HIGH PROBABILITY BOUNDS FOR A CLASS OF NON-CONVEX ALGORITHMS WITH ADAGRAD STEPSIZE

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose a new, simplified high probability analysis of AdaGrad for smooth, non-convex problems. More specifically, we focus on a particular accelerated gradient (AGD) template [Lan, 2020], through which we recover the original AdaGrad and its variant with averaging, and prove a convergence rate of $\mathcal{O}(1/\sqrt{T})$ with high probability without the knowledge of smoothness and variance. We use a particular version of Freedman's concentration bound for martingale difference sequences [Kakade & Tewari, 2008] which enables us to achieve the best-known dependence of $\log(1/\delta)$ on the probability margin δ. We present our analysis in a modular way and obtain a complementary $\mathcal{O}(1/T)$ convergence rate in the deterministic setting for the same class of algorithms. To the best of our knowledge, this is the first high probability result for AdaGrad with a truly adaptive scheme, i.e., completely oblivious to the knowledge of smoothness and uniform variance bound, which simultaneously has best-known dependence on probability margin δ.

1 INTRODUCTION

Adaptive gradient methods are a staple of machine learning (ML) in solving core problems such as

$$\min_{x\in\mathbb{R}^d} f(x) := \mathbb{E}_{z\sim\mathcal{D}}\left[f(x;z)\right], \tag{P}$$

where the objective $f(x)$ is possibly non-convex, and $\mathcal{D}$ is a probability distribution from which the random vector z is drawn. Problem (P) captures for instance empirical risk minimization or finite-sum minimization (Shalev-Shwartz & Ben-David, 2014) problems, where z represents the mini-batches and $\mathcal{D}$ corresponds to the distribution governing the data or its sampling strategy.

Within the context of large-scale problems, including streaming data, computing full gradients is extremely costly, if not impossible. Hence, stochastic iterative methods are the main optimizer choice in these scenarios. The so-called adaptive methods such as AdaGrad (Duchi et al., 2011), Adam (Kingma & Ba, 2014) and AmsGrad (Reddi et al., 2018) have witnessed a surge of interest both theoretically and practically due to their off-the-shelf performance. For instance, adaptive optimization methods are known to show superior performance in various learning tasks such as machine translation (Zhang et al., 2020; Vaswani et al., 2017).

From a theoretical point of view, existing literature provides a quite comprehensive understanding regarding the *expected* behaviour of existing adaptive learning methods. Nevertheless, these results do not capture the behaviour of adaptive methods within a single/few runs, which is related to the *probabilistic* nature of these methods. While there exist *high probability* analysis of vanilla SGD for non-convex problems (Ghadimi & Lan, 2013), adaptive methods have received limited attention in this context.

Our main goal in this paper is to understand the probabilistic convergence properties of adaptive algorithms, specifically AdaGrad, while focusing on their *problem parameter* adaptation capabilities in the non-convex setting. In this manuscript, adaptivity refers to the ability of an algorithm to ensure convergence without requiring the knowledge of quantities such as smoothness modulus or variance of noise. Studies along this direction largely exist for the convex objectives; for instance, Levy et al. (2018) shows that AdaGrad can (implicitly) exploit smoothness and adapt to the magnitude of noise in the gradients when $f(x)$ is convex in (P).

This alternative perspective to adaptivity is crucial because most existing analysis, both for classical and adaptive methods, assume to have access to smoothness constant, bound on gradients (Reddi et al., 2018) and even noise variance (Ghadimi & Lan, 2013). In practice, it is difficult, if not impossible, to compute or even estimate such quantities. For this purpose, in the setting of (P) we study a class of adaptive gradient methods that enable us to handle noisy gradient feedback without requiring the knowledge of the objective's smoothness modulus, nor a bound on gradient norms neither noise variance.

We summarize our contributions as follows:

1. We provide a modular and simplified high probability analysis for AdaGrad-type adaptive methods.
2. We present the first *optimal high probability convergence result* of the *original AdaGrad* algorithm for non-convex smooth problems. Concretely,
   (a) we analyze a fully adaptive step-size that is oblivious to Lipschitz constant L and the noise level σ,
   (b) our analysis demonstrates the best known dependence of $\log(1/\delta)$ on the probability margin δ.
3. We present new adaptive versions of AdaGrad extensions that include averaging and momentum primitives, and prove similar high probability bounds for these methods as well. Concretely, we present a general adaptive template and provide complementary convergence results for different step-size regimes which individually recovers standard AdaGrad, AdaGrad with averaging and adaptive RSAG (Ghadimi & Lan, 2016).

In the next section, we will provide a broad overview of related work with an emphasis on the recent developments. Section 3 formalizes the problem setting and states our blanket assumptions. Section 4 introduces the building blocks of our proposed proof technique while proving convergence results for AdaGrad. We generalize the convergence results of AdaGrad for a class of nonconvex, adaptive algorithms in Section 5. Finally, we present concluding remarks in the last section.

2 Related Work

Adaptive methods for stochastic optimization As an extended version of the online (projected) GD (Zinkevich, 2003), AdaGrad (Duchi et al., 2011) is the pioneering work behind most of the contemporary adaptive optimization algorithms Adam, AmsGrad and RmsProp (Tieleman & Hinton, 2012) to name a few. Simply put, such AdaGrad-type methods compute step-sizes on-the-fly by accumulating gradient information and achieve adaptive regret bounds as a function of gradient history (see also (Tran & Phong, 2019; Alacaoglu et al., 2020b; Luo et al., 2019; Huang et al., 2019)).

Universality, adaptive methods and acceleration We call an algorithm *universal* if it achieves optimal convergence rates under different settings, without any modifications. For *convex* minimization problems, Levy et al. (2018) showed that AdaGrad attains a rate of $\mathcal{O}(1/T + \sigma/\sqrt{T})$ by implicitly adapting to smoothness and noise levels in the feedback; here T is the number of noisy gradient queries and σ is the noise variance. They also proposed an accelerated AdaGrad variant with scalar step-size. The latter result was extended for compactly constrained problems by (Kavis et al., 2019), by devising accelerated Mirror-Prox algorithm. Very recently, Ene et al. (2021) have further generalized the latter results by designing and analyzing a novel adaptive and accelerated AdaGrad version with per-coordinate step-sizes. Convergence properties of these universal (and some accelerated) adaptive algorithms under smooth, non-convex losses are unknown to date.

Adaptive methods for nonconvex optimization Following the popularity of neural networks, adaptive methods have attracted massive attention due to their favorable performance in training and their ease of tuning. The literature is quite vast, which is impossible to cover exhaustively here. Within the representative subset (Chen et al., 2019; Zaheer et al., 2018; Li & Orabona, 2019; Zou et al., 2019; Defossez et al., 2020; Alacaoglu et al., 2020a; Chen et al., 2020). The majority of the theoretical results on adaptive methods for nonconvex problems focus on their *in expectation* performance.

High probability results Ghadimi & Lan (2013) are the first to analyze SGD in the non-convex regime and provide tight convergence bounds. Nevertheless, their method requires a prior knowledge of the smoothness modulus and noise variance. In the context of adaptive methods, Li & Orabona (2020) considers delayed AdaGrad (with lag-one-behind step-size) for smooth, non-convex losses under subgaussian noise and proved $O(\sigma\sqrt{\log(T/\delta)}/\sqrt{T})$ rate. Under similar conditions, Zhou et al. (2018) proves convergence of order $O((\sigma^2\log(1/\delta))/T + 1/\sqrt{T})$ for AdaGrad. However, both works require the knowledge of smoothness to set the step-size. Moreover, Ward et al. (2019) guarantees that AdaGrad with scalar step-size convergences at $O((1/\delta)\log(T)/\sqrt{T})$ rate with high probability. Although their framework is oblivious to smoothness constant, their dependence of probability margin is far from optimal. More recently, under heavy-tailed noise having bounded pth moment for $p \in (1, 2)$, Cutkosky & Mehta (2021) proves a rate of $O(\log(T/\delta)/T^{(p-1)/(3p-2)})$ for clipped normalized SGD with momentum; nevertheless their method requires the knowledge of (a bound on) the behavior of the heavy tails.

3 Setup and Preliminaries

As we stated in the introduction, we consider the unconstrained minimization setting

$$\min_{x\in\mathbb{R}^d} f(x) := \mathbb{E}_{z\sim\mathcal{D}}\left[f(x;z)\right],$$

where the differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is a smooth and (possibly) nonconvex function.

We are interested is finding a first-order ϵ-stationary point satisfying $\|\nabla f(x_t)\|^2 \le \epsilon$, where $\|\cdot\|$ denotes the Euclidean norm for the sake of simplicity. As the standard measure of convergence, we will quantify the performance of algorithms with respect to *average* gradient norm, $\frac{1}{T}\sum_{t=1}^T \|\nabla f(x_t)\|^2$. It immediately implies convergence in *minimum* gradient norm, $\min_{t\in[T]} \|\nabla f(x_t)\|^2$. Moreover, note that if we are able to bound $\frac{1}{T}\sum_{t=1}^T \|\nabla f(x_t)\|^2$, then by choosing to output a solution $\bar{x}_T$ which is chosen uniformly at random from the set of query points $\{x_1, \dots, x_T\}$, then we ensure that $\mathbf{E}\|\nabla f(\bar{x}_T)\|^2 := \frac{1}{T}\sum_{t=1}^T \|\nabla f(x_t)\|^2$ is bounded.

A function is called G-Lipschitz continuous if it satisfies

$$|f(x) - f(y)| \le G\|x - y\|, \quad \forall x, y \in \text{dom}(f), \tag{1}$$

which immediately implies that

$$\|\nabla f(x)\| \le G, \quad \forall x \in \text{dom}(f). \tag{2}$$

A differentiable function is called L-smooth if it has L-Lipschitz gradient

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall x, y \in \text{dom}(\nabla f). \tag{3}$$

An equivalent characterization of smoothness is sometimes referred to as the "descent lemma" (Ward et al., 2019; Beck, 2017), i.e.,

$$|f(x) - f(y) - \langle \nabla f(y), x - y\rangle| \le \frac{L^2}{2}\|x - y\|^2. \tag{4}$$

Assumptions on oracle model: We denote stochastic gradients with $\widetilde{\nabla} f(x) = \nabla f(x; z)$, for some random vector drawn from distribution $\mathcal{D}$. Since our template embraces single-call algorithms, we use this shorthand notation for simplicity. An oracle is called (conditionally) unbiased if

$$\mathbb{E}\left[\widetilde{\nabla} f(x)|x\right] = \nabla f(x), \quad \forall x \in \text{dom}(\nabla f). \tag{5}$$

Gradient estimates generated by a first-order oracle are said to have bounded variance if they satisfy

$$\mathbb{E}\left[\|\widetilde{\nabla} f(x) - \nabla f(x)\|^2|x\right] \le \sigma^2, \quad \forall \in \text{dom}(\nabla f). \tag{6}$$

Finally, we assume that the stochastic gradient are bounded almost surely, i.e.,

$$\|\widetilde{\nabla} f(x)\| \le \tilde{G}, \quad \forall x \in \text{dom}(\nabla f). \tag{7}$$

Remark 1. *Bounded variance assumption (6) is standard in the analysis of stochastic methods (Lan, 2020). Similarly, for the analysis of adaptive methods in the nonconvex realm, it is very common to assume bounded stochastic gradients (see (Zaheer et al., 2018; Zhou et al., 2018; Chen et al., 2019; Li & Orabona, 2020) and references therein).*

4 PROPOSED ANALYSIS AND ADAGRAD

This section introduces our proposed proof technique as well as our main theoretical results for AdaGrad with proof sketches and respective discussions for clarifying key elements of our theoretical findings. We will present a high-level overview of our simplified, modular proof strategy while proving a complementary convergence result for AdaGrad under deterministic oracles. In the sequel, we refer to the name AdaGrad as the scalar step-size variant of the algorithm (also known as AdaGrad-Norm) as we present in Algorithm 1.

Algorithm 1 AdaGrad

Input: time horizon T , $x_1 \in \mathbb{R}^d$, step-size $\{\eta_t\}_{t\in[T]}$, $G_0 > 0$

1: **for** $t = 1, ..., T$ **do**
2: Generate $g_t = \widetilde{\nabla} f(x_t)$
3: $\eta_t = \frac{1}{\sqrt{G_0^2+\sum_{k=1}^{t}\|g_k\|^2}}$
4: $x_{t+1} = x_t - \eta_t g_t$
5: **end for**

Before moving forward with analysis, let us first establish the notation we will use to simplify the presentation. In the sequel, we use $[T]$ as a shorthand expression for the set $\{1, 2, ..., T\}$. We will use $\Delta_t = f(x_t) - \min_{x\in\mathbb{R}^d} f(x)$ as a concise notation for objective suboptimality and $\Delta_{\text{max}} = \max_{t\in[T+1]} \Delta_t$ will denote the maximum over Δ_t. In the rest of Section 4, we denote stochastic gradient of f at x_t by $g_t = \widetilde{\nabla} f(x_t) = \nabla f(x_t; z_t)$ and its deterministic equivalent as $\bar{g}_t := \nabla f(x_t)$. We also use the following notation for the noise vector $\xi_t := g_t - \bar{g}_t$.

Notice that AdaGrad (Alg. 1) does not require any prior knowledge regarding the smoothness modulus nor the noise variance. The main results in this section is Theorem 4.2, where we show that with high probability AdaGrad obtains an optimal rate $\tilde{O}(\log(1/\delta)/\sqrt{T})$ for finding an approximate stationary point. Moreover, Theorem 4.1 shows that in the deterministic case AdaGrad achieves an optimal rate of $O(1/T)$, thus establishing its universality.

4.1 TECHNICAL LEMMAS

We make use of a few technical lemmas while proving our main results, which we refer to in our proof sketches. We present them all at once before the main theorems for completeness. First, Lemmas 4.1 and 4.2 are well-known results from online learning, essential for handling adaptive stepsizes.

Lemma 4.1. *Let $a_1, ..., a_n$ be a sequence of non-negative real numbers. Then, it holds that*

$$\sqrt{\sum_{i=1}^{n} a_i} \leq \sum_{i=1}^{n} \frac{a_i}{\sqrt{\sum_{k=1}^{i} a_k}} \leq 2\sqrt{\sum_{i=1}^{n} a_i}$$

Lemma 4.2. *Let $a_1, ..., a_n$ be a sequence of non-negative real numbers. Then, it holds that*

$$\sum_{i=1}^{n} \frac{a_i}{\sum_{k=1}^{i} a_i} \leq 1 + \log\left(1 + \sum_{i=1}^{n} a_i\right)$$

The next lemma is the key for achieving the high probability bounds..

Lemma 4.3 (Lemma 3 in Kakade & Tewari (2008))**.** *Let X_t be a martingale difference sequence such that $|X_t| \leq b$. Let us also define*

$$\mathbf{Var}_{t-1}(X_t) = \mathbf{Var}\left(X_t \mid X_1, ..., X_{t-1}\right) = \mathbb{E}\left[X_t^2 \mid X_1, ..., X_{t-1}\right],$$

and $V_T = \sum_{t=1}^{T} \mathbf{Var}_{t-1}(X_t)$ as the sum of variances. For $\delta < 1/e$ and $T \geq 3$, it holds that

$$\mathbb{P}\left(\sum_{t=1}^{T} X_t > \max\left\{2\sqrt{V_T}, 3b\sqrt{\log(1/\delta)}\right\}\sqrt{\log(1/\delta)}\right) \leq 4\log(T)\delta \tag{8}$$

Please refer to first section in the appendix for the proofs of technical lemmas.

4.2 OVERVIEW OF PROPOSED ANALYSIS

We will start by presenting the individual steps of our proof and provide insight into its advantages. In the rest of this section, we solely focus on AdaGrad, however, the same intuition applies to the more general Algorithm 2 as we will make clear in the sequel. Using the shorthand notations of $\bar{g}_t = \nabla f(x_t)$ and $g_t = \nabla f(x_t; z_t) = \bar{g}_t + \xi_t$, the classical analysis begins with,

$$f(x_{t+1}) - f(x_t) \leq -\eta_t \|\bar{g}_t\|^2 - \eta_t \langle \bar{g}_t, \xi_t \rangle + \frac{L\eta_t^2}{2} \|g_t\|^2.$$

which is due to the smoothness property in Eq. (4). Re-arranging and summing over $t \in [T]$ yields,

$$\sum_{t=1}^{T} \eta_t \|\bar{g}_t\|^2 \leq f(x_1) - f(x^*) + \sum_{t=1}^{T} -\eta_t \langle \bar{g}_t, \xi_t \rangle + \frac{L}{2} \sum_{t=1}^{T} \eta_t^2 \|g_t\|^2.$$

The main issue in this expression is the $\eta_t \langle \bar{g}_t, \xi_t \rangle$ term, which creates measurability problems due to the fact hat η_t and ξ_t are dependent random variables. On the left hand side, the mismatch between η_t and $\|\bar{g}_t\|^2$ prohibits the use of technicals lemmas as we accumulate *stochastic* gradients for η_t. Moreover, we cannot make use of Holder-type inequalities as we deal with high probability results. Instead, we divide both sides by η_t, then sum over t and re-arrange to obtain a bound of the form,

$$\begin{aligned} \sum_{t=1}^{T} \|\bar{g}_t\|^2 &\leq \frac{\Delta_1}{\eta_1} + \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) \Delta_t + \sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t \rangle + \frac{L}{2} \sum_{t=1}^{T} \eta_t \|g_t\|^2 \\ &\leq \frac{\Delta_{\max}}{\eta_T} + \sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t \rangle + \frac{L}{2} \sum_{t=1}^{T} \eta_t \|g_t\|^2 \end{aligned} \tag{9}$$

This modification solves the two aforementioned problems, but we now need to argue about boundedness of function values, specifically the maximum distance to the optimum, $\Delta_{\max}$. In fact, neural networks with bounded activations (e.g. sigmoid function) in the last layer and some objective functions in robust non-convex optimization (e.g. Welsch loss Barron (2019)) satisfy bounded function values. However, this is a restrictive assumption to make for general smooth minimization problems and we will *prove* that it is bounded or at least it grows no faster than $O(\log(T))$. As a key element of our approach, we will show that this is the case for Algorithms 1 and 2. Now, we are at a position to state an overview of our proof strategy:

1. Show that $\Delta_{\max} \leq O(\log(T))$ with high probability or $\Delta_{\max} \leq O(1)$ (deterministic).
2. Prove that $\sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t \rangle \leq \tilde{O}(\sqrt{T})$ with high probability using Lemma 4.3.
3. Show that $\frac{L}{2} \sum_{t=1}^{T} \eta_t \|g_t\|^2 \leq O(\sqrt{T})$ by using Lemma 4.1.

For completeness, we will propose a simple proof for AdaGrad in the deterministic setting. This will showcase advantages of our approach, while providing some insight into it. We provide a sketch of the proof, whose full version will be accessible in the appendix.

Theorem 4.1. *Let x_t be generated by Algorithm 1 with $G_0 = 0$ for simplicity. Then, it holds that*

$$\frac{1}{T} \sum_{t=1}^{T} \|\nabla f(x_t)\|^2 \leq O\left(\frac{(\Delta_1 + L)^2}{T} \right).$$

Proof Sketch (Theorem 4.1). Setting $\bar{g}_t = \nabla f(x_t)$, deterministic counterpart of Eq. (9) becomes

$$\sum_{t=1}^{T} \|\bar{g}_t\|^2 \;\leq\; \frac{\Delta_{\max}}{\eta_T} + \frac{L}{2} \sum_{t=1}^{T} \eta_t \|\bar{g}_t\|^2 \;\leq\; (\Delta_{\max} + L) \sqrt{\sum_{t=1}^{T} \|\bar{g}_t\|^2},$$

where we obtain the final inequality using Lemma 4.1. Now, we show that Δ_{T+1} is bounded for any T. Using descent lemma and summing over $t \in [T]$,

$$f(x_{T+1}) - f(x^*) \leq f(x_1) - f(x^*) + \sum_{t=1}^{T} \left(\frac{L\eta_t}{2} - 1 \right) \eta_t \|\bar{g}_t\|^2$$

Now, define $t_0 = \max\left\{t \in [T] \mid \eta_t > \frac{2}{L}\right\}$, such that $\left(\frac{L\eta_t}{2} - 1\right) \leq 0$ for any $t > t_0$. Then,

$$f(x_{T+1}) - f(x^*) \leq \Delta_1 + \sum_{t=1}^{t_0}\left(\frac{L\eta_t}{2} - 1\right)\eta_t\|\bar{g}_t\|^2 + \sum_{t=t_0+1}^{T}\left(\frac{L\eta_t}{2} - 1\right)\eta_t\|\bar{g}_t\|^2$$
$$\leq \Delta_1 + \frac{L}{2}\sum_{t=1}^{t_0}\eta_t^2\|\bar{g}_t\|^2 \leq \Delta_1 + \frac{L}{2}\left(1 + \log\left(1 + L^2/4\right)\right),$$

where we use the definition of t_0 and Lemma 4.2 for the last inequality. Since this is true for any T, the bound holds for $\Delta_{\max}$, as well. Defining $X = \sqrt{\sum_{t=1}^{T}\|\bar{g}_t\|^2}$, the original expression reduces to $X^2 \leq (\Delta_{\max} + L)\,X$. Solving for X, plugging in the bound for $\Delta_{\max}$ and dividing by T results in

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(x_t)\|^2 \leq \frac{\left(\Delta_1 + \frac{L}{2}\left(3 + \log\left(L^2/4\right)\right)\right)^2}{T}.$$

□

Remark 2. *To our knowledge, the most relevant analysis was provided by Ward et al. (2019), which achieves $\mathcal{O}(\log(T)/T)$ convergence rate. Our new approach enables us to remove $\log(T)$ dependence at the cost of computing the bound on Δ_{max}.*

4.3 High probability convergence under stochastic oracle

Having introduced the building blocks of our strategy, we will now present the high probability convergence bound for AdaGrad (Algorithm 1). Let us begin by the departure point of our proof, which is Eq. (9)

$$\sum_{t=1}^{T}\|\bar{g}_t\|^2 \leq \underbrace{\frac{\Delta_{\max}}{\eta_T}}_{(*)} + \underbrace{\sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t\rangle}_{(**)} + \underbrace{\frac{L}{2}\sum_{t=1}^{T}\eta_t\|g_t\|^2}_{(***)} \tag{10}$$

We can readily bound expression $(***)$ using Lemma 4.1. Hence, what remains is to argue about high probability bounds for expressions $(*)$ and $(**)$, which we do in the following propositions.

Proposition 4.1. *Using Lemma 4.3, with probability $1 - 4\log(T)\delta$ and $\delta < 1/e$, we have*

$$\sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t\rangle \leq 2\sigma\sqrt{\log(1/\delta)}\sqrt{\sum_{t=1}^{T}\|\bar{g}_t\|^2} + 3(G^2 + G\tilde{G})\log(1/\delta).$$

The last ingredient of the analysis is the bound on Δ_t. The following proposition ensures a high probability bound of order $O(\log(t))$ on Δ_t under Algorithm 1.

Proposition 4.2. *Let x_t be generated by AdaGrad for $G_0 > 0$. With probability at least $1 - 4\log(t)\delta$,*

$$\Delta_{t+1} \leq \Delta_1 + 2L\left(1 + \log\left(\max\{1, G_0^2\} + \tilde{G}^2 t\right)\right) + G_0^{-1}(M_1 + \sigma^2)\log(1/\delta) + M_2,$$

where $M_1 = 3(G^2 + G\tilde{G})$ and $M_2 = G_0^{-1}(2G^2 + G\tilde{G})$.

As an immediate corollary, since the statement of Proposition 4.2 holds for any t, it holds for $\Delta_{\max}$ by definition. Hence, we have that $\max_{t\in[T]}\Delta_t = \Delta_{\max} \leq O\left(\Delta_1 + L\log(T) + \sigma^2\log(1/\delta)\right)$ with high probability for any time horizon T. In the light of the above results, we are now able to present our high probability bound for AdaGrad.

Theorem 4.2. *Let x_t be the sequence of iterates generated by AdaGrad. Under Assumptions 2, 6, 7, for $\Delta_{max} \leq O\left(\Delta_1 + L\log(T) + \sigma^2\log(1/\delta)\right)$, with probability at least $1 - 8\log(T)\delta$,*

$$\frac{1}{T}\sum_{t=1}^{T}\|\bar{g}_t\|^2 \leq \frac{(\Delta_{max} + L)\,G_0 + 3(G^2 + G\tilde{G})\log(1/\delta)}{T} + \frac{(\Delta_{max} + L)\,\tilde{G} + 2G\sigma\sqrt{\log(1/\delta)}}{\sqrt{T}}.$$

Remark 3. *Let the statement of Proposition 4.1 be denoted as event **A** and statement of Proposition 4.2 by event **B**. Theorem 4.2 essentially requires a lower bound for $\mathbb{P}(A \cap B)$. Aforementioned propositions already established that $\mathbb{P}(A) \geq 1 - 4\log(T)\delta$ and $\mathbb{P}(B) \geq 1 - 4\log(T)\delta$. By probability axioms it holds that $\mathbb{P}(A \cap B) \geq 1 - 8\log(T)\delta$, which we prove in the appendix.*

Proof Sketch (Theorem 4.2). Define $\bar{g}_t = \nabla f(x_t)$ and $g_t = \nabla f(x_t; z_t) = \bar{g}_t + \xi_t$. By Eq. (10),

$$\sum_{t=1}^{T} \|\bar{g}_t\|^2 \leq \frac{\Delta_{\max}}{\eta_T} + \sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t \rangle + \frac{L}{2} \sum_{t=1}^{T} \eta_t \|g_t\|^2.$$

Invoking Lemma 4.1 on the last sum and using Proposition 4.1 for the second expression, we have with probability at least $1 - 4\log(T)\delta$

$$\sum_{t=1}^{T} \|\bar{g}_t\|^2 \leq (\Delta_{\max} + L) \sqrt{G_0^2 + \sum_{t=1}^{T} \|g_t\|^2} + 2\sigma\sqrt{\log(1/\delta)} \sqrt{\sum_{t=1}^{T} \|\bar{g}_t\|^2} + 3(G^2 + G\tilde{G})\log(1/\delta)$$

Finally, we use the bounds on the gradient norms $\|\bar{g}_t\| \leq G$ and $\|g_t\| \leq \tilde{G}$, re-arrange the terms and divide both sides by T. Due to Proposition 4.2, with probability at least $1 - 8\log(T)\delta$,

$$\frac{1}{T} \sum_{t=1}^{T} \|\bar{g}_t\|^2 \leq \frac{(\Delta_{\max} + L)\, G_0 + 3(G^2 + G\tilde{G})\log(1/\delta)}{T} + \frac{(\Delta_{\max} + L)\, \tilde{G} + 2G\sigma\sqrt{\log(1/\delta)}}{\sqrt{T}},$$

where $\Delta_{\max} \leq O\left(\Delta_1 + L\log(T) + \sigma^2 \log(1/\delta)\right)$. We keep $\Delta_{\max}$ in the original bound due to space constraints. □

5 Generalization to AGD Template

Having proven the high probability convergence for AdaGrad, we will now present an extension of our analysis to the more general accelerated gradient (AGD) template, which corresponds to a specific reformulation of Nesterov's acceleration (Ghadimi & Lan, 2016).

Algorithm 2 Generic AGD Template

Input: Horizon T , $\tilde{x}_1 = x_1 \in \mathbb{R}^d$, $\alpha_t \in (0, 1]$, step-sizes $\{\eta_t\}_{t\in[T]}$, $\{\gamma_t\}_{t\in[T]}$

for $t = 1, ..., T$ **do**
  $\bar{x}_t = \alpha_t x_t + (1 - \alpha_t)\tilde{x}_t$
  Set $g_t = \nabla f(\bar{x}_t)$ or $g_t = \widetilde{\nabla} f(\bar{x}_t) = \nabla f(x_t; z_t)$
  $x_{t+1} = x_t - \eta_t g_t$
  $\tilde{x}_{t+1} = \bar{x}_t - \gamma_t g_t$
end for

This particular reformulation was recently referred to as linear coupling (Allen-Zhu & Orecchia, 2016), which is an intricate combination of mirror descent (MD), gradient descent (GD) and averaging. In the sequel, we focus on two aspects of the algorithm; averaging parameter α_t and selection of (adaptive) step-sizes η_t and γ_t. We could recover some well-known algorithms from this generic scheme depending on parameter choices, which we display in Table 1.

Our reason behind choosing this generic algorithm is two-fold. First, it helps us demonstrate flexibility of our simplified and modular proof technique by extending it to a much general algorithmic template. Second, as an integral element of this scheme, we wanted to study the notion of averaging (or equivalently momentum (Defazio, 2021)), which is an important primitive for several machine learning and optimization problems. For instance, it is necessary for achieving *accelerated* convergence in convex optimization (Nesterov, 1983; Kavis et al., 2019), while it helps improve performance in

neural network training and stabilizes the effect of noisy feedback (Defazio, 2021; Sutskever et al., 2013; Liu et al., 2020).

Next, we will briefly introduce instances of Algorithm 2, their properties and the corresponding parameter choices. We identify algorithms regarding the choice of averaging parameter α_t and step sizes η_t and γ_t. The averaging parameter α_t has two possible forms: $\alpha_t = 2/(t+1)$ for weighted averaging and $\alpha_t = 1/t$ for uniform averaging. We take $\alpha_t = 2/(t+1)$ by default in our analysis, as it is a key element in achieving acceleration in the convex setting. Let us define the AdaGrad step size once more, which we use to define η_t and γ_t,

$$\widetilde{\eta}_t = (G_0^2 + \sum_{k=1}^{t} \|g_k\|^2)^{-1/2}, \qquad G_0 > 0. \tag{11}$$

The first instance of Algorithm 2 is the AdaGrad, i.e. $x_{t+1} = x_t - \widetilde{\eta}_t g_t$. Since $\tilde{x}_1 = x_1$ by initialization, we have $\bar{x}_1 = \tilde{x}_1 = x_1$. The fact that $\eta_t = \gamma_t = \widetilde{\eta}_t$ implies the equivalence $\tilde{x}_t = x_t$ for any $t \in [T]$, which ignores averaging step and results in a single line of update. The second instance we recover is AdaGrad with averaging, i.e.,

$$\begin{aligned} \bar{x}_t &= \alpha_t x_t + (1-\alpha_t)\bar{x}_{t-1} \\ x_{t+1} &= x_t - \alpha_t \eta_t g_t, \end{aligned} \tag{12}$$

where $\eta_t = \widetilde{\eta}_t$ and $\gamma_t = 0$. For the initialization $x_1 = \tilde{x}_1$, we can obtain by induction that $\tilde{x}_t = \bar{x}_{t-1}$, hence the scheme above. The final scheme we will analyze is RSAG algorithm proposed by Ghadimi & Lan (2016). It selects a step size pair that satisfies $\gamma_t \approx (1+\alpha_t)\widetilde{\eta}_t$ and $\eta_t = \widetilde{\eta}_t$, generating a 3-sequence algorithm as in the original form of Algorithm 2.

Algorithm	**Weights** (α_t)	**Step-size** (η_t,γ_t)
AdaGrad	N/A	$\eta_t = \widetilde{\eta}_t$ $\gamma_t = \eta_t$
AdaGrad w/ Averaging	$\alpha_t = 2/(t+1)$ or $\alpha_t = 1/t$	$\eta_t = \alpha_t \widetilde{\eta}_t$ $\gamma_t = 0$
Adaptive RSAG(Ghadimi & Lan, 2016)	$\alpha_t = 2/(t+1)$	$\eta_t = \widetilde{\eta}_t$ $\gamma_t = (1+\alpha_t)\eta_t$
AcceleGrad(Levy et al., 2018)	$\alpha_t = 2/(t+1)$	$\eta_t \approx \frac{1}{\alpha_t}\widetilde{\eta}_t$ $\gamma_t \approx \widetilde{\eta}_t$

Table 1: Example methods covered by the generic AGD template.

Before moving on to convergence results, we have an interesting observation concerning time-scale difference between step sizes for AdaGrad, AdaGrad with averaging and RSAG. Precisely, γ_t is always *only* a constant factor away from η_t. This phenomenon has an immediate connection to acceleration in the convex realm, which is of independent interest to us and we will return to that later.

Having defined instances of Algorithm 2, we will present high probability convergence rates for them. Similar to Eq. (10) for AdaGrad, we first define a departure point of similar structure in Proposition 5.1, then apply Proposition 4.1 and 4.2 in the same spirit as before to finalize the bound.

Let us define $\bar{g}_t = \nabla f(\bar{x}_t)$ and $g_t = \nabla f(\bar{x}_t; z_t) = \bar{g}_t + \xi_t$. Following the notation in (Ghadimi & Lan, 2016), let $\Gamma_t = (1-\alpha_t)\Gamma_{t-1}$ with $\Gamma_1 = 1$. Now, we can begin with the departure point of the proof, which is due to Ghadimi & Lan (2016).

Proposition 5.1. *Let x_t be generated by Algorithm 2 where $g_t = \nabla f(\bar{x}_t; z_t) = \bar{g}_t + \xi_t$ and $\bar{g}_t = \nabla f(\bar{x}_t)$. Then, it holds that*

$$\sum_{t=1}^{T} \|\bar{g}_t\|^2 \leq \frac{\Delta_{max} + 2L}{\eta_T} + \frac{L}{2\eta_T} \sum_{t=1}^{T} \underbrace{\left[\sum_{k=t}^{T} (1-\alpha_k)\Gamma_k\right] \frac{\alpha_t}{\Gamma_t}}_{(*)} \frac{(\eta_t - \gamma_t)^2}{\alpha_t^2} \|g_t\|^2 + \underbrace{\sum_{t=1}^{T} -\langle \bar{g}_t, \xi_t \rangle}_{(**)} .$$

Next, we will deliver the complementary bound on term $(*)$ in Proposition 5.1.

Proposition 5.2. *Using the recursive definition of Γ, we have*

$$\left[\sum_{k=t}^{T} (1-\alpha_k)\Gamma_k\right] \frac{\alpha_t}{\Gamma_t} \leq \begin{cases} 2 & \text{if } \alpha_t = \frac{2}{t+1}; \\ \log(T+1) & \text{if } \alpha_t = \frac{1}{t}. \end{cases}$$

Finally, we present the high probability convergence rates for AdaGrad with averaging and RSAG.

Theorem 5.1. *Let x_t be the sequence of iterates generated by AdaGrad with averaging or adaptive RSAG. Under Assumptions 2, 6, 7, for $\Delta_{max} \leq O\left(\Delta_1 + L\log(T) + \sigma^2 \log(1/\delta)\right)$, with probability $1 - 8\log(T)\delta$,*

$$\frac{1}{T}\sum_{t=1}^{T} \|\bar{g}_t\|^2 \leq \frac{G_0\left(\Delta_{max} + 3L + L\log\left(\max\{1, G_0^{-2}\} + \tilde{G}^2 T\right)\right) + 3(G^2 + G\tilde{G})\log(1/\delta))}{T} + \frac{\tilde{G}\left(\Delta_{max} + 3L + L\log\left(\max\{1, G_0^{-2}\} + \tilde{G}^2 T\right)\right) + 2G\sigma\sqrt{\log(1/\delta)}}{\sqrt{T}}$$

A discussion on acceleration and nonconvex analysis: AGD and its variants are able to converge at the fast rate of $\mathcal{O}(1/T^2)$ (Nesterov, 2003) for smooth, convex objectives, while matching the convergence rate of GD in the nonconvex landscapes. In fact, the mechanism that allows them to converge faster could even restrict their performance when convexity assumption is lifted. We will conclude this section with a brief discussion on this phenomenon.

As we mentioned previously, step-sizes for AdaGrad, its averaging variant and RSAG have the same time scale up to a constant factor. However, AcceleGrad, an accelerated algorithm, has a time-scale difference of $\mathcal{O}(t)$ between η_t and γ_t, and it runs with a modified step-size of $\eta_t = \left(1 + \sum_{k=1}^{t} \alpha_k^{-2} \|g_t\|^2\right)^{-1/2}$. This scale difference is not possible to handle with standard approaches or our proposed analysis. If we look at second term in Proposition 5.1, it roughly evaluates to

$$\frac{L}{2\eta_T} \sum_{t=1}^{T} \frac{\eta_t^2}{\alpha_t^4} \|g_t\|^2,$$

where each summand is $\mathcal{O}(t^4)$ orders of magnitude larger compared to other methods. A factor of $\alpha_t^{-2} = t^2$ is absorbed by the modified step-size, but this term still grows faster than we can manage. We aim to understand it further and improve upon in our future attempts.

6 Conclusions

We propose a simple and modular high probability analysis for a class of AdaGrad-type algorithms. Bringing AdaGrad into the focus, we show that our new analysis techniques goes beyond and generalizes to the accelerated gradient template (Algorithm 2) which individually recovers AdaGrad, AdaGrad with averaging and adaptive version of RSAG (Ghadimi & Lan, 2016). By proposing a modification over standard analysis and relying on concentration bounds for martingales, we achieve high probability convergence bounds for the aforementioned algorithms *without* requiring the knowledge of smoothness L and variance σ while having optimal dependence of $\log(1/\delta)$ on probability margin δ. To our knowledge, this is the first such result for adaptive methods, including AdaGrad.

REFERENCES

Ahmet Alacaoglu, Yura Malitsky, and Volkan Cevher. Convergence of adaptive algorithms for weakly convex constrained optimization, 2020a.

Ahmet Alacaoglu, Yura Malitsky, Panayotis Mertikopoulos, and Volkan Cevher. A new regret analysis for Adam-type algorithms. In Hal Daum III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 202–210. PMLR, 13–18 Jul 2020b. URL http://proceedings.mlr.press/v119/alacaoglu20b.html.

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent, 2016.

Jonathan T. Barron. A general and adaptive robust loss function. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4326–4334, 2019. doi: 10.1109/CVPR.2019.00446.

Amir Beck. *First-Order Methods in Optimization*. SIAM-Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2017. ISBN 1611974984.

Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks, 2020.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1x-x309tm.

Ashok Cutkosky and Harsh Mehta. High-probability bounds for non-convex stochastic optimization with heavy tails, 2021.

Aaron Defazio. Momentum via primal averaging: Theoretical insights and learning rate schedules for non-convex optimization, 2021.

Alexandre Defossez, Lon Bottou, Francis Bach, and Nicolas Usunier. On the convergence of adam and adagrad, 03 2020.

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

Alina Ene, Huy L. Nguyen, and Adrian Vladu. Adaptive gradient methods for constrained convex optimization and variational inequalities, 2021.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.*, 156(12):5999, March 2016. ISSN 0025-5610. doi: 10.1007/s10107-015-0871-8. URL https://doi.org/10.1007/s10107-015-0871-8.

Haiwen Huang, Chang Wang, and Bin Dong. Nostalgic adam: Weighting more of the past gradients when designing the adaptive learning rate. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2556–2562. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/355. URL https://doi.org/10.24963/ijcai.2019/355.

Sham M. Kakade and Ambuj Tewari. On the generalization ability of online ¡i¿strongly¡/i¿ convex programming algorithms. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pp. 801808, Red Hook, NY, USA, 2008. Curran Associates Inc. ISBN 9781605609492.

Ali Kavis, Kfir Y. Levy, Francis Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 6260–6269. Curran Associates, Inc., 2019.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Guanghui Lan. *First-order and Stochastic Optimization Methods for Machine Learning*. Springer, 2020.

Kfir Y Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. In *Neural and Information Processing Systems (NeurIPS)*, December 2018.

Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 983–992. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/li19c.html.

Xiaoyu Li and Francesco Orabona. A high probability analysis of adaptive sgd with momentum, 2020.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum, 2020.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate, 2019.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

Yurii Nesterov. Introductory lectures on convex optimization. 2004, 2003.

Sashank Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014. ISBN 1107057132.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

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

Phuong T. Tran and L. T. Phong. On the convergence proof of amsgrad and a new version. *IEEE Access*, 7:61706–61716, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6677–6686. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/ward19a.html.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models?, 2020.

Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *ArXiv*, abs/1808.05671, 2018.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 928935. AAAI Press, 2003. ISBN 1577351894.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.