behavior in the non-convex settings corresponding to training deep learning models. This may be motivated by the fact that a significant part of the optimization performed when training such models occurs within convex regions (Leclerc and Madry 2020).

Algorithms: We evaluated our ADAACSA and ADAAGD+ algorithms against three popular methods, SGD with momentum, ADAGRAD, and ADAM. We also evaluated the algorithms against the recent method of (Joulani et al. 2020) which we refer to as JRGS. We performed extensive hyper-parameter tuning, such that each method we compare against has the opportunity to exhibit its best possible performance. We give the complete

logistic	train loss	test loss	test accuracy
SGD	2.44e-1±0.38e-2	2.86e-1±0.46e-2	92.25±0.21
ADAGRAD	2.31e-1±0.27e-2	2.84e-1±0.42e-2	92.41±0.20
ADAM	2.35e-1±0.17e-2	2.80e-1±0.22e-2	92.53±0.12
ADAACSA	**2.23e-1±0.10e-2**	2.90e-1±0.28e-2	92.36±0.13
ADAAGD+	2.38e-1±0.15e-2	**2.68e-1±0.14e-2**	**92.57±0.09**
JRGS	3.35e-1±1.22e-2	4.42e-1±1.15e-2	90.61±0.40

CNN	train loss	test loss	test accuracy
SGD	10.46e-4±207.62e-5	4.06e-2±1.39e-2	99.30±0.23
ADAGRAD	7.32e-4±25.06e-5	2.83e-2±0.17e-2	99.19±0.07
ADAM	**0.05e-4±0.04e-5**	3.28e-2±0.24e-2	**99.39±0.04**
ADAACSA	0.18e-4±0.71e-5	3.93e-2±0.26e-2	99.28±0.08
ADAAGD+	10.18e-4±57.92e-5	**2.49e-2±0.18e-2**	99.24±0.04
JRGS	4.43e-4±7.32e-5	3.33e-2±0.44e-2	99.27±0.08

ResNet18	train loss	test loss	test accuracy
SGD	0.10e-1±0.19e-2	0.50±1.11e-2	90.83±0.24
ADAGRAD*	0.07e-1±0.05e-2	0.61±2.01e-2	88.50±0.35
ADAM	**0.06e-1±0.09e-2**	**0.48±1.15e-2**	**91.44±0.18**
ADAACSA	0.20e-1±0.31e-2	1.00±9.13e-2	83.48±1.15
ADAAGD+	0.23e-1±0.18e-2	0.60±2.79e-2	88.10±0.60
JRGS	0.32e-1±0.16e-2	0.55±3.24e-2	88.20±0.55

Table 3: Experimental results for the classification experiments: logistic regression on MNIST; convolutional neural network on MNIST; residual network on CIFAR-10. We report the average value and standard deviation over 5 runs (except for ADAGRAD on CIFAR-10, where one run failed to converge).

experimental details in the full version (Ene, Nguyen, and Vladu 2021).

Synthetic experiment: First, we tested all the methods on a synthetic example, known as Nesterov's "worst function in the world," which is a canonical example used for testing accelerated gradient methods (Nesterov 2013):

$$f(x) = \frac{1}{2}\left(x_1^2 + x_n^2 + \sum_{i=1}^{n-1}(x_i - x_{i+1})^2\right) - x_1$$

Table 2 summarizes the results. We observe that ADAACSA outperforms all the other methods.

Classification experiments: Additionally, we tested these optimization methods on three different classification models typically encountered in machine learning. The first one is logistic regression on the MNIST dataset. This is a simple convex objective for which ADAACSA achieves the best training loss, while ADAAGD+ achieves the best test loss. The second is a convolutional neural network on the MNIST dataset. Despite non-convexity, both our methods behave well, and ADAAGD+ achieves the best test loss. The third is a residual network on the CIFAR-10 dataset. Table 3 summarizes the results.

Discussion: We verified experimentally that ADAACSA and ADAAGD+ behave very well on convex objectives, as anticipated by theory. For practical non-convex objectives, they show a remarkable degree of robustness, managing to reach close to zero training loss. By contrast, JRGS requires a significant amount of tuning in order to converge – our experiments show that in non-convex settings, without properly constraining the domain to a small ℓ_∞ ball, it is very hard for it to achieve any nontrivial progress.

Acknowledgments

AE was supported in part by NSF CAREER grant CCF-1750333, NSF grant CCF-1718342, and NSF grant III-1908510. HN was supported in part by NSF CAREER grant CCF-1750716 and NSF grant CCF-1909314. AV was supported in part by NSF grant CCF-1718342. We thank Aleksander Mądry for kindly providing us with computing resources to perform the experimental component of this work.

References

Bach, F.; and Levy, K. Y. 2019. A Universal Algorithm for Variational Inequalities Adaptive to Smoothness and Noise. In *Conference on Learning Theory (COLT)*, 164–194.

Chen, X.; Liu, S.; Sun, R.; and Hong, M. 2018. On the Convergence of A Class of Adam-Type Algorithms for Non-Convex Optimization. In *International Conference on Learning Representations (ICLR)*.

Cohen, M.; Diakonikolas, J.; and Orecchia, L. 2018. On Acceleration with Noise-Corrupted Gradients. In *International Conference on Machine Learning (ICML)*, 1018–1027.

Cutkosky, A. 2019. Anytime Online-to-Batch, Optimism and Acceleration. In *International Conference on Machine Learning (ICML)*, 1446–1454.

Cutkosky, A. 2020. Parameter-Free, Dynamic, and Strongly-

Adaptive Online Learning. In *International Conference on Machine Learning (ICML)*.

Cutkosky, A.; and Sarlos, T. 2019. Matrix-Free Preconditioning in Online Learning. In *International Conference on Machine Learning (ICML)*, 1455–1464.

Défossez, A.; Bottou, L.; Bach, F.; and Usunier, N. 2020. On the Convergence of Adam and Adagrad. *arXiv preprint arXiv:2003.02395* .

Dozat, T. 2016. Incorporating nesterov momentum into Adam. In *International Conference on Learning Representations (ICLR) Workshop*.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)* 12(7).

Ene, A.; Nguyen, H. L.; and Vladu, A. 2021. Adaptive Gradient Methods for Constrained Convex Optimization and Variational Inequalities. *arXiv preprint arXiv:2007.08840* .

Gupta, V.; Koren, T.; and Singer, Y. 2018. Shampoo: Preconditioned Stochastic Tensor Optimization. In *International Conference on Machine Learning (ICML)*, 1842–1850.

Joulani, P.; Raj, A.; György, A.; and Szepesvári, C. 2020. A Simpler Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism. In *International Conference on Machine Learning (ICML)*.

Kavis, A.; Levy, K. Y.; Bach, F.; and Cevher, V. 2019. UniXGrad: A Universal, Adaptive Algorithm with Optimal Guarantees for Constrained Optimization. In *Neural Information Processing Systems (NeurIPS)*, 6257–6266.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

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

Leclerc, G.; and Madry, A. 2020. The Two Regimes of Deep Network Training. *arXiv preprint arXiv:2002.10376* .

Levy, K. 2017. Online to offline conversions, universality and adaptive minibatch sizes. In *Neural Information Processing Systems (NeurIPS)*, 1613–1622.

Levy, K. Y.; Yurtsever, A.; and Cevher, V. 2018a. Online adaptive methods, universality and acceleration. In *Neural Information Processing Systems (NeurIPS)*, 6500–6509.

Levy, K. Y.; Yurtsever, A.; and Cevher, V. 2018b. Online Adaptive Methods, Universality and Acceleration. In *Neural Information Processing Systems (NeurIPS)*, 6501–6510.

Li, X.; and Orabona, F. 2019. On the convergence of stochastic gradient descent with adaptive stepsizes. In *Artificial Intelligence and Statistics (AISTATS)*, 983–992.

Mohri, M.; and Yang, S. 2016. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics (AISTATS)*, 848–856.

Nemirovski, A. 2004. Prox-method with rate of convergence O (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization* 15(1): 229–251.

Nesterov, Y. 2013. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

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

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

Tieleman, T.; and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2): 26–31.

Ward, R.; Wu, X.; and Bottou, L. 2019. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning (ICML)*, 6677–6686.

Zou, F.; Shen, L.; Jie, Z.; Sun, J.; and Liu, W. 2018. Weighted AdaGrad with unified momentum. *arXiv preprint arXiv:1808.03408* .

Zou, F.; Shen, L.; Jie, Z.; Zhang, W.; and Liu, W. 2019. A sufficient condition for convergences of adam and rmsprop. In *Computer Vision and Pattern Recognition (CVPR)*, 11127–11135.